A path tracking Algorithm using the IEEE 802.11 Infrastructure

Yacine Mezali
INRIA
Rocquencourt, France
philippe.jacquet@inria.fr

Philippe Jacquet
INRIA
Rocquencourt, France
yacine.mazali@inria.fr

Georgios Rodolakis
Macquarie University
Sydney Australia
georgios@ics.mq.edu.au

***Abstract*—We propose a new indoor tracking method which can be used on mobile nodes. Our method uses only the received signal strengths as input information and does not require a cinematic motion model to track the mobile node. We discuss in detail the features of our approach and its resulting algorithm. We evaluate the performances of our algorithm using a real signal map.**

I. Introduction

Over the last decade, there has been a tremendous growth in the development of wireless local area networks (WLANs). This proliferation has contributed to an increasing interest in developing location aware technologies, *i.e.* devices capable of locating themselves, and, in turn, providing location aware services. A basic location aware service is the ability to answer the obvious question "where am I?" but also more complicated questions such as "Where is the nearest emergency exit?" This problem has several names: location sensing [1], location estimation, location identification, location determination, geo-location, and is named radio-location when it uses wireless technologies [**?**]. We are interested in indoor self-positioning systems based on the IEEE802.11 wireless technology. Our approach makes use of probabilistic techniques and our method works under the assumption that the received signals are considered as a Gaussian vector whose parameters (mean vector, covariance matrix) are known functions of the plane that have been pre-determined by measurement and cartography.

Our paper is organized as follows: after stating our problem and models, we describe our algorithm, show some of its properties, and finally we present some simulation results.

II. Related Work

Indoor positioning systems based on 802.11 technology. In radio networks, many techniques are used for localization, including: Angle-Based techniques (Angle of Arrival), Lateration-Based techniques (Time of Arrival, Time Difference of Arrival), (RSS) Received Signal Strength based techniques. Angle and Lateration based techniques require capabilities (Hardware, software) which can not be found in a typical 802.11 network node; for example (Angle Based techniques) require directional antennas and a specific signal processing device; whereas 802.11 nodes are generally equipped with omnidirectional integrated antennas. In addition; the multipath propagation phenomenon, which is due to the indoor environment; disturbs the measurements of the directional antennas: and the case may arise when the received signal is not the direct line-of-sight signal, but a reflected one [2]. Moreover in 802.11 systems the time reference which is inherent to Lateration based techniques is of the order of 100 ns; which is insufficient to provide an accurate location [2]. Unlike the two previous approaches (Angle based and Lateration Based approaches), Received Signal Strength based techniques are more appropriate for indoor positioning systems, in fact the easy availability of the signal strength information in 802.11 terminals encourages the use of these techniques. RSS methods can be classified according to two criteria [2]: the first concerns the computational nature of the underlying localization algorithm, whether it is deterministic or probabilistic. The second criterion concerns where the signal strength measurements are collected; thus we may talk about a client-based system if the signal measurements are collected by the mobile node and we use the term infrastructure-based systems if the signal measurements are collected by the infrastructure. Some deterministic methods rely on propagation models to estimate the range to the transmitter. Other deterministic approaches use a radio signal map of the region of interest , which is a kind of radio profile of the region; which results in the region being divided into subregions (cells) and for each cell a signal strength vector is assigned. The estimation of the location is performed by minimizing the distance between The RSS vector (measured or transmitted)by the mobile node and all the RSSs vectors of the signals map. One of the best known deterministic methods in the literature is RADAR [3]. Examples of systems using deterministic client-based methods can be found in:[4],[3], and we can also find systems using deterministic infrastructure based methods in: [5]. Probabilistic methods consider the received signals as random processes, they also divide the area of interest into cells but instead of assigning a unique vector to each cell; each cell is assigned a probability density of the received signals, the received signals and the prior probability densities of the cells are used to compute the maximum likelihood or the maximum a posteriori of the position.

