

# Vehicular Mobility Prediction by Bayesian Networks

Kari R. Schougaard\*  
Center for Pervasive Computing  
University of Aarhus, Denmark  
kari@daimi.au.dk

## Abstract

*In mobile and ubiquitous computing the location of devices is often important both for the behavior of the applications and for communication and other middleware functionality. Mobility prediction enables proactively dealing with changes in location dependent functionality. In this project Bayesian networks' ability to reason on the basis of incomplete or inaccurate information is powering mobility prediction based on a map of the street grid and the current location and direction of the vehicle. We found that it is feasible to divide information of a map into smaller parts and generate a Bayesian network for each of these in order to make mobility prediction based on localized information. This makes the information stored in the Bayesian networks more manageable in size, which is important for resource constrained devices. Common sense knowledge of how vehicle moves is feeded into the networks and enables them to make a good prediction even when no information of the vehicles mobility history is used. Experiments on real world data show that in an area statically divided into hexagonal cells of 200m in diameter, we get 80.54% accuracy when using localized Bayesian networks to predict which cell a vehicle enters next.*

## 1 Introduction

The future location of mobile devices is often of interest in a ubiquitous computing environment. It may, for example, be used on various levels in a communication protocol for mobile ad-hoc networks. Such networks consist of mobile nodes that communicate without using centralized access points or fixed infrastructure. Nodes move in to and out of range of other nodes as communication ranges are limited - especially for resource-constrained devices. The network will, therefore, have a dynamically changing topology. Mobility prediction lets topology changes be foreseen.

This allows choice of more stable routes in the routing protocol. Furthermore, mobility prediction enables proactive route discovery and resource reservation, which may be a step towards real-time message delivery. Finally, prediction of the future location of nodes may motivate suspicion of link failure and may consequently speed up the detection of link failure and network divisions.

On a higher level, applications using context information will often benefit from a prediction of location changes as location is a basic context information and other types of context often depend on the location. When a location change is anticipated, services that depend on the current location may be closed down gracefully, or possibly downloaded for local use.

Previously, mobility prediction has been based upon a variety of artificial reasoning about the mobility history of the mobile device, for example neural networks [8] and compression techniques [2, 11]. Bayesian techniques have been used in [9, 10, 1]. Other approaches use recent location and velocity data in combination with a mobility model such as linear movements, random way point, or Gauss-Markov [5, 6]. To our knowledge no previous model, that makes predictions on a fine grained scale in an outdoor setting, have been tested on a larger amount of real world data. Furthermore, none of the projects using Bayesian techniques incorporate a commonsense knowledge of movements that enables the technique to achieve good predictions without adapting to the history of the users.

In the present study, an outdoor area of around 20 times 30 kilometers is divided statically into hexagonal cells of 200m in diameter and we predict which cell a vehicle is going to enter next. The prediction is based on the output of Bayesian networks. The Bayesian networks use input of the current location and direction of the vehicle, information about possible routes that is extracted from a street map, and common sense knowledge of how vehicles move on streets. In order to get a fast prediction the complexity of each network is lessened by using several networks each covering a separate physical area. The performance of the networks is tested on real world data from 20 different

---

\*This work was partly supported by MiNEMA, ESF

cars driving for between 10 and 54 days in the Danish city Aalborg. The networks are not adapted by feeding the mobility history of the vehicles into the networks. Therefore, the prediction accuracy obtained in the experiments pictures mobility prediction when the vehicle diverges from its habitual paths.

The next section presents related work. Section 3 then introduces Bayesian networks and argues for their use in mobility prediction. Thereafter, Section 4 explains the structure of the Bayesian networks we have used for mobility prediction. A short presentation of the implementation techniques follows in Section 5, and in Section 6 the experiments and their results are outlined. Finally we outline future work, and close with a conclusion.

