In: Mobile Multimedia: Communication Engineering Perspective ISBN 0000000000
Editors: Ismail Khalil Ibrahim, David Taniar, pp. 1-16

Chapter 1

CREATE PERVASIVE MULTIMEDIA APPLICATIONS USING MOBILITY AWARENESS

Frederic Lassabe, Philippe Canalda, Pascal Chatonnay, Francois Spies *
Laboratoire d'Informatique de l'Université de Franche-Comté
Damien Charlet †
INRIA Rocquencourt, ARLES project-team

Abstract

Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract

Keywords: Mobility, Multimedia.

*E-mail address: surname.name@pu-pm.univ-fcomte.fr
†E-mail address: damien.charlet@inria.fr

1 Introduction

+ paradigm : when middleware components, such as Wireless Indoor Positioning or mobility prediction, render pertinent the arrival of Lite Deployment of Best-Effort-like Mobile Multimedia Applications. + illustration and definitions of interoparable components which + participate to mobility awareness : - Positioning - tracking/journalizing - mobility prediction + interaction with other middleware components which proceed the continuity of multimedia flows + summary

2 Positioning, Logging, Journalizing and Tracking

2.1 State of the art

In this section, we focus on the WiFi indoor positioning systems. Indeed, outdoor positioning is easely achieved with good accuracy by the Global Positioning System (GPS). On the other hand, indoor positioning systems are currently being investigated. First, we present an infrared-based positioning system, giving the idea of the most simple WiFi positioning system. Second, we describe some projects based on data collection of signal strength measurements. Third, after exposing a new trilateration algorithm, we present several positioning systems based on trilateration. Fourth, we present a positioning system based on the hybridation of the data collection-based and the trilateration approaches.

A trivial way to determine an approximate mobile terminal's position is based on infrared sensors [**?**]. An infrared sensor having a very short range and a transmission capability only with line of sight, placing infrared sensors at critical points (building entrances, point of interest, etc.) allows to punctually know the position of a mobile terminal. Another coarse-grained positioning system is to approximate the mobile terminal's position as being the same as that of its access point.

In the RADAR project [1], a signal strength map is used to position the mobile terminals. A database containing points with known coordinates is built. For each point in the database, the geographical coordinates and the signal strength measured from each access point are stored. The signal strength map is established either by measurements or by computation following a radio wave propagation model. Positioning a mobile requires to measure the signal strength at an unknown point. By comparing the measurement with the ones in the database, one can deduce its position. The error median of RADAR's accuracy is between 2 and 3 meters.

Other projects use measurement sets to position mobile terminals. For example, statistical approaches, [2] and [3], use signal strength distribution on reference points to locate a mobile terminal. A project used the neural network approach [4] to determine a mobile terminal location. These projects are greatly inspired by the RADAR project.

Another way to determine a mobile's position is trilateration. It determines the position of a point, whose coordinates are unknown, by using the distances towards reference points. The reference points are points whose position is known at each moment. In a WiFi network, the first step, in order to position a mobile terminal by trilateration, is the distance computation. Indeed, knowing the distances between the mobile terminal and the reference points.

Several methods to solve the trilateration exist. The first one is a geometrical resolution based on the Pythagore theorem. The second one is based on analytical resolution. As the computed distance is rarely exact, the analytical and geometrical resolutions raise problems. The third approach is an iterative one [5], allowing to solve the problems due to distances miscalculation.

Knowing a point P (WiFi access point), its coordinates and its distance to the mobile terminal to locate, we consider a circle (respectively a sphere) from the plan (respectively the space) centered on the point P whose radius equals the distance to the point the coordinates of which are unknown. In figure 1, the 3 circles of respective centers C_1, C_2 and C_3 admit point M as an intersection. The ray of each circle is the distance between its center and the mobile terminal. Point M is the position of the mobile terminal obtained by trilateration. Dotted circles present the realistic situation, when the distances are not exact. In such cases, the mobile terminal is likely to be located in the shaded area.

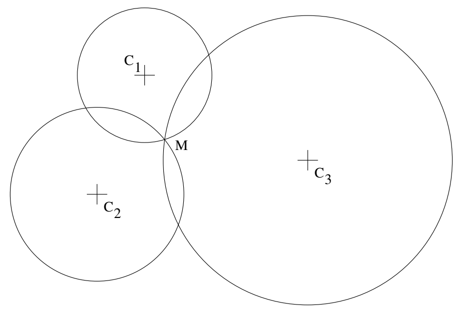


Figure 1: Trilateration principle.

The iterative approach works as follows (fig. 2.1): a grid of points regularly layed out is used. For each point, the greatest distance towards the circles' perimeters is computed. The position of the mobile terminal is the point for which the greatest distance is the smallest.

Many projects base the distance computation on the relation between the transmitter-receiver distance and the signal strength measured. The signal strength is the power received by the receiver. In the sequel, when writing *signal strength from a device*, we mean the signal strength corresponding to the signal sent by this device and received by another WiFi device.

