Collaborative Context Recognition for Mobile Devices

Pertti Huuskonen, Jani Mäntyjärvi and Ville Könönen

1 Introduction

The next wave of mobile applications is at hand. Mobile phones, PDAs, cameras, music players, and gaming gadgets are creating a connected mobile ecosystem where it is possible to implement systems with significant embedded intelligence. Such advances will make it possible to move many functions of the current PC-centric applications to the mobile domain. Since the inherent difficulties that come with mobility—limited UIs, short attention spans, power dependency, intermittent connectivity, to name but a few—are still not going away, new solutions are needed to make mobile computing satisfactory. We are facing the paradox of cramming ever more functions into our ever more portable devices, while seeking to achieve radically better usablility and semi-usable automated intelligence. How could we pull this off?

In the HCI (Human-Computer Interaction) community it is almost universally acknowledged that context awareness is a key function in next generation interfaces. To achieve more fluid interaction with the users, systems would need to develop a sense of the users' activities beyond the current application they are dealing with. The systems should have a rough idea of the users' location, social situation, tasks, activities, and many other factors that characterise human life. With this information it will be easier (though not trivial) to customise the offered information such that it is more relevant for the users' situation.

Mobile devices are following their users to all kinds of places. Therefore, they are taking part in situations that their desktop counterparts never needed to deal

Pertti Huuskonen
Nokia Research Center, Visiokatu 1, 33720 Tampere, Finland e-mail: pertti.huuskonen@nokia.com

Jani Mäntyjärvi and Ville Könönen
VTT Technical Research Centre of Finland, Kaitoväylä 1, 90571 Oulu, Finland e-mail: jani.mantyjarvi@vtt.fi, ville.kononen@vtt.fi

H. Nakashima et al. (eds.), *Handbook of Ambient Intelligence and Smart Environments*,
DOI 10.1007/978-0-387-93808-0_10,

with. For instance, the desktop PC never had to keep track of constantly changing locations. Some mobile devices are considered rather personal and are therefore usually carried along with the users practically everywhere, all the time. Many people consider their mobile phones to be the central hubs for their lives, expected to be available at all times. As such devices are moving with people, they become part of the social situations the users immerse themselves in. The devices, however, rarely have any idea of the situation around them. We seek to change that.

1.1 Humans: Context-aware Animals

We, humans, are the most advanced context-aware entities on the planet. Large areas of our brains are connected with functions that, if implemented in machines, would fall in the domain of context awareness. We are, for instance, automatically aware of the place we are in, activities that take place around us, and other sentient beings present. We automatically follow events and developments around us and, if needed, take action (e.g. flee predators).

Fig. 1 Humans, context-aware animals, spend much of their early lives learning the proper behaviour for social situations through observing other people and imitating them.

A great deal of human context-aware behavior can be explained with primitive functions—survival, reproduction, subsistence. Over the course of time, more and more of our brain functions have been devoted towards social behavior. Several of the most recently evolved brain structures have become necessary to deal with the range of complications found in human societies. Such features are far less pronounced with other animals [56].

Consider the case when you arrive late into a department meeting (or school, theatre, concert...). You know how to enter and find your place without causing too much disturbance in the event. You are aware of the proper behavior for that event. How? First, you have been educated in this throughout your life. Growing up, you have been immersed in all kinds of situations where you have learned the proper behaviour from other people. Second, you have become an expert in interpreting the little clues in your environment that suggest how to behave: how people converse, sit or stand, flock. Their dresses, the furniture, the decoration, in fact everything in the place gives you hints on the situation. Manners, social etiquette, could be seen as rules for context-driven behaviour[1].

Humans are quite good in imitating other humans. We may consciously or subconsciously adapt our behavior to match that of other people. This may happen, for instance, to appear polite, to be safer, or to show appreciation. Or, walking with a friend, we may adapt our pace to march in rhythm. Parts of our brain appear to be associated with observing other people and simulating their behavior [46], which makes it easier for us to adapt.

Of course, human life is more complicated than that. We are deliberately simplifying the situation to make our point: people know how to deal with all kinds of situations and adapt. How could we teach machines to do the same—even to a rudimentary level?

Our approach is use a simplified model of human behavior for context awareness: when in doubt, do as others do. When lost, ask others for advice.

Our focus in this chapter is to describe the principle of the proposed scheme for collaborative context understanding, some decision making algorithms and networking issues in the domain of connected mobile devices. Further, we outline a platform for realising the scheme. We conclude with implications of the method towards future systems.

2 Context From Collaboration

In signal understanding systems it is often useful to have reasonable starting hypotheses of the expected inputs. For instance, in a speech recognition system, it is beneficial to know the kind of language that the speaker will most likely utter. Such starting hypotheses, when accurate, can drastically improve the efficiency of recognition. In classic AI terms, good initial conditions can significantly reduce the search space.

Context aware systems are fundamentally doing signal understanding. Their key function is combine sensor signals from multiple sources, both physical and informational, and process those signals such that an interpretation of the situation is formed. Therefore context aware systems will benefit of good initial hypotheses just like any signal understanding system.

A useful way to form initial hypotheses for context recognition is to ask surrounding devices what their interpretation of the current situation is, and then to abstract an internal context hypothesis out of those interpretations. This is the approach we propose in this chapter.

[1] This is a mechanistic view, and is used here only to motivate our approach. Human sciences have shown that analyzing and modeling human behavior is a vast sea of research, far from being well understood.

The context aware nearby devices will have already tried to interpret the situation, or possibly their users have already explicitly set some of the context clues (such as the ringtone volume). This information is the more likely to be available the more devices there are nearby. On the other hand, if there are no nearby devices, then probably the situation is less socially sensitive, so it matters less if context is interpreted incorrectly.

This scheme—collaborative context recognition—requires communication channels between the devices. Such short-range links are already well available, though arguably not yet ubiquitous. For instance, most smartphones contain Bluetooth transceivers, and wireless LAN support is becoming increasingly common in all kinds of devices. In general, there is a trend in consumer devices towards wireless links.

Obviously the links are not enough: it matters what kind of data is exchanged. The devices need to share common definitions of context. Ontologies for context recognition are needed before useful context exchanges can happen between devices, or at least a shared set of context tags must be agreed.

3 Mobile Context Awareness

Context awareness has been an active field for well over a decade. In the space of computing research, it falls in the cross-section of ubiquitous computing, artificial intelligence, and human-computer interaction. With the recent advances in mobile technologies, particularly mobile telephony and mobile media players, the focus of context awareness research is shifting towards mobile systems. This is quite appropriate, since context information is arguably richer when the devices are moving around with their users. In this section we summarize some typical approaches to mobile context awareness.