## 2 Related work

A dynamic Bayesian network was used for indoor mobility prediction in [10]. The prediction was tested in an office environment with 15 rooms on mobility traces of 4 users in two periods of respectively 2 and 5 weeks. A symbolic location model with rooms constituting locations was used. The potential uncertainty of which room the user is placed in is not discussed in the paper, and probably the uncertainty is very small compared to the uncertainty faced when mapping a GPS position to a street section. The Bayesian network used in the project is correspondingly simple; the 15 locations are easily represented in one network, and the current location is simply represented in one node. They compare the prediction of next room based on current room, to predictions where a mobility history of up to 5 rooms is used and predictions where the time of day and the weekday are included in the network. Interestingly, this extra information does not unequivocally improve the predictions. Their experiments show that it gives better results to use trained networks. The prediction accuracy varies between 35 and 90% in the various setups. No prediction is made the first time a specific combination of inputs appears, which is the opposite approach of ours: we set up common sense rules for movement predictions and did not adapt the networks to the mobility history of the vehicles.

In [8] a project where mobility prediction is used for home automation is presented. A neural network predicts which rooms are going to be occupied based on the entire history of location changes and of noise, door and window status. It is not discussed in the paper how the pattern detection in the neural network would scale to a large outdoor environment with a huge number of locations, nor is it clear which kind of secondary input, such as noise, would be interesting in the outdoor setting.

Lempel-Ziv compression is used for indoor mobility prediction in [11] and for outdoor mobility prediction in [2]. In the indoor case rooms and corridors or other doors between

rooms are represented as symbols, and in the outdoor case cells in a cellular network are represented as symbols. The compression technique builds a trie (a multi-tree) of the mobility history, which is then used for predictions according to the number of times a sequence of symbols was found in the past. In the outdoor prediction case time is added as another factor and a symbol is repeated when longer time is spent in the corresponding location. The problem in scaling such an approach is that the amount of data needed in order to get reliable predictions grows with the number of symbols in the dictionary.

In [9] a Bayesian model, implemented as a particle filter, was used for mobility prediction in an outdoor scenario. Their primary goal was to infer high-level behaviour, defined as changes in transportation mode, from the location information, but the accuracy of prediction of future location is also assessed. The project resembles ours in that outdoor mobility is predicted in a larger area. The prediction is done using a filtering technique which means the movements are described as a mathematical function, where uncertainty is handled by using random variables. The choice of an approximative particle filter over a Kalman filter means the estimated posteriors over the state space are not restricted to a unimodal distribution. It might also make it easier to handle the large number of states of the variables. To evaluate the prediction they present the prediction accuracy in terms of prediction length in number of city blocks. Usually there are only one street surrounding a city block, sometimes with a smaller street or two crossing through it. In our scenario the vehicle will cross 0 to 5 junctions before entering the next cell. In their adapted model the prediction descends from 100% to 52% for prediction 1 to 5 city blocks ahead. But because of the difference in unit of measurement it is difficult to compare that to our results. They do not state the location prediction accuracy for the unadopted model. The amount of test data is another difference between the projects: They evaluate the model on a dataset chosen from a log of one person's movements. The 29 episodes that cover 12 hours of log are chosen such that they contain a change in transportation mode, and in the figure visualizing the training data they seem simple.

Markov models are used in [1] to predict future locations. A method for automatic identification of important, meaningful locations is presented. These locations are entered in a Markov model, which may be used for prediction. In this project a plethora of GPS positions spread over a large area are reduced to a handful or two of locations. This makes the number of locations entered in the predictive model manageable in size. The possible applications using prediction of transitions between important locations are different from the applications that would use prediction of next cell. It would be interesting to combine the two approaches to a predictive model that uses both fine grained

and coarse grained prediction.

In for example [5, 6] mobility patterns are described using a mobility model, and the use of mobility prediction for different uses is analysed. The predictions are then tested against simulations, where the mobile entity moves according to the used mobility model. It is not clear how these mobility models relate to real movement patterns.