In the SNAP-WPS project [6], measurements of the signal strength from the access points are carried out. The measurements are carried out at coordinates measured. Thus, the distances between the measurement points and the access points are known. The data collected are used in a third degree polynomial regression. The resulting polynomial expression is used to compute the distances corresponding to each signal strength measured. According to its authors, the accuracy of the SNAP-WPS is about 1 to 3 meters.

Interlink Networks' (IN) approach [7] uses an alternative to the Friis equation [8] to take the indoor obstacles into account. The Friis equation is the following:

$$\frac{P_R}{P_T} = G_R G_T \left(\frac{\lambda}{4\pi d}\right)^2$$

where:

- P_R and P_T are respectively the signal strength measured at receiver and the signal strength transmitted by the transmitter,
- G_R and G_T are the antenna gains of respectively the receiver and the transmitter,
- λ is the wavelength of the radio signal,
- d is the distance between the receiver and the transmitter.

It is achieved by changing the Friis exponent, currently 2, applied on the distance. After studying several buildings, the authors of this system have decided to use an exponent equal to 3.5. According to its authors, the IN's system has an accuracy of about 3 meters. However, points exist, where the precision is bad. Building topology heterogeneity explain these points.

In [5], we present the *Friis-based Calibrated Model* (FBCM) as an improvement of the Interlink Networks' approach. In the FBCM, the exponent replacing the quare of the distance is calibrated. The calibration is achieved by a small set of measurements carried out at various places in the building. Considering the exponent as the unknown in the Friis equation, and knowing the distance d and the signal strength P_R, a new exponent is computed for each measurement point. The mean value of the whole set of exponents is used in the alternative to the Friis equation. The accuracy of the FBCM is about 15 meters. This result will be discussed later.

The best results, with an error of about 1 meter, are obtained by an hybrid approach. This is the *FBCM and RADAR-based Hybrid Model* (FRBHM) [9]. It combines the deterministic reference points approach to the FBCM. Thus, the errors of the FBCM due to the heterogeneity of the topology are reduced by restraining the search field to a small area thanks to the reference points approach. Then, a Friis exponent is computed within the area. It fits better the topology and improves the accuracy.

Determining the distances between the mobile terminal and points whose coordinates are known can also be achieved by using two different waves. For example, the cricket compass [**?**] system uses the time difference between the arrival of an ultrasound signal and a radio signal. Knowing the ultrasound wave speed, computation of $D = V \times T$ gives the distance towards the beacon. By deploying numerous (hundreds) of radio/ultrasound beacons, the accuracy of trilateration is improved.

Add carpetlan and its results.

2.2 Analysis

From the works exposed above, we can identify two main families: one is based on discrete approaches whereas the other one is based on continuous approaches. We discuss and analyse the discrete approaches first and then the continuous ones. For both families, we expose their positioning systems' strength and flaws in terms of:

- cost (time/money/computation/memory),
- scalability,
- accuracy,

- refresh rate.

The whole of these criteria are critical within the scope of the deployment of an indoor positioning system.

Discrete approaches

The discrete approaches allow to determine a mobile terminal's position among a set of positions. In the Active Badge system and the access point-based positioning system, the positions possible are defined in the set of the infrared emitters' coordinates in the case of Active Badge and in the set of the access points coordinates in the case of the access point-based system. Both these systems are quick to deploy because they do not implie heavy computation nor lot of data to be stored. However, their accuracy if limited to the number and distance between either the IR transmitters or the access points. The Active Badge system can achieve an accuracy of several meters by deploying a great number of IR devices. The counterpart to the accuracy is the cost in terms of money to buy all the IR devices and in terms of time in order to set up every IR device. The access point-based positioning technique is scalable because WiFi AP have an indoor range up to 30 meters. Several AP can cover an area of thousands of square meters. Moreover, the access points are also used a network infrastructure and their cost has fallen with the spreading of the WiFi technology. The IR-based positioning system scales not much because a lot of devices are required to cover great areas.

The reference points-based positioning systems [1, 2, 3] have a good accuracy with an about 3-meters error. They are scalable but the time required to setup such a system grows with the growth of the deployment area. Indeed, the bigger the area, the more reference points are needed. The expected refresh rate of the position depends on the number of mobile terminals and the software structure of the measurement component. If client-centric, the measurements are transmitted from the client to the positioning server which returns the client's location. It requires only two messages. If infrastructure-centric, the measurements are performed by the access points after a positioning packet which is broadcasted. The AP send the measurements to the positioning server, which aggregate the measurements to compare them to the database of reference points and sends the position back to the client. It generates more messages and use of the radio channels therefore it is less scalable than client-centric measurements. The cost in computation and memory is not negligible but most of the computation is done by the positioning server. Thus, it is not expensive to deploy such a positioning server and its infrastructure.

The important drawback of reference points-based positioning systems is the time required to perform the measurements for all reference points. Moreover, is an access point is moved, removed or added, and if the topology changes (e.g a wall is destroyed in the building), the offline measurements need to be made again.