3.1 Context Sources

Context aware systems reported in the literature obtain their context data from a number of different context sources. The most popular form of context data today is location, since navigation systems are now commonplace in vehicles and starting to appear in pedestrian applications. These systems may not directly offer context aware features but offer excellent ground for implementing them.

Various kinds of positioning systems have been applied, including GPS, cellular networks, wireless lans, ultrasonic and RFID tags, and Bluetooth devices [32, 2, 1, 13]. The position data they produce varies significantly in terms of accuracy, noise, intermittency. The systems reported in the literature often make use of several techniques simultaneously to improve the accuracy and reliability, for instance falling back to cellular network identification where GPS is unavailable.

Environmental sensors are often part of context-aware ubicomp systems. Typical household applications measure temperatures, air pressure, gas concentrations or humidity levels [20, 4]. Such ambient parameters usually give slowly changing data that yield only general suggestions of the context, so they need to be combined with other data sources. Some smart space projects have experimented with sensored floors and furniture to recognize human presence and/or activities [19, 48, 44].

A number of experimenters have measured biometric data on humans for use in context awareness [42, 43, 25]. Signals of biological origin—heart rate, breathing rate, skin conductivity, ECG, body temperatures, to name a few—can be quite troublesome to measure reliably, particularly in the case of moving people. As a result, many reported systems have had to focus on instrumentation issues. However, with the increase of low-cost, low-power wireless sensors, we are expecting a surge in biometric context data sets.

Acceleration sensors have been added to mobile devices with the general aim of studying the movements of their users. The sensors have become so cheap and accurate that they are today found in many high-end mobile phones and media players and consumer game devices. Again, the focus in consumer devices to date has not been context awareness, but the richness of acceleration data has made it a popular target for context awareness research [3, 27, 54, 11].

Besides position and acceleration sensing, various sensors have been tested with mobile devices, such as touch or proximity [17]. The internal functions of the mobile devices also give useful information on the users' activities. Monitors for key presses, application launches, media viewing, calling and messaging are being reported [45, 26].

The multimedia functions in modern gadgets offer rich audio-visual data. Mobile phones and digital cameras today produce billions of images that can be used as sensors for various kinds of data—for instance, illumination parameters, movement detection, or presence of people or vehicles [28]. Microphones are being trialled for differentiating between music, speech and noise [26] or for measuring the properties of the aural space. In wearable computers, cameras and microphones have been used for context sensing [52].

Processing sensor measurements into analyzed signal features can be quite involved. Sometimes signal processing can be bypassed—for instance, if the data already resides in symbolic form. The informational "sensors" available in the devices and through online sources offer context data that is already easily computer readable (though not easily understood by them). Online calendars, email systems, instant messaging and blogs have been used as data sources for context aware applications [7].

The presence of people can be sensed in a number of ways, including infrared motion detectors, computer vision, tags and radio signals (e.g. cellular or Bluetooth transceivers). However, the largest use of these technologies to date has been in property monitoring and surveillance, not in context awareness.

3.2 Application Areas

To date, most context aware applications are based on some form of location sensing. Popular types include situated messaging [39], location-aware advertising [40], geotagging of images [38], location-based searches (e.g. Google Maps). Location is clearly the context type that has received most use both in research and in industry [15].

Driven by the success of navigation tools, the market will be bustling with location aware applications in the coming years. Suggestive evidence can be found, for instance, in the applications submitted to the Android developer contest (`http://code.google.com/android/adc_gallery/`). About half of those applications are based on the positioning capabilities of the upcoming mobile phone platform.

One rising trend is combining positioning data with media. Geotagging capabilities for images and video have been added in a number of on-line media bases [37, 53]. Such tools allow users to attach location information to the media they have created, and use location later when searching for media created by themselves—and others. Some devices equipped with GPS embed position data in the image metadata, while other tools automate later manual tagging [55]. In the near future, other forms of context data will find uses in media files [51, 31]. Context-aware messaging has been tried for text messages [21] or advertisements [40]. In those trials, location was again the most frequently applied context type, but provisions were made for future additions.

Presence information has been popular in IM (instant messaging) applications for a long time. Users enter their availability (online, back soon, not reachable...) so other people know if they can be contacted. Variations of this scheme have been proposed for phonebooks [50]. More recently, microblogging applications such as Jaiku and Twitter have experimented with enhanced presence information that includes user's current position and state of activity, as deduced by their client software.

Activity analysis of humans has been, besides location, the second most popular topic in mobile context awareness research. A number of approaches have been tried to recognizing users' physical states, for instance sitting, walking, running, sleeping [10, 47, 16]. This state information, in turn, has been used in combination with other context data for various purposes, or for physiological analysis such as estimating one's energy consumption [22].

Context awareness is often said to be the key to intelligent user interfaces. Some successes have been reported [18], and some failures, including the Microsoft paperclip. In mobile phones, context information could be used to overcome the problem arising from users' short attention spans [41]. Rapid sensing of the use situation could help to focus the interaction to the crucial information.

At the moment, context awareness research has mostly produced novel features for existing UI types, such as shaking a music player to shuffle songs in recent iPod Nanos. It could be argued that once context aware features for UIs are understood well enough, they are re-implemented with more traditional computation techniques. In this way, context awareness echoes the development patterns found in the larger area of artificial intelligence.

The class of applications for lifelogging (or memory prosthesis, augmented memory) is closely related to context awareness [30, 12]. Such applications record a number of data items people have encountered in their lives: for instance, locations they have visited, people they have met, pictures they have taken, web pages and messages they have read or written. In addition, multiple types of sensor data is typically recorded to the databases for later abstraction and recognition. Lifelogging usually emphasises immediate storage and later retrieval of life data, whereas context awareness often seeks to recognize context on the fly. These two approaches are complementary, and can fruitfully feed each other.

Finally, new gaming devices, both consoles (such as Nintendo Wii) and mobile phones (such as Apple iPhone) have started using acceleration sensors as key inputs. Also, cameras have been applied to gaming (e.g. in Sony Eyetoy). This is introducing a new class of games where the physical world meets the game world. This approach is adopted on a larger scale in pervasive games, that take place in the real world with augmented object. Some of such games have made use of context awareness [5, 35].

3.3 Context Recognition

Mobile context recognition has fascinated researchers from the mid 90's when "smarter" mobile phones and PDAs appeared and positioning and miniaturised sensors emerged. Main challenges in building context recognition systems are (1) how to perform mapping from data obtained from suitable context sources to meaningful situations of a mobile device user, (2) how to implement methods to extract context data on a mobile device, and (3) how to design algorithms able to adapt user's routines and exceptions.