### 3 Motivation

Mobility prediction involves estimating an unknown given some observations. We have prior knowledge about how the unknown is connected to the observable phenomena. But both the observations and the connections are uncertain. We must, therefore, turn to probability theory in order to reason about the future location of mobile devices. It enables us to work with statements such as “if the device is on road  $Y$  the probability of getting a GPS reading in area  $X$  is 0.7” or “if the device is on road  $Y$ , the probability of getting a direction reading of north is 0.2”. Bayes’ rule:  $P(Y|X) = P(X|Y)P(Y)/P(X)$  gives us the means of reasoning in the opposite direction of the known posterior probabilities. This enables us to answer questions such as: “what is the probability of being on road  $Y$  if we get a GPS reading in area  $X$  and the device is going north”.

Bayesian networks<sup>1</sup> is a formalization of the human way of reasoning using propositional logic in combination with uncertain events. The definition of a Bayesian network given by Jensen [4] is: A Bayesian network consists of the following:

- A set of *variables* and a set of *directed edges* between variables.
- Each variable has a finite set of mutually exclusive states.
- The variables together with the directed edges form a *directed acyclic graph*. (A directed graph is *acyclic* if there is no directed path  $A_1 \rightarrow \dots \rightarrow A_n$  s.t.  $A_1 = A_n$ .)
- To each variable  $A$  with parents  $B_1, \dots, B_n$ , there is attached the potential table  $P(A|B_1, \dots, B_n)$ .

Note that if  $A$  has no parents, then the table reduces to unconditional probabilities  $P(A)$ .

In Bayesian networks the assertion of independence of variables is expressed in the topology of the network: A variable should be independent of its predecessors given its parents. Variable independence allows the full joint distribution, which would otherwise be used to find the probability of a variable’s value, to be broken up into much

<sup>1</sup>Bayesian networks are also called belief networks, probabilistic networks, causal networks, and knowledge maps

smaller pieces. When comparing the number of probabilities needed for a full joint distribution table and the number needed for a Bayesian network with the same variables, we often see an exponential reduction.

### 4 Location Prediction Design

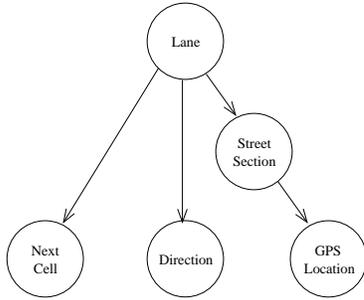
The prediction model is envisioned to be used for outdoor applications for vehicles. Therefore, no natural boundaries to the area exist, with the exception of the boundaries of the island or continent, where no car ferries cross. The solution is to let each vehicle use a Bayesian network for each cell; the prediction will be based on the network for the cell where the vehicle is currently located. Thus, the current cell as an entity is not represented in the network. In this project hexagonal cells with a diameter of 200m are used.

The actual location of the vehicle may be different from the measured location. This means there is some uncertainty regarding in which cell the vehicle is currently located, at least when it is close to a border between cells. Therefore, which network to choose for prediction is also to some extent uncertain. This may be countered by making the area the Bayesian network covers larger than a cell, corresponding to a satisfying amount of errors of location measurements. For now the mobility prediction is based on the cell the logged GPS coordinate is placed in, even though this might give some problems with determining on which street the vehicle is placed.

A *minimal* Bayesian network must reflect the causal structure of the problem it models [4], which is a good reason to strive to represent causal relations between variables. This is done by making the cause variable a parent of the effect variable, and is visually represented as an arrow from the cause variable to the effect variable. Often children are placed below parents. We reason about which cell is entered in the future. As causal relations proceed in time, the variable we want to determine is in the bottom of the network. This is contrary to the usual case, where the Bayesian network supports belief in a hypothesis that causes certain observable variables to be in certain states.

A Bayesian network for prediction of the next cell based on current location and direction and on the structure of the street network is presented in Figure 1. Below we will argue for the structure of the network and explain how the probability tables of the nodes are produced.