Continuous approaches

The continuous positioning systems allow to determine precisely the position of a mobile terminal based on mathematical computation. In particular, the use of trilateration allows to compute a mobile terminal's position in real plan or space.

Positioning system	Avg error (m)	Standard deviation (m)
Interlink Networks' PS	29.38	12.17
SNAP-WPS	22.78	14.07
FBCM	15.86	9.34
RADAR+true distances	4.32	2.23
FRBHM	1.07	0.32

Table 1: Accuracy of the positioning models.

The SNAP-WPS, the IN's positioning system and the FBCM position a mobile terminal the same way: when the distance between the APs and the mobile terminal are known, a trilateration computation is performed and returns the mobile terminal's position. The IN's positioning system is the most scalable and quick to setup. It is just needed to run the positioning program on the client to position oneself. Therefore, it is fully dynamical, can be scaled on every WiFi network and is cheap. Memory comsumption and computational cost are also small because of the use of trilateration. Compared to the IN's approach, both the FBCM and SNAP-WPS are longer to setup. They need data to be calibrated before being used. They are consequently a bit less scalable and a bit more expensive in terms of time. However, their accuracy is better than that of the IN's positioning system. Each of these positioning systems has a fast refresh rate, going down to 0.5 second.

Hybrid approach

The FRBHM combines the FBCM to a RADAR-like approach, requiring an offline data collection. The combination to the FBCM reduces the amount of data necessary to obtain the same accuracy. In fact, the FBCM part of the FRBHM makes it more accurate than any reference points-based positioning system. Indeed, the reference points-based approaches cannot determine exactly the position of a mobile terminal located at coordinates not included in the reference points database. Being an hybrid model, its strength are between those of the discrete approaches and those of the continuous approaches. However, its accuracy is the best observed, particularly when dealing with the experiments.

2.3 Experiments

In this section, we present the experimentations carried out on the positioning systems we have presented presented. The whole of the tests takes place in the same building. For each point we test, the location computation is performed. The tests are conducted the following way: the true coordinates and the signal strength measured are provided to the positioning system. It computes the distances to the access points. The positioning systems tested are Interlink Networks [7], SNAP-WPS [6], the FBCM, the FRBHM and RADAR with computation of the true deistances between the reference points.The true coordinates allow to compute the positioning error. The locations according to each method and the corresponding errors are displayed.

The calibrated model accuracy is in most cases better than the accuracy of the models presented by Interlink Networks [7] and SNAP-WPS [6]. The points where the accuracy of the calibrated model is less than that of the others projects are located behind heavy obstacles (ie. a load-bearing wall or in a stairwell). These locations are not part of the calibration area, which explains the lack of precision.

The absolute error made by the calibrated model on the ground floor is easy to explain. Indeed, the calibration was carried out on the first floor. The use on the ground floor creates huge distance miscalculation because the topology is different and the access points are not relatively located at the same coordinates. But the calibrated model remains competitive with the other positioning models. A quick overview of the results raises a question. How can the errors made by the three models be explained ? The errors are great compared to the building size. We came to the conclusion of topology influence on the positioning system.

It is easy to explain this fact mathematically. Let us consider an access point and two points of the plan at an equal distance d of the access point. A point has a line of sight with the access point. The other point is hidden by a wall. On each point, the signal strength measured or theoretical is different. It shows that the relation between the distance and the signal strength is not bijective. Therefore, this relation does not admit a reciprocal relation. The reciprocal expressions of the Friis equation and its alternatives are approximate compared to the real conditions. Therefore, it can be concluded that, the more heterogenous the topology, the more inaccurate using a reciprocal expression based on the Friis equation is.

3 Mobility Prediction

3.1 State of the art

While not much research has been done in th field of mobility prediction, much have been done within the scope of web page prefetching in order to improve the WWW latency. Suc works are interesting because they can be applied to mobility prediction. In fact, formal models such as Markov Models (MM) can be used to model many stochastic data, from users surfing the web to mobile terminals moving from cell to cell.

In this section, we present the works related to prediction. As all the works are not initially meant to be applied to mobility prediction, we will call this comportemantal prediction. The related works are divided in three parts: the first one is the short-term prediction, which is not really our concern within the scope of mobile multimedia and handoff anticipation. But it is worth being mentionned, in particular concerning outdoor mobile terminals and ad hoc routing. The second part is the middle term prediction, which is of great interest in the field of handoff anticipation. The third and last part is the long term, considering people's strong habits, like going home each evening.

Short-term

The mobility prediction can be achieved by trajectory computation [10], allowing to predict short-term movements based on the speed vector of an object. As WiFi positioning is not sufficiently robust, various methods allow to smooth the mobile's trajectory, such as Kalman filters [11] and double exponential smoothing [12]. However, all of these methods require fine-grained positioning to be used. Therefore, they are mostly suited to virtual reality devices than to WiFi mobile positioning.

In [13], the authors present a hierarchical mobility model. The local mobility model uses trajectory computation to make short-term prediction whereas the global mobility model considers user mobility patterns (UMP) and makes middle-term prediction by matching the UMP with the user's actual path (UAP). Merging those models allows to determine next cell crossing with accuracy and increases the matching of the UAP with the UMPs.