Research on context recognition has been conducted using various approaches; HMM, neural networks, SOM, dynamic programming, data mining, etc. Very attractive context recognition rates have been reported—up to 90%. However, context recognition can never be bulletproof and approaches—on UI level and on algorithm level - for tolerating uncertainty are also under development.

A central element of context recognition on mobile terminals is the framework for managing context data processing on a device. Several suggestions for such frameworks have been developed [26, 45, 8].

While such frameworks provide platform dependent solutions for fluent processing, they can't solve the inherent problem of uncertainty. Ideally, context processing should give data that could be used up to prediction and recommendation levels. Collaborative context recognition seeks to provide some improvement to recognition accuracies.

3.4 Distributed Context Awareness

A basic element of any context recognition system is a pattern classification algorithm for mapping observations to context classes. However, the classification is usually more or less inaccurate. Errors are caused by imperfect observations and the model in the pattern classification system.

Collaborative context reasoning (CCR) seeks to decrease errors by providing joint context recognition based on several imperfect context recognition systems located in the same context. The objective is that recognition systems can utilize information available from each other, and thus their joint context recognition accuracy can be significantly higher than individual context recognition accuracies. In the next section, we discuss social rules and mechanisms for CCR, with an emphasis on voting techniques.

4 Making Rational Decisions

In *cooperative distributed problem solving*, there exists a special entity, for example the designer of the system, that defines an interaction protocol for all agents in the system and a function that stipulates how the agents select their actions. Context recognition systems differ from cooperative distributed problem solving systems in that all context recognizers act autonomously and implement their own context selection procedures. In CCR, individual context recognizers then input their context estimates to an external mechanism that combines individual estimates for providing more accurate joint context estimates. The joint context estimate can be either enforcing in the sense that the context recognizers should use the joint estimate even if it is contradictory to their own individual estimates or it can be more free; individual context recognizers can utilize this information if they are not very confident about their own estimates.

As the context recognizers make their own estimates autonomously, a reasonable assumption is that the context recognizers act rationally, i.e. they maximize their own welfare. Therefore, it is rational to utilize joint context estimates only if the context recognizer benefit from this additional information. Later in this section, discussing communication requirements, we argue that linking confidence values to context estimates provide a way to evaluate the usefulness of joint context estimates.

We start our discussion by going inside a context recognizer and placing its functionality into the decision making framework. Then we proceed with discussing different voting mechanisms and explaining how to use voting as a CCR method.

4.1 Distributed Decision Making

In decision making, the goal is to make optimal decisions with respect to a utility function modelling preferences of a decision maker. In distributed decision making, the decision maker has to take into account also decisions of other active components in the systems, i.e. other decision makers.

In the traditional game theory, a decision maker (player) does not only know its own utility function but also utility functions of all other decision makers in the system. Based on this knowledge, all the decision makers are able to calculate an equilibrium solution, i.e. a solution from which it is not rational to deviate for any individual decision maker alone. However, in real-world problems, having such comprehensive knowledge about the utility functions of all decision makers is not possible. Usually it is even not possible to learn these utility functions by continuously interacting with other decision makers. Therefore some other opponent modeling techniques are needed, for example probabilistic opponent models. A thorough dissection of the basic game theoretic methods is [36].

In context recognition systems, context recognizers select a context class from a set of all context classes. Putting this selection task into the decision making framework we have to associate a utility value to each context class. How to determine this value, depends on a pattern classification algorithm that maps sensory input to the contexts. Later in this section, we use as an example the Mimimun-distance classifier that simply calculates a distance of the current sensory input from the prototype (ideal) vectors of all context classes and selects a class with the minimum distance. With this method, Euclidean distance between the current input and a prototype vector is used as a utility value.

4.2 Voting Protocols as Distributed Decision Making Strategies

Distributed rational decision making mechanisms such as auctions and voting protocols are extensively studied in the game theory and machine learning research fields. A good survey of differerent mechanisms is [49]. Using voting mechanisms as a basis for distributed pattern recognition is studied in [29]. Committee machines provide an alternative way to combine different classifiers, see e.g. [14]. Combining local context information with the information from other context recognizers is studied in [33].

In voting mechanisms, the setting includes several independent voters that vote on the common subject. In addition to the voters there is a voting protocol, i.e. a social rule that stipulates how to combine individual votes to the final outcome.

Next we will discuss several commonly used voting protocols. In each case we assume that there exist m alternatives of which one is selected to be a final outcome of the voting.

- *Borda count*. Each voter arranges the alternatives to an order that portray her preferences. The first choice gets m votes, the second one $m-1$ votes and the last one only 1 vote. The final outcome of the scheme is an alternative with the maximal number of votes. Note that this is exactly the voting scheme that is used in the Eurovision Song Contest.
- *Majority voting*. Each voter gives a vote to her preferred alternative. The final outcome of the scheme is an alternative with the maximal number of votes. The plain majority voting mechanism is used formally in direct elections and informally for doing daily practical things, for example deciding where to go for eating.
- *Weighted majority voting*. This scheme differs from the plain majority voting in a way that each voter weights its vote with a confidence value. This value tells to other voters how sure the voter is about its choice. Note that the weighted majority voting is actually a broad class of different voting schemes with different weighting methods. For example, paper referees are often asked also to judge their own expertise on the research area of the paper to be evaluated.
- *Binary protocol*. In this protocol, only two options are presented to the voters at each time step. Then an alternative with the maximum number of votes is competed against a new alternative. The outcome of this scheme is dependent on the order in which the alternatives are presented to the voters.

By selecting diffent weighting methods, weighted majority voting brovides a reasonably versatile class of different voting schemes for most application areas. In the remaining part of this text, we will focus on the weighted majority voting and its utilization in CCR.

Another important property of the above discussed voting protocols is that they assume that all the voters act sincerely. When discussing voting as a CCR method, we also assume sincerity of the voters. However, we also deliberate consequences of insicerity in applications that utilize CCR.

4.3 Voting as CCR Method

Here we view a context recognition system as a voter that votes on different contexts. Although there can be several context recognizers in the same environment, they all might have a slightly different model of their context and therefore they are able to complete each other. This hopefully leads to an improved joint recognition accuracy.

4.3.1 Confidence Evaluation

As discussed above, by using a weighted majority voting scheme, a voter reveals how confident she is about her confidence estimate. In other words, a context recognizer should judge how well its internal model, i.e. parameters of a pattern recognition method and system, can capture the relationship between sensory information and estimated context classes.

