INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Localization and Path Tracking Using the IEEE 802.11 Infrastructure

Philippe Jacquet — Tonia Papakonstantinou — Georgios Rodolakis

N° 6499

Avril 2008

Thème COM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6499--FR+ENG

Localization and Path Tracking Using the IEEE 802.11 Infrastructure

Philippe Jacquet, Tonia Papakonstantinou , Georgios Rodolakis

Thème COM — Systèmes communicants
Projets Hipercom

Rapport de recherche n° 6499 — Avril 2008 — 13 pages

Abstract: We propose a new localization method in order to estimate the position of wireless devices in real time, using already existent infrastructure (such as IEEE 802.11 access points). We present a filtering algorithm that computes the most likely path taken by a mobile node, based on signal strength measurements and a known signal strength map. In our approach we do not assume that the node mobility model is known. The algorithm calculates analytically in real time the optimal correction in the mobile node's estimated position, using current signal measurements. We discuss how this algorithm can be used in a practical implementation and we evaluate the performance of our method using numerical simulations.

Key-words: localization, wireless, IEEE 802.11, filter

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Géolocalisation utilisant l'infrastructure IEEE 802.11

Résumé : Nous proposons une nouvelle méthode de géolocalisation pour estimer la position d'équipements mobiles en temps réel, en utilisant l'infrastrucure sans-fil existante. Nous présentons un algorithme de filtrage qui calcule le chemin le plus probable d'un nœud mobile, basé sur des mesures de puissance de signal et une cartographie du signal radio. Dans notre approche, nous ne faisons aucune hypothèse sur la mobilité des nœuds. L'algorithme déduit la correction optimale de la position d'un nœud mobile en temps réel, en se basant sur les mesures de signal. Nous discutons comment cet algorithme peut être utilisé dans une implémentation pratique et nous évaluons les performances avec des simulations numériques.

Mots-clés : géolocalisation, sans-fil, IEEE 802.11, filtre

1 Introduction

Mobility creates a need for location aware applications, which aim to determine the physical position of a mobile device. With the recent thrive in ubiquitous computing, dozens of applications demand such location information; for example, asset tracking in warehouses and hospitals, locating children in public areas, emergency management, vehicles with navigation tools and location-based services, guiding visitors in museums, and virtual reality applications. Therefore, the accurate localization of objects and people in indoor environments is considered an important building block for ubiquitous computing applications.

GPS, which uses the location of satellites orbiting earth to triangulate longitude and latitude, has been used in many commercial applications. However, this approach is limited to environments in which a clear line of sight to the satellites is available. Inside buildings and even in some outdoor environments, structure and foliage can affect the ability to communicate with GPS satellites. Moreover, while many outdoors applications can work successfully with an accuracy of hundreds of meters, indoor applications usually require a granularity of a few meters. Hence, most research on indoor localization systems has been based on the use of short-range signals, such as Wi-Fi [12, 5, 2], Bluetooth [3], ultrasound [11, 9], infrared [13], or RFID [7, 10].

Early localization systems were based exclusively on signal strength measurements and simple triangulation methods. However, such systems are limited, due to interferences and the transient characteristics of radio propagation. The dynamic characteristics of the environment impose important challenges, for the design of a scalable, easily deployed, and computationally inexpensive localization system.

Furthermore, due to cost and maintenance constraints, the deployment of a specialized infrastructure for a localization system is not always feasible. Our general aim is to design a localization system, that does not depend on specialized hardware or on extensive training. Given the wide deployment of the IEEE 802.11 communication infrastructure, its use for both communication and positioning becomes an attractive choice.

Our main contribution is the development of a new localization algorithm which is well adapted to this particular context, and can be used as an alternative to systems using Kalman filters [8] or particle filters [14]. The proposed method has the advantage of being based on analytical calculations, which give closed formulas for the node estimated position, using signal strength measurements. Therefore, the implementation of the algorithm is very simple and computationally inexpensive. Moreover, it does not require keeping a history of previous node positions and the localization can be performed in real time. In fact, our approach consists in an adaptation of the Kalman Filter method, in the general framework described in [6], without any constraint or assumption about the node mobility. The proposed localization algorithm is based on an analytical drift evaluation. At each step, the algorithm deduces the optimal correction in the mobile node's position based on the current estimated location, the signal measurements and the known signal distribution database. The corrections performed result in a random walk, which on average follows the real path taken by the mobile node; in fact the expectation of the error in the estimated position is 0.