Middle-term

Sarukkai [?] proposes to model users path by Markov chains (Markov Model). He foresees four goals with users' path modeling:

- generate a surfing tour on a web site,
- predict pages accesses in order to prefetch the pages and improve the WWW latency,
- advice some links based upon others users' habits comparison with the current user's surfing path,
- identify the hubs.

The second item is interesting within the scope of mobility prediction. Many models proposed in the field of web page prefetching can be applied to mobility prediction. The MM described by Sarukkai is defined as (S, A, λ) where S is the space of states (URL, HTTP request or action such as email, database update, etc.), A is the transition probabilities matrix and λ is the initial state distribution. Given n states, the matrix A is an $n \times n$ matrix. The learning of the model is based upon log files which are processed to identify users' sessions. Then, counting the number of incoming transitions and the number of occurrences of each outgoing transition allows to determine the transition probability for each one. Two methods use the MM to predict next page access: the first considers which page is being visited and considers the most probable page in the MM as being the next page. The second method tries to use a longer path history by computing ${}^{T}S(t) = a_0 I(t-1)A + a_1 I(t-2)A^2 + ... + a_{j-1} I(t-j)A^j + ... + a_{k-1} I(t-k)A^k$ where:

- A is the transition probabilities matrix,
- a_{i-1} is the weigth of the i^{th} history state, $\sum_{l=0}^{k-1} a_l = 1$,
- $I(t-i)$ is the state vector of the history path at time $t-i$.

The vector S(t) gives the probabilities for each state to be the next state in the user's surfing path. The experiments carried out on the MM achieve an accuracy between 60 and 70%. Although it is an acceptable accuracy, considering longer path history can improve the overall accuracy of a MM-based predictive model.

Pirolli et al. [?] extend the MM to model longer users' paths. It is achived by considering various lengths of N-grams. An N-gram is a t-uple $(X_1, X_2, ..., X_N)$ with X_i being a web page. The N-grams extracted from the log files allow to build a MM whose states are labeled with $k = N - 1$ pages visited by the user. The N^{th} page of the N-gram is used to train the model. Such a MM is called a Kth-order Markov Model (KMM). In [14], Pitkow et al. give an extension of the KMM. It is called the All Kth-order Markov Model

(AKMM). It consists in building the KMM from 1^{st}-order to k^{th} order. Then, the next state is predicted by the highest KMM in which the current state exists. The higher order a MM, the less chances to match a state we have. Thus, the AKMM addresses this problem while still allowing predictions with the highest possible degree of history. Although the AKMM have great predictive power, they consume much memory to be built and stored. Therefore, several methods intend to reduce the size of the AKMM while not losing too much predictive accuracy.

In [14], J. Pitkow et al. describe how to mine the *longest repeating sequences* (LRS) in users paths. The method consists in identifying in users path the subsequences which follow these criteria:

- subsequences are composed of a set of consecutive items,
- they must be repeated T times, with T typically equal to 1,
- at least once, the LRS is the longest repeating (ie. it is not part of a longest LRS).

Mining the LRS allows to significantly reduce the amount of data before building the AKMM required to predict pages visits while not losing accuracy. In fact, accuracy marginally decreases but it is neglectable compared to the space gain.

In [15], the authors extend the use of the All K^{th}-order MM by pruning some states to reduce the size of the automata. Their model's name is *Selective Markov Model* (SMM). The states are pruned when they do not carry much sense. A state has not much sense either when occuring not frequently or when having few difference between the probabilities on its outgoing transitions or when having a bad error rate during the validation step. The SMM is an approach similar to the LRS one. It allows to drastically reduce the amount of data while staying close to the full MM's accuracy.

An approach close to the MM one is based on Hypertext Probabilistic Grammar (HPG) [**?**]. In a HPG, pages are non-terminal symbols, states S and F (beginning and ending of a page sequence) are terminal symbols and the production rules are links between pages. The transition probabilities are computed following the same formula than in a MM excepted for the transitions from S to any state. In the initial state distribution, a parameter α. α parameters the weight given to a page occurrence as the first of a sequence compared to the weight given to pages anywhere in a sequence. The probability to start at state E_i (ie. going from S to E_i) is computed as follows: $\pi(E_i) = \frac{\alpha Occ(E_i)}{N_T} + \frac{(1-\alpha)Start(E_i)}{N_{seq}}$ where

- $\pi(E_i)$ is the probability to go from state S to state E_i,
- $Occ(E_i)$ are the occurrences of state E_i within all the sequences,
- N_T is the number of transitions in all the sequences,
- $Start(E_i)$ are the occurrences of state E_i being the first state of a sequence,
- N_{seq} is the number of sequences in the log files.