Unfortunately, such a confidence information is not directly available from all pattern recognition methods such as Support Vector Machines, or Feed-forward neural networks [9]. Here we present a simple pattern recognition method, the Minimum-distance classifier, and discuss a related confidence estimation method.

Consider a classification task where we have to associate an N-dimensional sample $\mathbf{s}$ to one of the C-classes. For example, dimensions can be features computed from sensory inputs. For each class $j = 1,\ldots,C$, we have I^j example samples $\mathbf{x}_i^j, i = 1,\ldots,I^j$. Further, let $\mathbf{c}^j$ represent the ideal vector, i.e. a vector that represents the class in the best possible way. A natural way to choose the ideal vector is to use the class mean vector, i.e.:

$$\mathbf{c}^j = \frac{1}{I^j}\sum_{i=1}^{I^j}\mathbf{x}_i^j.$$

Now, the classification to the class j^* can be accomplished as follows:

$$j^* = \arg\min_{j=1}^{C} ||\mathbf{s} - \mathbf{c}^j||,$$

where $||\cdot||$ is a norm, e.g. Euclidean norm.

The above described linear classifier has several advantages. It has small computational and space requirements, it is easy to implement on various platforms and special teaching algorithms are not needed—it is enough only to have an ideal vector for each context class.

However, for getting the perfect classification result, context classes should be linearly separable which is rarely the case. Therefore, if the separativity of the context classes is highly nonlinear, the Minimum-distance classifier leads to suboptimal results. Detailed analysis of the Minimum-distance classifier can be found in [9].

The Minimum-distance classifier compares the sample to be classified to the class centers and therefore we can identify the following extreme cases:

- A sample is located very near of a class mean vector. In this case, the sample belongs almost surely to the class.
- Distances between a sample and the mean vectors are almost identical. In this case, we cannot distinguish between the classes.

So it seems plausible that we can use the distance as a basis for the confidence evaluation. A distance between a sample $\mathbf{s}$ and the class mean vector $\mathbf{c}^i$ is shown in Eq. 1.

$$d^i = ||\mathbf{s} - \mathbf{c}^i||. \tag{1}$$

It would also be a desirable property to limit the confidence values to the fixed interval, e.g. to the unit interval. This can be accomplished by dividing all the distances by the maximal distance value. The procedure is shown in Eq. 2.

$$\tilde{d^i} = \frac{d^i}{\max_{j=1...C} d^j}. \tag{2}$$

In this paper, we use an absolute difference between the maximal and the minimal scaled distance value as a weighting function. The weight w for the estimate is therefore:

$$w = 1 - \min_{i=1...C} \tilde{d^i}. \tag{3}$$

If the sample $\mathbf{s}$ is very near of a meanvector, corresponding $\tilde{d^i}$ is small and the w gets value near of 1. On the other hand, if the distances d^i are almost identical, $\tilde{d^i}$ is near of 1 and weight w is correspondingly near of 0.

Detailed discussion of the Minimum-distance classifier as a CCR method can be found in [23].

4.3.2 Insincerity

Where several devices obtain information from each other, and they rely on the data for critical functions, it may become very inviting for malicious parties to tamper with the CCR process or misuse the information. Here, we briefly explain a couple of such cases and discuss some solutions to overcome the problems.

The CCR process creates rounds of queries and based on confidences on the collaboratively agreed context and on each device's own subjective idea of the current context. In certain situations the process can be highly unstable and even prone to oscillatory behaviour. Furthermore, this may ultimately lead some devices to "endless" CCR loops. Also, the devices that eagerly try to obtain a proper collaborative context value may eventually drain their batteries. Example of such a situation is when several false parties publish contexts with low confidence values.

Another way to cause harm for innocent mobile devices is to publish wrong context value with high confidence. Again, with suitable relative amount of insincere devices interfering with the CCR process, it can converge to totally wrong context values. In the worst scenario, innocent devices may even perform serious actions and malfunction based on collaboratively agreed faulty values.

According to evaluation with real context data [24] the situations described above are likely to occur, and thus it is very welcome to discuss solutions to avoid them. First of all when dealing with context aware functionality on a mobile device, it is crucial to let a user to be in charge in executing serious actions on a device. Secondly, devices participating in a CCR process should have mechanisms to detect possible extreme cases leading to hazard, and prevent them. This can be accomplished e.g. by regulating the transceiver according to progress of context value's confidence or by regulating the acceptance of the collaborative context value according to similarity to one's own current context value [33].

4.3.3 Properties of communication protocols supporting voting as a CCR method

Collaborative context recognition seeks to mimic combination of human-type perception and decision making. Due to this design philosophy, CCR is quite local, information hungry/query intense, but open and social method, and creates high data traffic with small size individual data transactions.

In addition, response and setup times must be very quick with very low power consumption. Unfortunately, such communication requirements differ quite a bit from established local area mobile communication protocols (such as Bluetooth).

Although the amount of data to be exchanged between devices in the voting protocols is small, each time we want to make a joint context estimation, data have to be broadcasted among the voting devices. In many cases, the devices are battery-driven and therefore continuous communication is quite demanding for such restricted devices. Communication requirements can be divided into two categories:

- Listening to the context estimates and the confidence values coming from other devices
- Broadcasting own context estimates and confidence values

However, based on the confidence information, we can significantly reduce these requirements. Let us consider the first item; if a voter continuously reports low confidence values, it is plausible to assume that the quality of its contextual information is really low. Therefore we can reduce the rate in which we observe the voter. On the other hand, if we are really confident about our own context estimates, we do not necessarily need the information coming from the other voters and we can act in isolation, especially in the case that we are running out of battery. In here we refer back to the beginning of the section; it is rational for a context recognizer to utilize context information coming from other devices only if it can be used to improve context recognition accuracy.

5 Case Study: Physical Activity Recognition

In this section we discuss a realistic CCR example in which several independent context recognizers are located in the same context, doing the same physical activity in this case, and give their estimate on the current context. We start the section by describing the features of the dataset recorded from several physical activities. Then we proceed with test settings and results obtained from the test runs.

5.1 Data Set and Test Settings

In this text, we illustrate CCR methods by using a realistic data set [42]. It is a large data set comprising measurements from several sport activities as well as daily activities such as shopping and eating in a restaurant. In this text we concentrate only on sport activities.