III. Problem Statement

We assume k fixed nodes in the plane which are called anchored nodes. These nodes are similar to the *Access Points* of the infrastructure mode used in wifi networks [6]. We

978-1-4577-0899-2/11/$26.00

consider a mobile node A whose position on the map at time t is the vector $z(t) = (x(t), y(t))$.

We denote by $v(t)$ the signal level sampling vector of mobile A at the time t: $v(t) = (v_1(t), v_2(t), v_3(t), .., v_k(t))$ where each component $v_i(t)$ represents the signal strength level in (dB) received by the mobile node A from the anchored node AP_i at time t.

The basic *localization problem* consists in giving an evaluation of vector $z(t)$ via a vector $\tilde{z}(t) = (\tilde{x}(t), \tilde{y}(t))$ at time t, *given* the past history of signal level vector $v(t)$. To do this we assume that the signal level vector is periodically sampled every δ_T time, and therefore we have the signal level vectors $v(k\delta_T)$ for every integer k such that $k\delta_T \leq t$. Given this information we aim to get the estimate $\tilde{z}(t)$ when t is an integer multiple of δ_T.

A. The Model

We will assume that at each point $z \in P$ the signal level vector is a random Gaussian vector v distributed as $\mathfrak{N}(m(z), \mathbf{Q}^{-1}(z))$ where: $m(z) = \mathbf{E}v$ is the expectation vector and $\mathbf{Q}^{-1}(z) = \mathbf{E}(v - m(z))(v - m(z))^T$ is the covariance matrix, therefore $Q(z)$ corresponds to the inverse covariance matrix; the Gaussian vector v is distributed according to the density function:

$$p(u|z) = \sqrt{\frac{\det(\mathbf{Q}(z))}{\pi^k}} \exp\Big\{-\frac{1}{2}\langle(u-m(z)), \mathbf{Q}(z)(u-m(z))\rangle\Big\}$$

where $\langle a, b\rangle$ denotes the dot product of two vectors a and b. Furthermore we suppose that given the path $(z(0), z(\delta), \ldots, z(k\delta), \ldots)$ the signal level vectors are independent in distribution. This is a strong assumption but it is reasonable if we consider that δ_T is large compared to the Dopler period of the radio system.

B. Localization Algorithm parameters

Our objective is to give an estimate of a node's path $(\tilde{z}(0), \tilde{z}(\delta_T), \ldots, \tilde{z}(k\delta_T))$. The localization algorithm is based on an *a priori* knowledge of functions $\mathbf{Q}(z)$ and $m(z)$. For reasons of clarity we will sometime drop the variable z when no ambiguity is possible and refer to $\mathbf{Q}$ and m.

We define the matrix $\mathbf{R}(z)$ as function of position z in the following way:

$$\mathbf{R}(z) = -\frac{1}{2}\mathrm{tr}\left((\nabla_z\mathbf{Q})\mathbf{Q}^{-1}(\nabla_z\mathbf{Q})\mathbf{Q}^{-1}\right) + \mathrm{tr}\left(\mathbf{Q}(\nabla_z m)\otimes(\nabla_z m)\right) \ . \quad (1)$$

where ∇_z denotes the differential operator with respect to vector z. Notice that the matrix $\mathbf{R}$ issued from tensor products is a symmetric 2×2 matrix with:

$$\mathrm{tr}\left((\nabla_z\mathbf{Q})\mathbf{Q}^{-1}(\nabla_z\mathbf{Q})\mathbf{Q}^{-1}\right) = \left[\begin{array}{cc} \mathrm{tr}\left((\frac{\partial}{\partial x}\mathbf{Q})\mathbf{Q}^{-1}(\frac{\partial}{\partial x}\mathbf{Q})\mathbf{Q}^{-1}\right) & \mathrm{tr}\left((\frac{\partial}{\partial x}\mathbf{Q})\mathbf{Q}^{-1}(\frac{\partial}{\partial y}\mathbf{Q})\mathbf{Q}^{-1}\right) \\ \mathrm{tr}\left((\frac{\partial}{\partial y}\mathbf{Q})\mathbf{Q}^{-1}(\frac{\partial}{\partial x}\mathbf{Q})\mathbf{Q}^{-1}\right) & \mathrm{tr}\left((\frac{\partial}{\partial y}\mathbf{Q})\mathbf{Q}^{-1}(\frac{\partial}{\partial y}\mathbf{Q})\mathbf{Q}^{-1}\right) \end{array}\right]$$