The two variables *Lane* and *Street Section* are closely connected. Each state of *Street Section* represents a straight section of the street, its ends are either intersections or points inserted to approximate curved streets by straight street sections. The possible states of the variable are extracted from the digital map. A state of *Lane* represents one direction of a state of *Street Section*. Each state of *Lane*



**Figure 1. Bayesian network for mobility prediction**

may cover several lanes as all lanes in the same direction on larger streets are covered by one state. The potential table of *Lane* is very simple: every state of the *Lane* variable has the same probability. The potential table of *Street Section* states the probability of being on a certain street section, given the lane. Given the lane we know the street section, so for each of the lane states, only one street section state has a non-zero entry. This is also an argument for that *Street Section* is independent of its siblings given its parent.

The child of *Street Section* is called *GPS Location*. It represents the observations of the location and supports the belief about which street section the vehicle is located on. Location is, when defined as an x and a y coordinate, a continuous variable. When the position is based on GPS, the accuracy of the GPS device gives a possible interval but this interval is still much too fine-grained to be represented directly in the network. Thus we need to define intervals of GPS coordinates. They should be as big as possible in order to make the probability table as small as possible, and it should be easy to determine whether a GPS position falls inside the area. The approach we have chosen is to define circles around junctions and rectangles around street sections. Many GPS receivers have an accuracy of for example 5 meters, but with the shading buildings and reflected signals common in an urban setting, that kind of accuracy is not obtained on actual measurements. To take the measurement inaccuracies into account we have defined two zones of 10 and 40 meters around street sections and of 15 and 57 meters around junctions, in the following called *primary zones*. The longer distances to junctions are chosen because there is some uncertainty in the map; the junctions are not necessarily exactly at the position stated in the map. A GPS reading inside a circle is interpreted as there being an equal chance of the vehicle being on either of the streets of the junction. We would like for a circle around a junction to cover the overlap between the rectangles around the street sections that are part of the junction. When the distances are chosen as stated, the overlaps between primary zones

around street sections of the junction are covered by the circle around the junction, when the angle between the street sections is  $90^\circ$  or more. The primary zones of different street sections and junctions often overlap. In order to get mutually exclusive states, as is required for Bayesian networks [4], an overlap between two or more primary zones will define yet another zone. The exception to this is that a small distance zone around a junction takes precedence over a small distance zone of a street section that is part of the junction. When the vehicle is placed in the overlap of two or more primary zones, it should be understood as there being a high or medium probability of it being on either of the corresponding street sections.

The potential table of *GPS Location* states the probabilities of getting a GPS reading in a zone given that the vehicle is on a certain street section. In an ad-hoc fashion the probability of being inside a small primary zone is set to 0.7 (the GPS reading is less than 10 or 15 meters away from the actual location), the probability of being inside a large primary zone is set to 0.25 (an error of between 10 and 40 meters, or between 15 and 57), and the last 0.05 accounts for when the GPS reading is more than 40 or 57 meters away from the actual location. When the primary zones overlap with each other the sub zones thus created get a probability corresponding to their area, but such that, for example, the summed probabilities of all zones inside the small primary zone equals 0.7. The area of a zone is not calculated accurately, but on the basis of a count of how many “pixels” with 1 meter distance the zone contains.

Driving in a lane will cause the car to move in a certain direction. The variable *Direction* is therefore a child of *Lane*. The states of *Direction* are the eight major directions: N, S, E, W, NE, NW, SE and SW. This is a coarse grained division of direction into states, but given the lane there is actually only two directions that are possible for the vehicle to move in: up and down the street. To get a fast change in direction, when the vehicle changes its direction only the last three GPS position readings are used to determine the direction. This means that there are much uncertainty in the direction input because the GPS receiver uses its internal direction calculations to correct the GPS position before it outputs the GPS position. This means that the log of GPS positions tend to make soft turns rather than sharp ones, especially when the reception conditions are not ideal. Therefore it will not reflect the information available to let the direction input influence the belief of which lane we are placed on in a critical manner.

The potential table of *Direction* again reflects the large amount of uncertainty in the direction readings. If the direction matches exactly with one of the major directions, this direction has a probability of 0.5, the neighboring directions a probability of 0.2 and the directions perpendicular to the calculated a probability of 0.05. If the direction is

between mayor directions the two closest directions have a probability of 0.4 and the following a probability of 0.1.