Following this computation, a state never being the first in the log files still has a probability to be the first in a further sequence. The first formal representation of the HPG is an automata. In [**?**], the authors extend the HPG by modeling it with MM. Then, the MM is used to compute patterns. Patterns are subsequences whose probability to occur is greater than a threshold $\lambda = \theta\delta$ where θ is the support threshold and δ is the confidence threshold. According to the threshold, a tree of all the patterns is built. This method is useful to determine frequently used patterns and can be used as a base to the mobility prediction by keeping only data that are frequent enough to achieve an accurate prediction.

In [**?**], the accuracy between the 1^{st}-order MM and the 2^{nd}-order MM is compared in order to decide whether or not a state is cloned to improve the 1^{st}-order MM's accuracy. It aims at improving the 1^{st}-order MM's accuracy while not using as much space as a All 2^{nd}-order MM. The decision to clone a state is bound to the classification of a state as being inaccurate. Let $p_{i,k,j}$ be the probability of state j following states i then k. Let $p_{k,j}$ be the probability of state j following state k. Let O be the number of outgoing transitions from state x and I be the number of incoming transitions to state x. If $\forall$ 1¡=i¡=I, 1¡=o¡=O, $-\gamma < p_{i,x,o} - p_{x,o} < \gamma$, the state is accurate. If not, the state is cloned. As a result, the accuracy of the 1^{st}-order MM is improved (slightly lower than the 2^{nd}-order MM) and the number of states in the model is lower than the number of states in the 2^{nd}-order MM.

Hidden Markov Model (HMM) [16] can be used to determine the mobile's future position after a learning phase. The tutorial [17] of L. R. Rabiner defines an HMM as being composed of

- a set of physical states,
- a set of observation symbols to be matched with the set of physical states,
- A, the state transition probability distribution,
- B, the observation symbol probability distribution whitch matches the observation symbols with the physical states,
- π, the initial state distribution.

The compact notation for the HMM is $\lambda = (A, B, \pi)$. The problem to be solved with the HMM and when predicting the next movement of a mobile terminal is: given the observation sequence $O_1O_2...O_T$, with O_i in the set of observation symbols, and a model λ, how do we select a corresponding state sequence $q_1q_2...q_T$, with q_i in the set of physical states, which is optimal according to the current problem's criteria ? Using a Viterbi algorithm, the problem can be solved. The use of the HMM takes into account the errors in the observation of the mobile terminals' location. Indeed, in most of the indoor positioning systems, the accuracy is such as the observation can not be considered as real. The observations are matched against the reality according to a stochastic process. In such case, the physical states are the real locations of the mobile terminals and the observation symbols are the coordinates computed by the positioning system. The probabilities of the transitions from one state to another compose the second stochastic process. The HMM has predictive power but does not take into account path history longer than 1.

The KMM and AKMM can be extended to the mobility prediction. Instead of prefetching the most probable state, several states can be prefetched. To trigger a prefetch, the *K-past* and *K-to-1 past* models [18] use a prefetch threshold. K-past is a KMM and K-to-1 past is an AKMM. Considering the actual state, the K-past triggers a prefetch for each state of the model having a probability greater or equal to the threshold. The K-to-1 past works the same way, but similarly to the AKMM, the prediction is made by the highest order model containing the current state. In some cases, increasing the K-to-1 past's order can decrease the accuracy of the model. This will be discussed in details in section 3.2. It is therefore necessary to address this problem. The K-to-1 past* model [18] creates a set composed of the union of the results for each k-past with 1¡=k¡=K. It improves the accuracy but increases the cost of the prefetch by triggering a handoff in more cells.

Long-term

Something ?

3.2 Analysis

The base analysis of the state of tha art being mage, there are some points which require to be underlined. First, why does someone want to take into account longer path history ? An example can easely prove this assertion. Given states A, B, C, D, E, F, G, H and users' paths $ABCD$, $ABCE$ and $EBCD$, if we observe a current user's path beginning with ABC, what should be the next state ? The 1^{st} and 2^{nd} orders MM will compute a 33% probability to go to state E and a 67% probability to go to state D. According to the users' paths, if someone follows states ABC, there is a 50% probability to go to state D and a 50% probability to go to state E, which is given by the 3^{rd} order MM.

On the other hand, long path history is not always more accurate and can lead to prediction errors. In the case of the k-to-1 past model, learning with the users' paths $ABCD$, $ABCE$, $ABCF$ and $HBCG$, will give the 1^{st} to 3^{rd} orders MM exposed in figure 2.

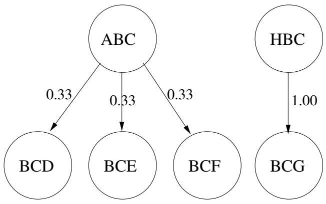
Let the threshold be 0.2. Considering the new peregrination, ABC, then G (to be predicted by the model). The prediction algorithm begins trying to find potential transitions in the 3-rd MM. Here, a 3^{rd} order state ABC exists. It has three outgoing transitions, each occuring with a 0.33 probability. The algorithm returns the three target states: BCD, BCE and BCF, meaning to trigger a handoff in states D, E and F. The mobile going in G, the prediction is wrong. The prediction with 2^{nd} order MM would return states D, E, F and G, which is more correct.