and

$$\mathrm{tr}\left(\mathbf{Q}(\nabla_z m)\otimes(\nabla_z m)\right) = \left[\begin{array}{cc} \mathrm{tr}\left(\mathbf{Q}(\frac{\partial}{\partial x}m)\otimes(\frac{\partial}{\partial x}m)\right) & \mathrm{tr}\left(\mathbf{Q}(\frac{\partial}{\partial x}m)\otimes(\frac{\partial}{\partial y}m)\right) \\ \mathrm{tr}\left(\mathbf{Q}(\frac{\partial}{\partial y}m)\otimes(\frac{\partial}{\partial x}m)\right) & \mathrm{tr}\left(\mathbf{Q}(\frac{\partial}{\partial y}m)\otimes(\frac{\partial}{\partial y}m)\right) \end{array}\right] .$$

Let v be an arbitrary vector of the signal. We define $L(z, v) = -\log(p(v|z)$ as a quadratic function of v defined in III-A, and therefore we assume that $\nabla_z L(z, v)$ is a computable quadratic vector of v with two components: $\frac{\partial}{\partial x}L(z, v)$ and $\frac{\partial}{\partial y}L(z, v)$.

IV. The path tracking algorithm and its properties

A. The path tracking algorithm

The proposed path tracking algorithm is the following

Proposition 1 *Let $\tilde{z}(t) = z$ when t is an integer multiple of δ_T we define:*

$$\tilde{z}(t + \delta_T) = \tilde{z} + \mathbf{R}^{-1}(z)\nabla_z L(z, v(t + \delta_T)) \ .$$

Lemma 1 *We have the expression*

$$\nabla_z L(z, v) = \mathrm{tr}(\mathbf{Q}(v - m(z)) \otimes \nabla_z m(z)) - \frac{1}{2}\mathrm{tr}((v - m(z)) \otimes (v - m(z))\nabla_z\mathbf{Q}) + \frac{1}{2}\mathrm{tr}(\mathbf{Q}^{-1}\nabla_z\mathbf{Q}) \quad (2)$$

Proof: Noticing that in (III-A) $\langle(u - m(z)), \mathbf{Q}(z)(u - m(z))\rangle = \mathrm{tr}(\mathbf{Q}(u - m(z)) \otimes (u - m(z)))$ and also that $\log(\det(\mathbf{Q})) = \mathrm{tr}(\log(\mathbf{Q}))$ and $\nabla_z\mathrm{tr}(\log(\mathbf{Q})) = \mathrm{tr}(\mathbf{Q}^{-1}\nabla_z\mathbf{Q})$ we obtain the claimed result. ■

B. Properties of the algorithm

In this section we assume that the trajectory of the mobile node is determinist. Since only the signals are random, the estimated path $\tilde{z}(t)$ is a random variable. In the following we will assume that $\tilde{z}(t)$ is given at a given time, and we analyze the distribution of $\tilde{z}(t+\delta_T)$. We call $\tilde{z}(t+\delta_T)-\tilde{z}(t)$ the drift of the algorithm at time t

Theorem 1 (zero average drift property) *Let $z(t) = z(t + \delta_T) = z$, given that $\tilde{z}(t) = z$ we have*

$$\boldsymbol{E}(\tilde{z}(t + \delta_T)) = z \ .$$

Theorem 2 (Small average drift property) *Let $z(t) = z$ and $z(t + \delta_T) = z + \delta_Z$, given that $\tilde{z}(t) = z$ we have when $\delta_z \to 0$*

$$\boldsymbol{E}(\tilde{z}(t + \delta_T)) = z(t + \delta_T) + O(\|\delta_Z\|^2) \ .$$