The variable *Next Cell* is also a child of *Lane*. Its states are the six adjacent cells of the current cell. Being in a certain lane will sometimes cause the vehicle to enter a new cell with a high probability. When there is a junction ahead, there may be several possibilities and the probabilities are based on common sense knowledge about how cars move: that they drive on a path in the street network, that they tend to choose larger roads with higher speed limits, that they move purposefully (most of the time) and thus do not often turn around and go in the opposite direction.

To make the potential table all possible routes starting in the given lane and leaving the cell without returning to an earlier point are generated. To this purpose the digital map is represented as a graph. Junctions correspond to nodes and an edge denotes a street section between junctions. A standard search algorithm is then used to find possible routes on the map. The probability of each route is calculated as explained below, the probability of the most probable route to a certain end lane is chosen, and the probabilities of going to any of the lanes entering each of the adjacent cells is summed. To get the probability of a route the following rules for attaching a probability of choosing each of the possible lanes in a junction are observed:

- With weight 0.15: The probability of taking a certain lane in a junction is linearly dependent on the angle between the current lane and the next. An angle of  $0^\circ$  has a probability of 1 and an angle of  $180^\circ$  a probability of 0.
- With weight 0.35: The probability of taking a certain lane is linearly dependent on the angle between the next lane and the general direction of the path. The general direction of the path is calculated from the future route. To simplify the feedback into the network the history of current movements is not taken into account.
- With a weight of 0.5: Speed limits of the possible lanes are compared. Streets with a speed limit of 30 km/hour is given a weight of 10 and else the weight is given as speed limit minus 20. The probability is found by a normalizing the numbers thus obtained.

As is obvious there are many ad hoc choices lying underneath these probabilities. Another possibility would be to give all possible streets of a junction the same probability. This would waive the idea of using the expectation of purposefulness to predict the future movements. A better support for the probabilities could be obtained by analyzing traces of moving vehicles. Even if the probabilities do not exactly mirror the standard behaviour of drivers, they result

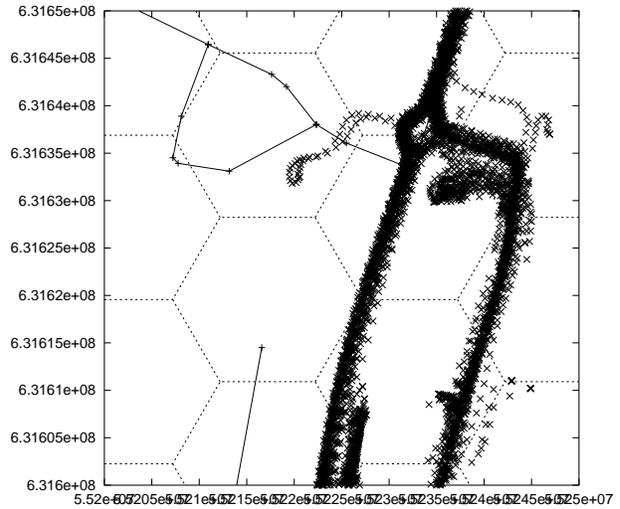


Figure 2. Map detail with data of one car

in differences in belief of which cell is entered next, which follows common sense knowledge of how people drive.

## 5 Implementation

A program that parses a digital map of Aalborg city and generates the the Bayesian networks for mobility prediction according to the principles explained above has been implemented using Java. The Bayesian network tool Netica is used for implementation and inference of the networks. It compiles Bayesian networks into a junction tree of cliques for probabilistic reasoning.