That is why, although requiring more cells to be prepared for a handoff, the K-to-1 past* model is more efficient than the K-past and K-to-1 past ones.

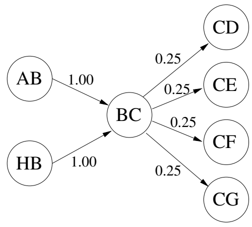
3.3 Experiments

In this section, experiments of the K-past, K-to-1 past and K-to-1 past* models are presented. Tests were carried out on data sets provided by the **Institut fur Pervasive Computing**[1]. A first data set is the *Augsburg Indoor Location Tracking Benchmarks*, built with data from 4 persons clicking their location in a building each time they change their room. Each entry of the log includes the room and the unix time when the user entered the room. The second data set is the *Nokia Context Data*, built from GSM data, including unix times and cell location.

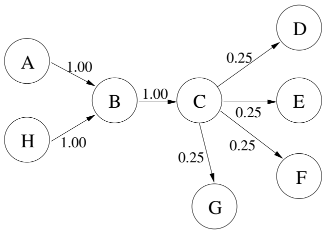
[1] http://www.pervasive.jku.at/Research/Context_Database/index.php



(a) 3^{rd} MM



(b) 2^{nd} MM



(c) 1^{st} MM

```
If possible, apply (a)
else
  If possible, apply (b)
  else
    apply (c)
  endif
endif
```

(d) All-3^{rd} MM algorithm

Figure 2: An All-3-th Markov Model Sample.

In the Augsburg Location Tracking, there are two series of log files. The *_fall were built during the autumn, and the *_summer were built during the summer. The prefix a, b, c and d identify the 4 users. Table 2 presents the test of the 1^{st} order MM on the Augsburg Location Tracking data set. It consists in separated tests for each file in the package and one test on the overall. The overall test is carried out by concatenating all the files together. For each test, the training is made considering the first 75% peregrinations of the log file. The remaining 25% are used to test the training. The results show two files whose accuracy is less or equal to 50%. In both cases, it is explained by the size of the log file, which contains few peregrinations. Thus, the training is not exhaustive enough, leading the model to make wrong predictions. Globally, the results are good because the training is made on the same user's profile than the test. It proves the influence of the context within the scope of the mobility prediction. The lower accuracy observed in the overall test is logical. Indeed, the concatenation of the files into another one involves that the system uses the 4 user's fall profiles to train before confronting the resulting model to the user's movements in summer. As the users may have different habits over different periods, the accuracy is lower.

Tables 3 and table 4 expose the K-past's accuracy on the Nokia Context Data and the Augsburg Location Tracking. Several conclusions are drawn from the results:

- the lower the probability threshold, the better the accuracy is. Indeed, decreasing the threshold increases the number of states selected for the handoff, increasing the success probability,

Log file	(learning/validation percent)	1-past
a_fall	(75/25)	0.7841
b_fall	(75/25)	0.8131
c_fall	(75/25)	0.6592
d_fall	(75/25)	0.7127
a_summer	(75/25)	0.5000
b_summer	(75/25)	0.7857
c_summer	(75/25)	0.7375
d_summer	(75/25)	0.3333
Overall	a_fall to c_fall (100/0)	0.6470
	d_fall (94/6), *_summer (0/100)	

Table 2: Mobility prediction accuracy, 1^{st} order Markov Model, Augsburg files.

Threshold	1-past	2-past	3-past	4-past
1.0	0.0694	**0.2729**	0.2705	0.2077
0.9	0.1632	**0.3551**	0.3430	0.2754
0.8	0.2882	**0.4589**	0.4324	0.3527
0.7	0.2899	**0.5459**	0.4831	0.3792
0.6	0.3368	**0.5676**	0.4976	0.3816
0.5	0.5191	**0.6618**	0.5507	0.4082
0.4	0.6458	**0.6667**	0.5556	0.4082
0.3	**0.7396**	0.7198	0.5749	0.4251
0.2	**0.7951**	0.7560	0.5942	0.4251
0.1	**0.9080**	0.8213	0.6135	0.4324
0.0	**0.9583**	0.8430	0.6256	0.4444

Table 3: K-past Mobility prediction accuracy, Nokia Context Data.

- 1-past accuracy is better than the other's ones below a given threshold. As the 1-past training gives the most possible outgoing transitions from one state, the transitions have less probabilities to occur than the ones in the 2+ past models: the 1-past model has therefore a lower accuracy with high thresholds but becomes the best with low thresholds,

- for $K > 1$, the K-past model's accuracy decreases when K increases. Indeed, increasing K increases the path size stored in each state of a model. The longer the past, the lower is the probability to observe the same path in further peregrinations, especially when not having much training data, thus reducing the overall accuracy of the model. There could be some cases whose precision would grow with the increase of the automaton's degree, for example in an environment which users always follow long paths crossing from time to time,

- for several thresholds, the accuracy remains stable and increases suddenly when low-