The rest of the paper is organized as follows. In Section 2 we present the methodology used. In Section 3 we give some analytical examples to illustrate our approach. We evaluate the performance of the algorithm using numerical simulations in Section 4 We conclude and we discuss some directions for further research in Section 5.

2 Methodology

2.1 Model and problem statement

We assume k fixed wireless nodes called anchored points (AP). We consider a mobile node A that moves in a plane. Our aim is to track this node's movement. We denote $z(t) = (x(t), y(t))$ the position of node A at time t. We denote $\mathbf{v}(t)$ the signal level sampling vector at time t: $\mathbf{v}(t) = (u_1(t), u_2(t), \ldots, u_k(t))$ where $u_i(t)$ is the signal level (in dB) that node A receives from AP_i at time t. We assume that there is a database established prior to the measurement that provides the probability distribution of the signal level vector at any position $z = (x, y)$. For example and in order to simplify computations, we will assume that, at point z, the signal level vector is distributed according to a normal distribution with mean vector $m(z)$ and inverse covariance matrix $\mathbf{Q}_z$. More precisely, the probability density of the distribution of $\mathbf{v}$ on vector $\mathbf{u}$ is $p(z, \mathbf{u}) = \sqrt{\frac{det(\mathbf{Q}_z)}{\pi^k}} \exp(-\frac{1}{2}\langle(\mathbf{u} - m(z))\mathbf{Q}_z(\mathbf{u} - m(z))\rangle)$.

The main objective is to find a path $(z(0), z(1), \ldots, z(T))$ such that the product $\prod_{i=0}^{i=T} p(z(i), \mathbf{v}(i))$ is maximal, assuming sampling at integer multiples of a time unit. If we denote $\ell(z, \mathbf{u}) = -\log p(z, \mathbf{u})$, this is equivalent to minimizing the sum $\sum_{i=0}^{i=T} \ell(z(i), \mathbf{v}(i))$, or the integral $\int_0^T \ell(z(t), \mathbf{v}(t))dt$, assuming a continuous approximation of sampling times.

2.2 Path optimization

In this section, we will describe the methodology we use in order to derive a path tracking algorithm which achieves the previously defined objective. In our approach, we assume a perturbation of the the mobile node's position with respect to the most likely path. In order to obtain a localization algorithm, we assume that the node's speed is bounded and we calculate analytically the signal strength perturbation. We will describe how to derive an algorithm which, at each step, aims to correct the discrepancy between the real and estimated positions. As we will see, the corrections performed result in a random walk, which we tune so that it follows on average the real path taken by the mobile node.

In order to minimize the integral $\int_0^T \ell(z(t), \mathbf{v}(t))dt$, we assume a small perturbation $z(t) + \delta(t)$ in the mobile node's position at time t. The quantity we wish to optimize will become:

$$\int_0^T \ell(z(t) + \delta(t), \mathbf{v}(t))dt = \int_0^T \ell(z(t), \mathbf{v}(t))dt + \int_0^T \langle \delta(t).\nabla_z \ell(z(t), \mathbf{v}(t)) \rangle \, dt. \quad (1)$$

Therefore, if we wish to optimize the integral without any constraint, at minimum we should have $\langle \delta(t).\nabla_z \ell(z(t), \mathbf{v}(t))\rangle = 0$ point-wise for all possible $\delta(t)$. This means that $\nabla_z \ell(z(t), \mathbf{v}(t)) = 0$ or $m(z(t)) = \mathbf{v}(t)$. The last condition would imply that $z(t)$ will follow locations where $m(z(t)) = \mathbf{v}(t)$, which may imply unbounded speed ż(t) or unrealistic teleportation in the estimated positions (when $\mathbf{v}(t)$ is discontinuous).

Path optimization with bounded speed

For this reason, we should introduce a constraint based on a bound for the speed. For example, we can assume that the speed at any given point x must be constant with only direction as a tunable parameter. In this case, we have the additional condition that $\delta(t)$ must be orthogonal to the speed ż(t). In combination with condition $\langle \delta(t).\nabla_z \ell(z(t), \mathbf{v}(t))\rangle = 0$, we should have ż(t) proportional to $\nabla_z \ell(z(t), \mathbf{v}(t))$.