## 6 Experiments

The test data consists of logs of GPS data collected by 20 cars driving for between 10 and 54 days in the city of Aalborg, Denmark [3]. A detail of the map and data from one car is shown in Figure 2. The data contains location and velocity data collected at one second intervals. This amounts to each car giving between 14291 and 264721 entries, corresponding to between 4 and 73 hours of driving, with an average 26 hours. The data has been made anonymous by removing all data that falls within a  $2\text{km} \times 2\text{km}$  square, where the center of the square is less than 1km from the residence of the driver(s). The data set is divided into portions that have no missing GPS entries for more than 5 seconds, and all of the data is used.

The data of the digital street map was provided by Cowi. The data was created some time before the test data was collected. This means there will be some discrepancies between the map data and the street network the cars actually traveled on as street networks evolve over time. The data

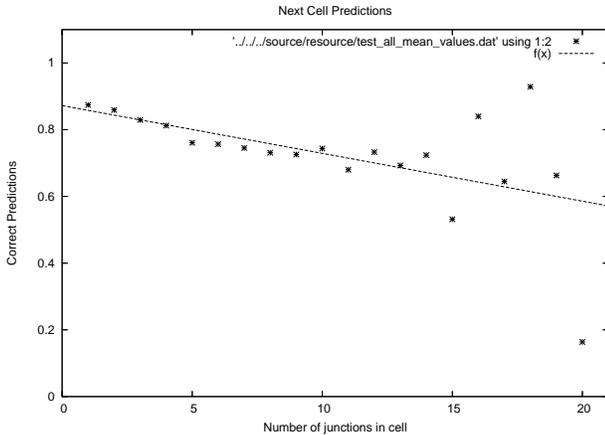


Figure 3. Next Cell Prediction

# junctions	# cells	# junctions	# cells
1	796	11	35
2	481	12	39
3	354	13	26
4	278	14	15
5	215	15	13
6	168	16	2
7	136	17	2
8	113	18	2
9	79	19	2
10	54	20	1

Figure 4. Number of cells with data

contains endpoints for street sections where the endpoints are inserted either in order to approximate a curved street or because there is an intersection with another street.

A prediction has been made for each GPS log entry. Of 893,638 predictions 80.54% were correct. In Figure 3 the mean values for predictions are depicted as a function of the number of junctions in the cell. When the data is fitted to a linear function, the result is:  $f(x) = -0.014 * x + 0.87$ . In Fig. 4, the number of cells with predictions is given by the number of junctions in the cells. As is seen in Fig. 4, the cars drove through only a limited number of cells with more than 15 junctions. The large variation in prediction for cells with many junctions is probably due to this.

The number of residents in Aalborg municipality was around 160,000 at the time the data was collected. For larger cities the number of cells with a medium density of junctions will be larger, but the number of cells with the highest density will probably not be much larger as these are centered around the old part of town. Because of the higher number of cells with a medium number of cells in larger cities, mobility prediction performance will be some-

what inferior here, while it will be better in smaller towns.

## 7 Future Work

### 7.1 Adaptation

The structure of the Bayesian networks are supposedly the same for all users. But different users will have different probabilities for example for which lanes (ie. street section and direction) they use most. The probability of which cell is entered next, given the lane, is another part of the networks where there are differences between users. It is an interesting point to investigate the effect of using an adaptation technique [4] to adapt the networks to different users by learning from the user history.

For some adaptations it might also be rewarding to adapt the networks to the mobility history of all vehicles in the population. In related work both strategies are used, see [5, 6] and [8]. We plan to test the difference of performance between predictions in a non-adapting network, a network adapted only to the vehicle which owns the network and a network adapted to data from the whole population of vehicles. There is both an information dissemination issue and a privacy issue in using mobility data from other vehicles to adapt the Bayesian networks.

### 7.2 Complex Mobility Patterns

The mobility model presented in this paper models which cell the car is moving into from the present street section, using an input of the current direction. It is possible to base the prediction on more complex movement patterns. If personal habitual driving routes are represented in the network, the Bayesian network would be able to distinguish between two or more different routes with one or more lanes in common. Other factors such as weekday or the time of the day may also support prediction based on complex mobility patterns.