Threshold	1-past	2-past	3-past	4-past	10-past
1.0	0.1719	0.1255	**0.2937**	0.2862	0.0688
0.9	0.4758	**0.4814**	0.4526	0.4359	0.1078
0.8	**0.4879**	0.4814	0.4563	0.4638	0.1143
0.7	0.4879	0.4814	0.4563	**0.4926**	0.1217
0.6	0.4879	0.4814	0.4563	**0.5344**	0.1236
0.5	0.4879	0.4851	0.4600	**0.5530**	0.1320
0.4	0.4879	0.5279	0.5074	**0.5967**	0.1366
0.3	0.4879	0.6190	0.5864	**0.6478**	0.1561
0.2	0.6097	0.6887	0.6571	**0.7026**	0.1757
0.1	**0.8457**	0.8429	0.8123	0.7667	0.1914
0.0	**0.9898**	0.9665	0.9210	0.8411	0.1961

Table 4: K-past Mobility prediction accuracy, Augsburg Location Tracking.

ering the threshold. It is due to transitions frequently taken. Such a transition has a high probability to occur, whereas the other have a really small probability. This explains that the success of the prediction suddenly grows for a low threshold, while it chosed only one possible state for higher thresholds. Such cases occur when users have strong habits in their movements, such as going to one's own office.

The K-past model's results are interesting: the lowering of the accuracy we observe when taking into account longer paths is not the goal of a mobility prediction model. That is why other models where tested.

In table 5, the results of the K-to-1 past model on the Nokia Context Data are given. Increasing the K-to-1 past degree does not always increase the accuracy. Given the explanation exposed in section 3.2, it is not really surprising. Finally, we observe that the K-to-1 past model has better accuracy than the K-past one. This obsevation is obviously logical as the K-to-1 past model selects at least the same transitions than the K-past model.

Threshold	1-past	2-past	2-to-1 past	3-past	3-to-1 past
1.0	0.0694	0.2729	0.2923	0.2705	**0.3357**
0.9	0.1632	0.3551	0.3744	0.3430	**0.4082**
0.8	0.2882	0.4589	0.4855	0.4324	**0.5097**
0.7	0.2899	0.5459	**0.5725**	0.4831	0.5652
0.6	0.3368	0.5676	**0.5894**	0.4976	0.5797
0.5	0.5191	0.6618	**0.7029**	0.5507	0.6667
0.4	0.6458	0.6667	**0.7150**	0.5556	0.6715
0.3	0.7396	0.7198	**0.7681**	0.5749	0.7005
0.2	0.7951	0.7560	**0.8092**	0.5942	0.7303
0.1	0.9080	0.8213	**0.8816**	0.6135	0.7778
0.0	**0.9583**	0.8430	0.9203	0.6256	0.7923

Table 5: Mobility prediction accuracy with K-to-1 past, Nokia Context Data.

The last test presented concerns the K-to-1 past*. This model returns the union of the selections from each K-past model's degree. Tables 6 and 7 respectively show the results with the Nokia Context Data and the Augsburg Location Tracking data set. It underlines the increase of the accuracy when increasing the K-to-1 past* model's order. The union of the predictions made by the kMM with k going from 1 to K increases the accuracy of the model of approximately 14 % for the 3-to-1 past*.

Threshold	1-past	2-to-1 past	3-to-1 past	12-to-1 past
1.0	0.0694	0.2483	0.3142	**0.5017**
0.9	0.1632	0.3663	0.4080	**0.5590**
0.8	0.2882	0.5243	0.5503	**0.6545**
0.7	0.2899	0.5660	0.6042	**0.6875**
0.6	0.3368	0.6441	0.6632	**0.7222**
0.5	0.5191	0.7066	0.7274	**0.7882**
0.4	0.6458	0.7535	0.7830	**0.8229**
0.3	0.7396	0.8299	0.8490	**0.8715**
0.2	0.7951	0.8628	0.8802	**0.8906**
0.1	0.9080	0.9288	0.9288	**0.9358**
0.0	**0.9583**	**0.9583**	**0.9583**	**0.9583**

Table 6: Mobility prediction accuracy with K-to-1 past*, Nokia Context Data.

Threshold	1-past	2-to-1 past	3-to-1 past
1.0	0.1719	0.1766	**0.3615**
0.9	0.4758	0.4879	**0.4879**
0.8	0.4879	0.4879	**0.4879**
0.7	0.4879	0.4879	**0.4879**
0.6	0.4879	0.4879	**0.4879**
0.5	0.4879	0.4916	**0.4916**
0.4	0.4879	0.5344	**0.5409**
0.3	0.4879	0.6255	**0.6255**
0.2	0.6097	0.7054	**0.7082**
0.1	0.8457	0.8866	**0.8931**
0.0	**0.9898**	**0.9898**	**0.9898**

Table 7: Mobility prediction accuracy with K-to-1 past*, Augsburg Location Tracking.