Twelve persons took part in the data recording. Their age range was from 19 to 49 years, with an average of 27 years. The average length of the recordings was almost seven hours. They contain both sessions with pre-arranged activities and free-form sessions in which the test subjects were allowed to freely do activities they were interested in. The pre-arranged activities were annotated by a test supervisor, while the free activities were annotated by the test persons themselves. In this study we focus only a few sport activities and we do not make a difference between two different annotation schemes. The following activities were considered in our study: bicycling, soccer, lying still, nordic walking, rowing, running, sitting, standing, and walking.

5.1.1 Sensor Units

The principal sensor units for this study are two three-channel acceleration sensors and a heart rate monitor. Signals from the acceleration sensors are recorded by using Embla A10, a 19-channel recorder device. Annotations are made by using iPaq device running a custom annotation software. The sensor placement is illustrated in Fig. 2.

5.1.2 Feature Space for the Activity Recognition

In addition to the raw signals from the sensors units, the data set also provides a vast number of features calculated from the raw data. These include both time domain and frequency domain features.

In this study we use 10 features that are selected by using automatic feature selection methods (SFS and SFFS). The list of the selected features and a detailed description of the methods can be found in [24].

5.1.3 Test Settings

Although data recording sessions were carried out for each test subject separately, the environments in the sessions were very similar. We therefore make the artificial assumption that all the recording sessions for the same activity were carried out at the same time. This allows us to use the data for a simulation of group decision making experiment. Transformed into the same time scale, all test persons can be

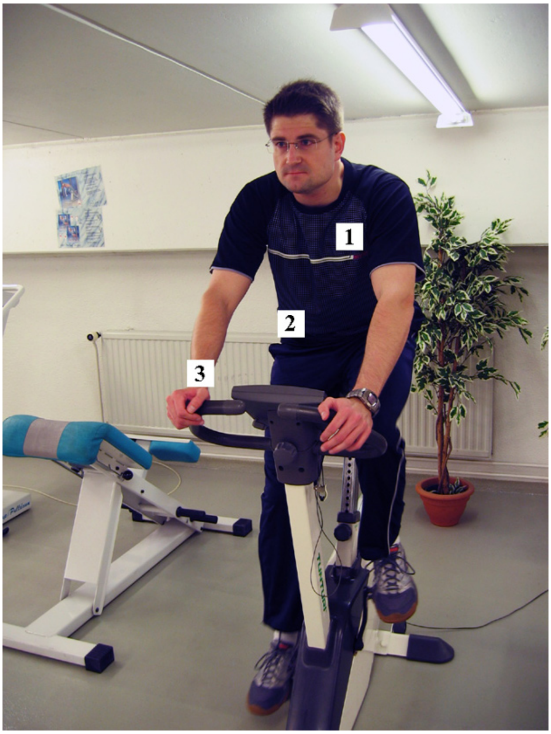

Fig. 2 An image depicting sensor placements during the data recording sessions. 1: Heart rate sensor, 2: Acceleration sensor on hip, 3: Acceleration sensor on wrist.

seen acting as individual decision makers and together they vote on recognition of the current activity.

The group size varied from an individual person to 6 persons in our test runs. The Minimum-distance classifier was trained with remaining 6 persons. For each group a random activity was selected and the group members voted on the activity. In the test runs, the total number of groups was 10000. A clarifying example is shown in Fig. 3 in the case of the group containing three persons. In this example, all persons are performing the same activity and hence the datasets for all persons (implemented as Minimum-distance classifiers) are almost the same. However, small modification took place between the recording sessions, for example small variations in the location of the exercice bike and therefore all the persons see the activity from slightly different “angle”.

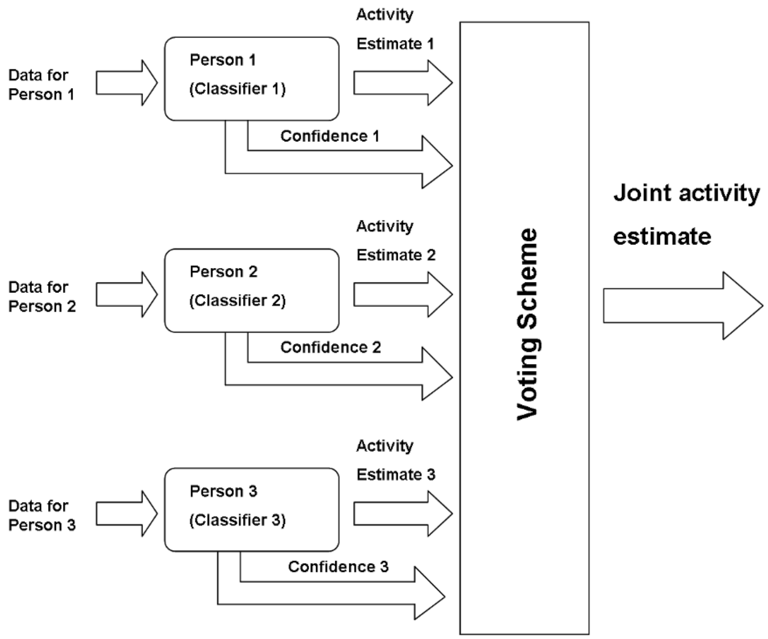

Fig. 3 An example of joint activity estimation in the case of three idependent voters. Possible voting schemes include Borda count, majority voting, and weighted majority voting. Confidence values for weighted majority voting can be computed by using Eq. 3. Classifiers are Minimum-distance classifiers and data sets for different persons are from the same activity but from slightly different "angle".

5.2 Empirical Results

Fig. 4 depicts the joint recognition accuracies. Clearly, the accuracies with all voting mechanisms generally improve when the number of voters increase. The Borda count leads clearly to the worst result. The weighted majority voting leads to smoother curve than plain majority voting, because individual unsure voters do not have significant contributions to the overall recognition accuracy.

From the Fig. 4 it can be seen that individual context recognizers can significantly benefit from using the joint context estimates. Note that in these test runs the pattern classification method was really simple one, Minimum-distance classifier, which can seen as relatively low context recognition accuracies when recognition is done in isolation. However, with 6 recognizer, the accuracy is already over 90%, which can be considered to be very high in real-world applications. This is significant, since our method allows the individual recognizers to be very simple (and thus computation and energy efficient), but together they produce useful results.