Theorem 3 (Error estimate) *Let $z(t) = z(t+\delta_T) = z$, given that $\tilde{z}(t) = z$ we have the covariance matrix of $\tilde{z}(t+\delta_T)$ equal*

$$\mathbf{Cov}(\tilde{z}(t + \delta_T)) = R^{-1}F(\Theta)(R^{-1})^T$$

Where $F(\Theta)$ corresponds to the Fisher information matrix of the multivariate normal distribution which is followed by the vector of the received signals and θ is the parameter vector $\Theta = [\theta_1 = x, \theta_2 = y]$ which means: $\Theta = z$.

$$\mathbf{F(\Theta)_{n,m}} = \frac{\partial m(z)^T}{\partial\theta_n}\mathbf{Q}(z)\frac{\partial m(z)}{\partial\theta_m} + \frac{1}{2}\mathrm{tr}(\mathbf{Q}(z)\frac{\partial Q(z)^{-1}}{\partial\theta_n}\mathbf{Q}(z)\frac{\partial Q(z)^{-1}}{\partial\theta_m})$$

V. Experimentations

A. Simulations based on pre-determined differentiable functions

To simplify, we assume that the access point signals are independent distributions, *i.e.* matrix $\mathbf{Q}(z)$ are always diagonal. The diagonal coefficients of $\mathbf{Q}(z)$ are the sequence $(\frac{1}{\mathbf{var}_1(z)}, \dots, \frac{1}{\mathbf{var}_k(z)})$.

We set $m(z) = (m_1(z), \dots, m_k(z))$. We assume that the statistic parameters: $m_i(z), \mathbf{var}_i(z)$ are the pre-determined analytical functions of the plane;

$$m_i(z) = \frac{-50}{(x - x_i)^2 + (y - y_i)^2 + 1} \mathbf{var}_i(z) = \frac{1}{(x - x_i)^2 + (y - y_i)^2 + 1} \quad (3)$$

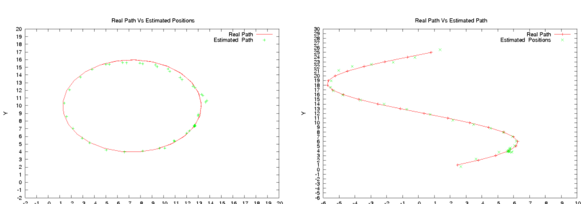
where the (x_i, y_i), $i \in [1, 6]$ represents the coordinates of the access point number i in the plan. The mean model given above is realistic since the received signal strength decreases when the distance from the AP increases.

We start by simulating a uniform circular motion (figure **??**(a)), whose center is located at $(8, 4)$ and whose angular velocity $\omega = \angle 14.32°/second$. Then, we check whether our algorithm can track and predict a more complicated non-uniform motion such as a sinusoidal motion . The parametric equations of these two motions are given below: with $a = \frac{3}{2}, b = 4, c = 1$:

$$\begin{cases} x(t) = a\cos(t) + b \\ y(t) = ct \end{cases} \quad (4)$$

We tracked the cirucular and the sinusoidal motions assuming that each signal is distributed according to an independent normal distribution $\mathcal{N}(m_i(z), \sqrt{\mathbf{var}_i(z)})$, (figure **??**(b)). Then we tracked the same motions considering that each signal received in the case of the circular motion is an independent gamma distribution $\Gamma(k, mean_i(z))$ with $k = 7$.(figure 2(a)) and each signal received in the case of the sinusoidal motion is distributed exponentially with a parametric mean equals to $m_i(z)$ (figure 2(b)). we can notice that our algorithm is still faithful even for a non Gaussian distributed signal strengths.