Two main phenomena are also noticeable: accuracy is best for a threshold equal to zero. When the threshold equals zero, the prediction is done by selecting every outgoing transition in each order. Another interesting observation is that the accuracy when the threshold equals zero is the same for every order of the K-to-1 past*. In fact, the first degree model contains every possible transition from one state to another one. Thus, each transition of a higher order MM exists in the 1^{st} order MM. As the higher degrees models do not contain every transition in the 1^{st} order MM, their probabilities are higher. When the threshold equals zero, the 1^{st} order MM makes a prediction as accurate as the higher orders models' one. Due to the same properties, the accuracy increases better for a high threshold. When the threshold is 1, the first degree model has a very poor accuracy whereas the increase by using the second degree and third degree models is important.

4 Interactions with middleware ensuring Multimedia communication

continuity + context : - Multimedia communications in parallel with services (localization, prediction of mobility or services, ...) - multi-dimensional sensitiveness : positioning, mobility, network, profile user, adaptive contents + Typical infrastructure addressing the continuity of mobility multimedia services - mixer : adaptive streaming QoS, multi-flow switches, ... - caches : abstraction layer, performing a match between mobile client positioning and contents of some physical network nodes - handover : how to install virtual continuity over station based networks - congestion management, low-consumption management, ... + Interfacing various middleware to achieve simultaneously localization and continuity - examples of predictive cache management and predictive horizontal handoff - experimentations + future trends ad hoc routing, etc.

5 experiments

+ Experimental studies + New arriving multimedia applications - characteristics : wireless, heterogeneous topology, lite-infrastructure, concurrent communications and services, rich medias, - typical applications rich and geo-based multimedia information systems

6 Conclusion and future trends

The conclusion

References

[1] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000.

[2] M.A. Youssef, A. Agrawala, A.U. Shankar, and S.H. Noh. A probabilistic clustering-based indoor location determination system. Technical report, Maryland Information and Network Dynamics Laboratory, March 2002.

[3] Z. Xiang, S. Song, J. Chen, H. Wang, J. Huang, and X. Gao. A wireless lan-based indoor positioning technology. *IBM Technical Journals*, 48(5), 2004.

[4] R. Battiti, A. Villani, and T. Le Nhat. Neural network models for intelligent networks: deriving the location from signal patterns, 2002.

[5] F. Lassabe, D. Charlet, P. Canalda, P. Chatonnay, and F. Spies. Friis and Iterative Trilateration Based WiFi Devices Tracking. In *Proceedings of the 14th Euromicro Conf. on Parallel, Distributed and Network-based Processing (PDP 2006)*, Februar 2006. to be published.

[6] Y. Wang, X. Jia, and H.K Lee. An indoors wireless positioning system based on wireless local area network infrastructure. In *6th Int. Symp. on Satellite Navigation Technology Including Mobile Positioning & Location Services*, number paper 54, Melbourne, July 2003. CD-ROM proc.

[7] Interlink Networks, Inc. A practical approach to identifying and tracking unauthorized 802.11 cards and access points. Technical report, 2002.

[8] L.V. Blake. *Radar Range-Performance Analysis*. Artech House Radar Library, December 1986.

[9] F. Lassabe, Ph. Canalda, D. Charlet, P. Chatonnay, and F. Spies. Refining WiFi indoor positionning renders pertinent deploying location-based multimedia guide. In *Procs of IEEE Int. Workshop on Pervasive Computing and Ad Hoc Communications (PCAC06), in conjunction with the IEEE 20th Int. Conf. on Advanced Information Networking and Applications (AINA06)*, pages ***–***, Vienna, Austria, April 2006. 6 pages, to appear.

[10] S.J. Lee, W. Su, and M. Gerla. Mobility prediction in wireless networks. In *Proceedings of IEEE ICCCN*, pages 22–25, October 2000.

[11] R.G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, Inc., 1997.

[12] Joseph J. LaViola Jr. Double exponential smoothing: An alternative to kalman filter-based predictive tracking. In *Proceedings of the workshop on Virtual environments 2003*, pages 199–206. ACM Press, 2003.

[13] T. Liu, P. Bahl, and I. Chlamtac. Mobility modeling, location tracking and trajectory prediction in wireless atm networks. *IEEE Journal on Selected Areas in Communications*, 16(6):922–936, August 1998.

[14] J. Pitkow and P. Pirolli. Mining longest repeating subsequences to predict world wide web surfing. In *Proceedings of USITS'99: The 2nd USENIX Symposium on Internet Technologies and Systems*, volume V, pages X–Y. USENIX, USENIX, October 1999.

[15] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *ACM Transactions on Internet Technology (TOIT)*, 4(2):163–184, May 2004.

[16] J.M. Francois, G. Leduc, and S. Martin. Learning movement patterns in mobile networks : a generic method. In *European Wireless 2004*, pages 128–134, February 2004.

[17] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. volume 77, pages 257–286. IEEE, February 1989.

[18] F. Lassabe, D. Charlet, P. Canalda, P. Chatonnay, and F. Spies. Predictive mobility models based on kth markov models. In *Proceedings of the IEEE International Conference on Pervasive Services (ICPS'06)*, June 2006. Article submitted.