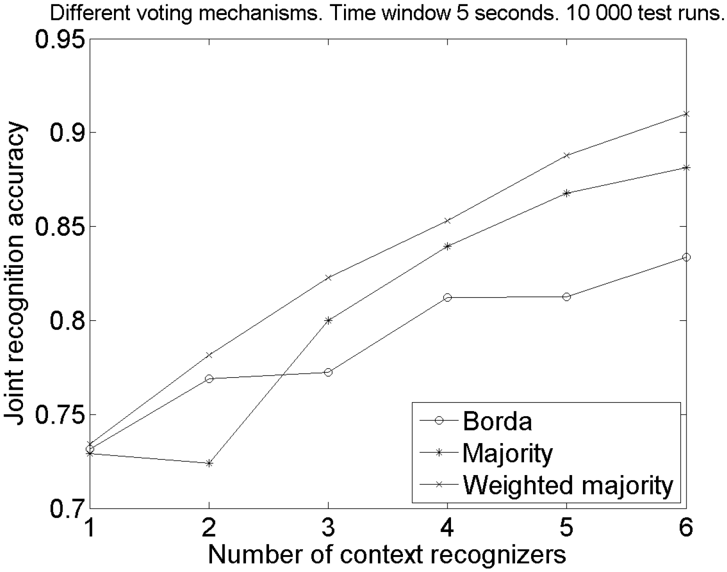

Fig. 4 Joint recognition accuracies with different voting schemes.

6 Context Recognition Platform

For testing and integrating CCR methods with applications, we have implemented a context recognition platform for Symbian S60 mobile phones. The schematic view of the platform is depicted in Fig. 5. The core of the platform is the Minimum-distance classifier and it contains different modules for feature computation and sensory information integration from either built-in sensors or sensors connected via Bluetooth interface.

As the mobile platforms usually have quite limited resources, we offload some of their data to an external server. The information is sent over HTTP encoded into XML messages that can include raw data, computed feature values or context estimates with confidence values. The server side is implemented by using Apache Tomcat-server. The server can have several connections on simultaneously and is therefore able to send information further to other mobile devices. It is also possible to apply more complex data mining algorithms to the data on the server.

The platform has been applied in a number of tests. However, for the CCR method we do not want to rely on servers too much, as in real world settings all the devices participating in a CCR process are likely to be mobile.

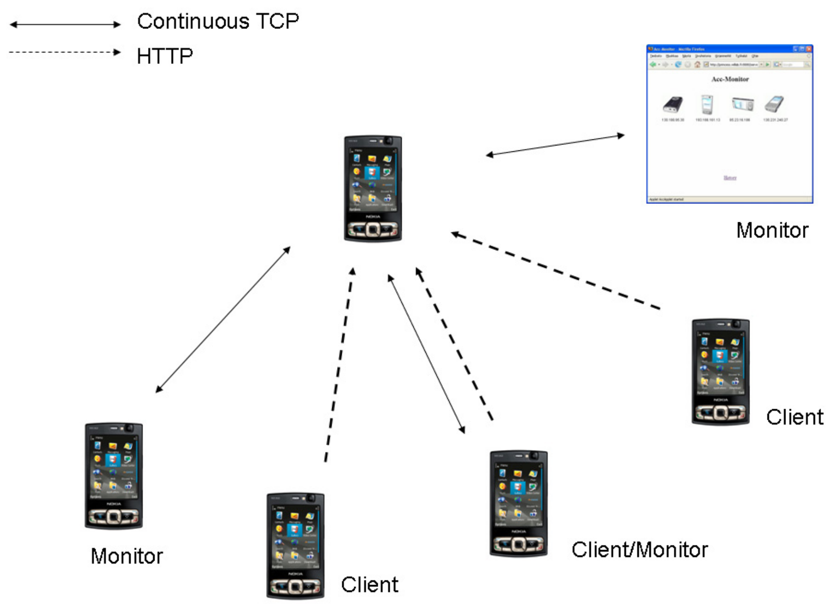

Fig. 5 A schematic view of the context recognizers that run context recognition platform. Application in this small example only shows positions of the other phones connected to the system.

7 The Way Forward

We are running experiments with the context recognition platform in a small scale. However, to test the feasibility of our approach, it will be necessary to deploy the platform in a larger number of devices that are used in everyday situations. Such larger-scale prototyping will undoubtedly give us insight in the feasibility of the CCR method, but also bring light to some of the real world complications. Some of them are already clear from lab tests.

7.1 Energy Savings

Battery consumption is an essential factor of all mobile systems. Many devices go to considerable lengths to save power. For instance, mobile phones sleep a lot: they switch most of their key functions off when not needed, but prepare to bring them up again in a microsecond's notice. The cellular radio transceiver is a major power drain, and therefore cellular protocols are designed so that the receiver can be switched off for relatively large fractions of use time.

Our method is inherently problematic for sleep-mode devices. The communication with nearby devices needs to go on at least periodically, if not constantly. Synchronous communication among the nearby devices would help, but the protocols will be hard to realize for a heterogenous ad-hoc group of devices. Moreover, the background recognition algorithms may take considerable time (and therefore power) to run in case of complex situations with large number of nodes.

We seek to achieve greater power efficiency through better communication protocols, that require less always-on listening, and through optimizing processing such that it can be carried out incrementally and periodically, not constantly. For networked devices, a centralized server can take over much of the work from mobile devices. In a completely mobile ad-hoc system a central server can't be relied upon, however.

Improvements in short-range data links will improve the feasibility of our system. The current generation of Bluetooth transceivers still suffer from very long discovery times, which make them unsuitable for rapidly changing situations (such as walking through a crowd). Also, in some areas the public opinion seems to be currently going against Bluetooth, due to risks that the press are bringing forward.

7.2 Risks

The potential risks of our approach are indeed numerous, if left unaddressed. It's easy to imagine possible ways of attacking the mobile devices. For instance, malicious nodes could deliberately set their context to something improper to cause harm to other nodes' recognition (as explained in Section 4.3.2) . The context data could include spam, just like any other unregulated information channels. More seriously, the context data could be used for various kinds of attacks, including buffer overflows, which in turn could be used for intrusion. Such misdeed will happen as soon as there are financial benefits to be had.

Such risks are by no means unique to our approach, and will have to be addressed for most mobile devices and applications. However, we get some help from the social nature of our domain. Collaborative context recognition is most useful in a shared situation, that is, where a number of people are present. This shared physical presence is key factor to decrease (but not eliminate) mischievous acts. It takes less nerve to attack computers over the internet than people in the same room.

The Big Brother syndrome is a larger concern that our approach shares with many other systems. CCR is fundamentally based on sharing one's context data with other devices. Some of those devices could store and relay context data for dubious purposes. For instance, it will be possible to recognize that certain people have met in a certain place at a certain time, which is something that governments are keen to know (and criminals—imagine a bomb going off when enough people are in the same place). Anonymity techniques could help somewhat, but can't protect personal data against massive data mining. It may also become mandatory to give access to one's personal context store per juridical order. We are expecting a rich set of protection and crypting mechanisms to be conceived for short-range interaction, perhaps based on some variation of PKI (Public-Key Infrastructure).