Fig. 1. Results using a predetermined analytic functions for $m(z), \mathbf{Q}(z)$:

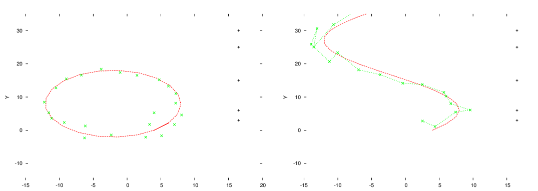


(a) Real Path Vs predicted Path, uniform circular motion (b) Real Path Vs predicted Path, sinusoidal motion

B. Experiments based on signal maps via a discrete model

Since we are trying to locate the mobile node on the plane, it is impossible to measure and store all the values of parameters $m(z)$ and $\mathbf{Q}(z)$ at every continuous position z; and we can only afford to measure the parameters $m(z)$, $\mathbf{Q}(z)$ for a finite set of positions S following a grid pattern. Given this information, we want to interpolate the values of $m(z), \mathbf{Q}(z)$ and $\nabla_z m(z), \nabla_z \mathbf{Q}(z))$ at any position on the plan. This is done via a triangulations spanning technique using barycentric coordinates in each triangle, such as is often used in finite element methods. Triangulation is obtained with the Voronoi method.

Fig. 2. Results assuming non Gaussian distribution of the received signals



(a) Real Path Vs predicted Path, uniform circular motion (b) Real Path Vs predicted Path, sinusoidal motion (RSS are exponentially distributed)

1) Description of the Environment: we carried out our experiments inside a hall building (figure:3(a)). hall building which contains more open space, but contains hosts many metallic grids which affect the signal propagation. This indoor environment is very heterogeneous for the experimentats.

2) The signal maps: The sensing process used to establish the signal map corresponds to capturing WIFI signals. To create the signal map, the signals are filtered via a Mmapsonte Carlo method to get its multivariate density. The mean vector and covariance matrix are calculated for each position in S.

We created a signal map for the hall building, this latter is a lattice of 96 points organized in twelve lines of eight rows (figure3(b)). As shown in (figure:3(b)), our test areas are relatively small compared to the test area of[7].

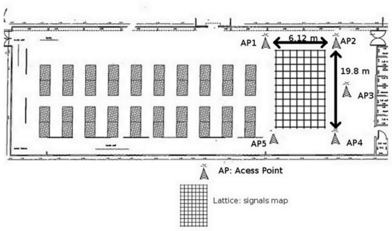
3) Describing the experiments: To simulate the measured signals on the mobile node, the vector of the received signals strength is created by simulating its multivariate density that has been obtained during the signal map creation process. But the signal map used by the tracking algorithm uses only the mean vector and covariance matrix obtained for each position. The mobile node's position is generated using parametric equations.

We describe our results using the following parameters: number of iterations, the number of received signals vectors and the position error. The number of iterations corresponds to the number of time steps taken by the path of our mobile node. An iteration corresponds to the prediction made by the algorithm for one time step. In our experiments one time step δ_T equals one second. To analyze the results, we compute the average position error, the 50, the 70 and the 90 percentiles errors in meters or expressed as the ratio with the signal map discretization unit length. We define the displacement vector as the vector between the two positions of the mobile node before and after one time step. We also compare the magnitude of the displacement vector to the average position error.

4) Results from the signal map of the hall building (Table: I,II): We tracked three different motions (figure 4, **??**): a uniform linear motion whose initial position is $(0.765, 0.765)$, a uniform circular motion whose center is $(3.39, 3.24)$, and finally a sinusoidal motion whose initial position is $(2.40, 0.45)$. and which evolves according to the equation ((4)) with

Fig. 3. Buildings

(a) Building 21



(b) signal map of building 21

$a = 0.3$, =2.40, =0.7 (figure:(d)). The magnitudes of the displacement vecto rs corresponding to each these motions are respectively: $0.75(m)$, $0.54(m)$, $0.58(m)$.