### 7.3 Time of Entering Next Cell

A valuable complement to information about which cell the vehicle is going to enter is *when* it is going to enter it. A model of traffic flow, which is widely accepted in the traffic management community [7] might be used for this. The length of the street interval is equal to the distance that a single vehicle travels in one time step at free-flow speed. A street interval could for example be 13.9 meters long, which corresponds to movement for one second at the speed limit of 50 km/h which is normal in many urban areas. In the traffic management community formulas describing traffic

movements are researched. Among other things they incorporate free-flow speed and congestion factor. Such a formula could be used to estimate the time of entering the next cell.

## 8 Conclusion

We wrote a program that from the input of a digital map generates Bayesian networks for mobility prediction. We generated Bayesian networks for an area of around 20 times 30 kilometers. The area was divided into hexagonal cells of 200m in diameter, and a Bayesian network was generated for each of these cells. Prediction of which cell was entered next was tested on GPS logs from 20 different cars driving for between 4 and 73 hours; of 893,638 predictions 80.54% were correct.

The result establishes that mobility prediction may be based on localized information. This means that the total area the prediction is made for need not be bounded. Furthermore, the result confirms the use of statistical tools, such as Bayesian networks, for mobility prediction, that has earlier been supported by simulations and small-scale real-world tests. Finally, common sense knowledge of vehicle movements was feeded into the network, and the high accuracy of predictions supports the possibility of using mobility prediction even when diverging from one's habitual paths.

## Acknowledgments

I want to thank the ESF network MiNEMA, who sponsored my stay with the Distributed System Group at Trinity College Dublin, where the idea for the project was fostered. During my stay, I had many fruitful discussions with Prof. Vinny Cahill concerning the use of Bayesian Networks for outdoor mobility prediction, and the idea of using several networks was formed. I'm also grateful for discussions with prof. Erik Ernst, University of Aarhus, of many details.

Cowi A/S provided me with the digital map that made it possible to use the mobility data gathered during the INFATI project conducted by Aalborg University in cooperation with Sven Allan Jensen A/S and M-TEC A/S.

## References

- [1] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal Ubiquitous Comput.*, 7(5):275–286, 2003.
- [2] A. Bhattacharya and S. K. Das. Lezi-update: An information-theoretic approach to track mobile users in PCS networks. In *Mobile Computing and Networking*, pages 1–12, 1999.
- [3] C. S. Jensen, H. Lahrmann, S. Pakalnis, and J. Runge. The infati data. Technical Report TR-79, TimeCenter, 2004.

- [4] F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer, 2001.
- [5] S.-J. Lee, W. Su, and M. Gerla. Wireless ad hoc multicast routing with mobility prediction. *Mobile Networks and Applications*, 6(4):351–360, 2001.
- [6] B. Liang and Z. J. Haas. Predictive distance-based mobility management for pcs networks. In *Proceedings of the 18th IEEE INFOCOM*, 1999.
- [7] H. Lo and A. Chow. Control strategies for oversaturated traffic. *Transportation Engineering*, 130(4), 2004.
- [8] M. C. Mozer. Lessons from an adaptive house. In D. Cook and R. Das, editors, *Smart environments: Technologies, protocols, and applications*. J. Wiley & Sons, 2004.
- [9] D. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In A. K. Dey, A. Schmidt, and J. F. McCarthy, editors, *UbiComp 2003: Ubiquitous Computing, 5th International Conference, Seattle, WA, USA, October 12-15, 2003, Proceedings*, volume 2864 of *Lecture Notes in Computer Science*. Springer, 2003.
- [10] J. Petzold, A. Pietzowski, F. Bagci, W. Trumler, and T. Ungerer. Prediction of indoor movements using bayesian networks. In T. Strang and C. Linnhoff-Popien, editors, *Lecture Notes in Computer Science*, volume 3479, pages 211–222. SpringerLink, 2005.
- [11] A. Roy, S. Bhaumik, A. Bhattacharya, K. Basu, D. Cook, and S. Das. Location aware resource management in smart homes, 2003. In First IEEE International Conference on Pervasive Computing and Communications (PerCom'03).