CCR can be used for many purposes, by many applications. Some are inherently riskier than others. For instance, the archetypal use case for context awareness is to automatically silence your phone's ringtone when you enter a meeting. A CCR-enabled phone would collaborate with other phones in the meeting and notice that most of them have already been silenced, so your phone would do the same. Yet, this application is socially risky. It is humiliating if your phone decides that a wedding is not a meeting and rings in the church. Context recognition is always approximate, and bound to fail often. It can be preferable to use context awareness for applications that are more socially appropriate ("safer"), such as situated reminders, contextual advertising, and games.

These issues—privacy, protection, social appropriateness—have so far been outside our focus. However, they are necessary requirements for real-world systems. We are expecting some solutions from the large body of research and development now going on in the domain of mobile applications in general.

7.3 Knowledge-based CCR?

Context data needs higher level knowledge representation to be widely useful. Ultimately, mobile devices should be able to exchange contexts using a shared set of agreed symbols that convey some semantic meaning. The ongoing work on Semantic Web [6] and related efforts promise to bring shared machine-level vocabularies for distributed systems (such as ours). Yet, generic common-sense understanding of the world [34] is still some way in the future.

Context awareness deals with human life, which is a vast domain with billions of issues to be modeled. We need to confine us to much more limited domains (such as media sharing) to be able to model the knowledge involved. Even then, there are no generic context awareness ontologies agreed upon that devices could use for context collaboration. Until such ontologies are developed, CCR systems need to confine to tags, symbols whose meaning is shared on the application level.

In general, context awareness shares the same problem that plagues artificial intelligence and HCI all over. As soon as a system is able to demonstrate some level of intelligence (more precisely, functions that would appear intelligent if carried out by humans), the users raise their expectations on the system's capabilities, and are then disappointed in the next moment when the system fails to perform intelligently.

As an example, consider the previous example on ringtone silencing. A phone could be using trivial rule-based reasoning in the style of "IF there is a meeting in the calendar AND there are several people in the same room with you THEN silence the ringtone". Given the right situation, this simple rule can work like a charm, and the user is happy. Then, in the church, when the calendar condition fails, the phone ringing puts the user to shame. It will not be immediately obvious to the user why this happened. This simple rule could be understandable to even common people, but any more complex reasoning will remain opaque. It will be necessary to design collaborative context applications such that they fail gracefully: provide reasonable functions even when useful knowledge is absent. If graceful failure is not possible, then at minimum the system should be able to give meaningful error messages, and possibly further explanations for those users who want to know more. Of course, this has not yet been achieved in most systems today.

Acknowledgements This work has been supported by the Academy of Finland and Tekes.

References

[1] Anastasi G, Bandelloni R, Conti M, Delmastro F, Gregori E, Mainetto G (2003) Experimenting an indoor bluetooth-based positioning service. In: Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops (ICDSW 2003), pp 1067–1080
[2] Balakrishnan H, Baliga R, Curtis D, Goraczko M, Miu A, Priyantha B, Smith A, Steele K, Teller S, Wang K (2003) Lessons from developing and deploying the cricket indoor location system. Tech. rep., MIT Computer Science and AI Lab
[3] Bao L, Intille S (2004) Activity recognition from user-annotated acceleration data. In: Proceedings of the Second International Conference on Pervasive Computing, pp 1–17
[4] Beigl M, Krohn A, Zimmer T, Decker C (2004) Typical sensors needed in ubiquitous and pervasive computing. In: in Proceedings of the First International Workshop on Networked Sensing Systems (INSS '04, pp 153–158
[5] Benford S, Magerkurth C, Ljungstrand P (2005) Bridging the physical and digital in pervasive gaming. Commun ACM 48(3):54–57
[6] Berners-Lee T, Hendler J, Lassila O, et al (2001) The Semantic Web. Scientific American 284(5):28–37
[7] Dey A, Abowd G (2000) CybreMinder: A context-aware system for supporting reminders. Lecture notes in computer science pp 172–186
[8] Dey A, Abowd G (2000) The Context Toolkit: Aiding the Development of Context-Aware Applications. Workshop on Software Engineering for Wearable and Pervasive Computing
[9] Duda R, Hart P, Stork D (2001) Pattern Classification, 2nd edn. John Wiley & Sons, Inc.
[10] Farringdon J, Moore AJ, Tilbury N, Church J, Biemond PD (1999) Wearable sensor badge and sensor jacket for context awareness. In: ISWC '99: Proceedings of the 3rd IEEE International Symposium on Wearable Computers, IEEE Computer Society, Washington, DC, USA, p 107
[11] Gellersen H, Schmidt A, Beigl M (2002) Multi-sensor context-awareness in mobile devices and smart artifacts. Mobile Networks and Applications 7(5):341–351

[12] Gemmell J, Bell G, Lueder R, Drucker S, Wong C (2002) Mylifebits: fulfilling the memex vision. In: MULTIMEDIA ’02: Proceedings of the tenth ACM international conference on Multimedia, ACM, New York, NY, USA, pp 235–238
[13] Hallberg J, Nilsson M, Synnes K (2003) Positioning with bluetooth. In: Proceedings of the 10th International Conference on Telecommunications (ICT 2003), vol 2, pp 954–958
[14] Haykin S (1999) Neural networks: a comprehensive foundation, 2nd edn. Prentice Hall
[15] Hazas M, Scott J, Krumm J (2004) Location-aware computing comes of age. Computer 37(2):95–97
[16] Himberg J, Flanagan J, Mantyjarvi J (2003) Towards context awareness using symbol clustering map. Proceedings of the Workshop for Self-Organizing Maps pp 249–254
[17] Hinckley K, Pierce J, Horvitz E, Sinclair M (2005) Foreground and background interaction with sensor-enhanced mobile devices. ACM Trans Comput-Hum Interact 12(1):31–52
[18] Ho J, Intille SS (2005) Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In: CHI ’05: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, New York, NY, USA, pp 909–918
[19] Ito M, Iwaya A, Saito M, Nakanishi K, Matsumiya K, Nakazawa J, Nishio N, Takashio K, Tokuda H (2003) Smart furniture: Improvising ubiquitous hotspot environment. icdcsw 00:248
[20] Jantunen I, Laine H, Huuskonen P, Trossen D, Ermolov V (2008) Smart sensor architecture for mobile-terminal-centric ambient intelligence. Sensors and Actuators A: Physical 142(1):352–360
[21] Jung Y, Persson P, Blom J (2005) DeDe: design and evaluation of a context-enhanced mobile messaging system. Proceedings of the SIGCHI conference on Human factors in computing systems pp 351–360
[22] Knight F, Schwirtz A, Psomadelis F, Baber C, Bristow W, Arvanitis N (2005) The design of the sensvest. Personal Ubiquitous Comput 9(1):6–19
[23] Könönen V, Mäntyjärvi J (2008) Decision making strategies for mobile collaborative context recognition. In: Proceedings of the 4th International Mobile Multimedia Communications Conference (MobiMedia 2008)
[24] Könönen V, Mäntyjärvi J, Similä H, Pärkkä J, Ermes M (2008) A computationally light classification method for mobile wellness platforms. In: Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC’08)
[25] Korhonen I, Bardram J (2004) Guest editorial introduction to the special section on pervasive healthcare. Information Technology in Biomedicine, IEEE Transactions on 8(3):229–234
[26] Korpipää P (2005) Blackboard-based software framework and tool for mobile device context awareness. VTT Publications 579