TABLE I
RESULTS FROM THE SIGNAL MAP OF HALL BUILDING

	iteration	Average position error
Uniform linear motion	29	0.7427
Uniform circular motion	26	0.6688
Sinusoidal motion	24	0.5067

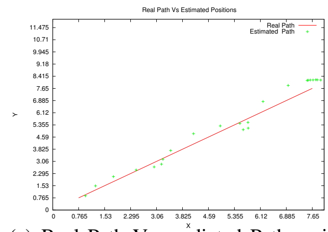
TABLE II
RESULTS FROM SIGNAL MAP OF HALL BUILDING

Percentiles of position error (m)			
	50%	70%	90%
Uniform linear motion	0.6696	0.9173	1.3660
Uniform circular motion	0.6770	0.8470	1.1118
Sinusoidal motion	0.5048	0.5952	0.8183

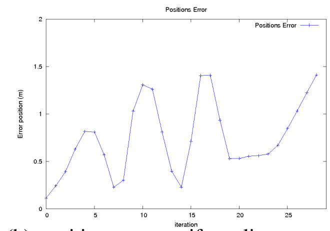
	displacement(m)	h
Uniform linear motion	0.7573	0.9708
Uniform circular motion	0, 5409	0.8742
Sinusoidal Motion	0, 58260	0.6598

We also compare the magnitude of the displacement vector of these motions to the average position error (table: V-B4). The discretization length of the signal map of the hall building is 0.765 m, which is almost three times smaller than the discretization length used in [7] (2 m), this should make our signal map more sensitive to human randomness. In spite of this sensitivity, it is worth noticing that in the three tracked motions, the average of the error position is always smaller than the magnitude of the displacement vectors (table: V-B4) . In addition, the ratio between the average position error and signal map discretization length is always smaller than one , whereas this ratio exceeds one in [7] (when using one test observation). We can notice that, even though the sinusoidal motion in (figure **??**) has no constant velocity vector, our algorithm is still able to track it.

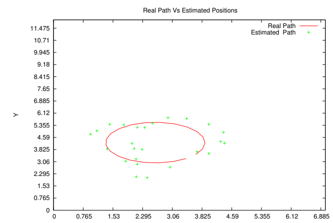
Fig. 4. Results using the signal map of the hall building:



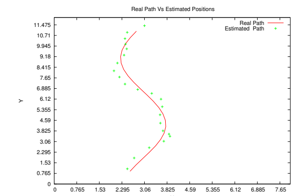
(a) Real Path Vs predicted Path, uniform linear motion



(b) position error, uniform linear motion



(c) Real Path Vs predicted Path, uniform circular motion



(d) Real Path Vs predicted Path, sinusoidal motion

VI. CONCLUSION

We have designed and developed a positioning algorithm which does not need any information about the motion model, but is based on a Gaussian model of the signal distribution in each position. We have shown analytically that the algorithm is faithfull on average for one time step, *i.e.* the average drift is zero when the mobile node stays motionless, and equal to the displacement vector when the mobile moves, as long as the displacement vector is small. We have an analytical evaluation for the position error by providing the covariance matrix of the drift. Our analysis is confirmed by simulations.

Future work can be the extention of our analytical results. We need to analyse the convergence of the algorithm when the initial position is arbitrary. The simulations suggest that the convergence is fast. We also need to prove that the error estimate does not amplify and diverge when the tracking duration increases. Simulations suggest that the error remains stable.

REFERENCES

[1] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE*, pp. 1067–1080, 2007.

[2] G. Mao and B. Fidan, *Introduction to Wireless Sensor Network Localization*. InformatIon scIence reference, 2009.

[3] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," *IEEE*, 2000.

[4] K. P. . C. P. Prasithsangaree, P., "On indoor position location with wireless lans.," *IEEE IEEE International Symposium on Personal Indoorand Mobile Radio Communications*, pp. 720–724, 2002.

[5] P. Krishnan, A. S. Krishnakumar, and . M. C. . G. S. . Ju, W., "A system for lease: Location estimation assisted by stationary emitters for indoor rf wireless networks.," *In proceedings of the IEEE Internationa lInternational Conference on Computer Communications*, pp. 1001–1011, 2004.

[6] W. A. A. N. S. Y. J. Wan, "Your 802.11 wireless network has no clothes," tech. rep., Department of Computer Science University of Maryland, 2001.

[7] T. Roos, P. Myllymaki, H. Tirri, P. Misikangas, and J. Sievanen, "A probabilistic approach to wlan user location estimation," *International Journal of wireless Networks*, vol. 9, pp. 155–162, 2002.