[27] Korpipää P, Koskinen M, Peltola J, Mäkelä SM, Seppänen T (2003) Bayesian approach to sensor-based context awareness. Personal Ubiquitous Comput 7(2):113–124

[28] Kulkarni P, Ganesan D, Shenoy P, Lu Q (2005) Senseye: a multi-tier camera sensor network. In: MULTIMEDIA 2005: Proceedings of the 13th annual ACM international conference on Multimedia, ACM, New York, NY, USA, pp 229–238

[29] Lam L, Suen S (1997) Application of majority voting to pattern recognition: an analysis of its behavior and performance. IEEE Transaction on Systems, Man and Cybernetics, Part A 27:553–568

[30] Lamming M, Flynn M (1994) Forget-me-not: Intimate computing in support of human memory. Proceedings of FRIEND21 94:2–4

[31] Lehikoinen J, Aaltonen A, Huuskonen P, Salminen I (2007) Personal Content Experience: Managing Digital Life in the Mobile Age. Wiley-Interscience

[32] Liu H, Darabi H, Banerjee P, Liu J (2007) Survey of wireless indoor positioning techniques and systems. IEEE transactions on systems, man, and cybernetics part c: applications and reviews 37(6):1067–1080

[33] Mäntyjärvi J, Himberg J, Huuskonen P (2003) Collaborative context recognition for handheld devices. In: Proceedings of the first IEEE International Conference on Pervasive Computing and Communications (PerCom 2003), pp 161–168

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

[35] Mitchell K, McCaffery D, Metaxas G, Finney J, Schmid S, Scott A (2003) Six in the city: introducing real tournament - a mobile ipv6 based context-aware multiplayer game. In: NetGames '03: Proceedings of the 2nd workshop on Network and system support for games, ACM, New York, NY, USA, pp 91–100

[36] Myerson R (1991) Game Theory: Analysis of Conflict. Harvard University Press

[37] Naaman M (2006) Eyes on the world. Computer 39(10):108–111

[38] Naaman M, Harada S, Wang Q, Garcia-Molina H, Paepcke A (2004) Context data in geo-referenced digital photo collections. In: MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia, ACM, New York, NY, USA, pp 196–203

[39] Nakanishi Y, Tsuji T, Ohyama M, Hakozaki K (2000) Context aware messaging service: A dynamical messaging delivery using location information and schedule information. Personal and Ubiquitous Computing 4(4):221–224

[40] Ojala T, Korhonen J, Aittola M, Ollila M, Koivumäki T, Tähtinen J, Karjaluoto H (2003) SmartRotuaari-Context-aware Mobile Multimedia Services. Proceedings of the 2nd International Conference on Mobile and Ubiquitous Multimedia, Norrköping, Sweden pp 8–19

[41] Oulasvirta A, Tamminen S, Roto V, Kuorelahti J (2005) Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile hci. In: CHI

'05: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, New York, NY, USA, pp 919–928
[42] Pärkkä J, Ermes M, Korpipää P, Mäntyjärvi J, Peltola J, Korhonen I (2006) Activity classification using realistic data from wearable sensors. IEEE Transactions on Information Technology in Biomedicine 10(1)
[43] Pentland AS (2004) Healthwear: Medical technology becomes wearable. Computer 37(5):42–49
[44] Pirttikangas S, Suutala J, Riekki J, Röning J (2003) Footstep identification from pressure signals using hidden markov models. Proceedings of the Finnish Signal Processing Symposium (FINSIG 2003) pp 124–128
[45] Raento M, Oulasvirta A, Petit R, Toivonen H (2005) Contextphone: A prototyping platform for context-aware mobile applications. IEEE Pervasive Computing 4(2)
[46] Ramachandran V, Oberman L (2006) Broken mirrors: A theory of autism. Scientific American 295(55):63–69
[47] Randell C, Muller H (2000) Context awareness by analysing accelerometer data. The Fourth International Symposium on Wearable Computers 1(1):175–176
[48] Salber D, Dey A, Orr R, Abowd G (1999) Designing for ubiquitous computing: A case study in context sensing. Graphics, Visualization, and Usability Center Technical Report 99 29:99–29
[49] Sandholm T (1999) Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence, MIT Press, chap Distributed Rational Decision Making, pp 201–258
[50] Schmidt A, Takaluoma A, Mäntyjärvi J (2000) Context-Aware telephony over WAP. Personal and Ubiquitous Computing 4(4):225–229
[51] Sorvari A, Jalkanen J, Jokela R, Black A, Koli K, Moberg M, Keinonen T (2004) Usability issues in utilizing context metadata in content management of mobile devices. Proceedings of the third Nordic conference on Human-computer interaction pp 357–363
[52] Starner T, Schiele B, Pentland A (1998) Visual contextual awareness in wearable computing. In: Proceedings of the Second International Symposium on Wearable Computers (ISWC 1998), pp 50–57
[53] Toyama K, Logan R, Roseway A (2003) Geographic location tags on digital images. Proceedings of the eleventh ACM international conference on Multimedia pp 156–166
[54] Van Laerhoven K, Schmidt A, Gellersen H (2002) Multi-sensor context aware clothing. Wearable Computers, 2002(ISWC 2002) Proceedings Sixth International Symposium on pp 49–56
[55] Viana W, Filho J, Gensel J, Oliver M, Martin H (2007) PhotoMap-Automatic Spatiotemporal Annotation for Mobile Photos. Lecture notes in computer science 4857:187
[56] Zhang J (2003) Evolution of the human ASPM gene, a major determinant of brain size. Genetics 165:2063–2070