Therefore a path such that ż(t) is always proportional to $\nabla_z \ell(z(t), \mathbf{v}(t))$ may be a good candidate path. For example one can assume that $z(i+1) = z(i) + C\nabla_z \ell(z(i), \mathbf{v}(i))$. If $\mathbf{v}(t)$ is random, then $z(i)$ will evolve like a random walk.

We will compute the gradient $\nabla_z \ell(z(t), \mathbf{v}(t))$ to investigate the evolution of this random walk. The expression of the gradient is the following:

$$\begin{aligned}\nabla_z \ell(z, \mathbf{v}) = \frac{1}{2}(&\langle(\mathbf{v} - m(z))\nabla_z \mathbf{Q}_z(\mathbf{v} - m(z))\rangle \\ &- 2\,\langle \nabla_z m(z)\mathbf{Q}_z(\mathbf{v} - m(z))\rangle \\ &- \nabla_z \log\det(\mathbf{Q}_z). \end{aligned} \tag{2}$$

Notice that $\log\det(\mathbf{Q}_z) = \mathrm{tr}(\log \mathbf{Q}_z)$ and therefore $\nabla_z \log\det(\mathbf{Q}_z) = \mathrm{tr}(\nabla_z \mathbf{Q}_z.\mathbf{Q}_z^{-1})$. We also have $\langle u\mathbf{Q}_z u\rangle = \mathrm{tr}(\mathbf{Q}_z u \otimes u)$ by simple equivalence of lecture.

Random walk optimization

In this section, we suppose that the algorithm for tracking the mobile position is the following:

$$z(i+1) = z(i) + C_z \nabla_z \ell(z(i), \mathbf{v}(i)) \tag{3}$$

or, with an arbitrary time step δt:

$$z(t+\delta t) = z(t) + \delta t C_z \nabla_z \ell(z(t), \mathbf{v}(t)), \tag{4}$$

where C_z is a function of z which we will compute. In fact, depending on C_z, many paths will be candidates, however the one that minimizes $\int_0^T \ell(z, \mathbf{v})dt$ will be the best approximation.

Since the evolution of a candidate path will look like a random walk, it would be interesting to know the average drift of this random walk, in order to compute the constant C_z. Assume that $\mathbf{v}$ is distributed according a normal distribution of inverse covariance matrix $\mathbf{Q}$ and mean m, i.e. $E(\mathbf{v}) = m$ and

$E((\mathbf{v}-m)\otimes(\mathbf{v}-m)) = \mathbf{Q}^{-1}$. Then, we have:

$$\begin{aligned} E(\nabla_z \ell(z,\mathbf{v})) = \frac{1}{2}(&\mathrm{tr}(\nabla_z \mathbf{Q}_z \mathbf{Q}^{-1}) \\ &+ \langle (m-m(z))\nabla_z \mathbf{Q}_z (m-m(z))\rangle \\ &- 2\,\langle \nabla_z m(z)\mathbf{Q}_z (m-m(z))\rangle \\ &- \mathrm{tr}(\nabla_z \mathbf{Q}_z.\mathbf{Q}_z^{-1})). \end{aligned} \tag{5}$$

First of all we notice that if $\mathbf{Q}=\mathbf{Q}_z$ and $m=m(z)$, that is if $\mathbf{v}$ is distributed exactly like the sampled signal at position z, then we have $E(\nabla_z \ell(z,\mathbf{v}))=0$ (thanks to $\mathrm{tr}(\nabla_z \mathbf{Q}_z.\mathbf{Q}_z^{-1})$ which is canceled by $\nabla_z \log\det(\mathbf{Q}_z)$). Now, assuming again that there is a small perturbation in the node's position, we have

$$m = m(z+\delta t\dot{z}) = m(z) + \langle \dot{z}.\nabla_z m(z)\rangle\,\delta t \tag{6}$$

and

$$\mathbf{Q} = \mathbf{Q}_{z+\dot{z}\delta t} = \mathbf{Q}_z + \langle \dot{z}.\nabla_z \mathbf{Q}_z\rangle\,\delta t. \tag{7}$$

Then, we can derive the average drift at first order:

$$\begin{aligned} E(\nabla_z \ell(z,\mathbf{v})) &= \frac{\delta t}{2}(\mathrm{tr}(\nabla_z \mathbf{Q}_z \mathbf{Q}_z^{-1}\,\langle \dot{z}.\nabla_z \mathbf{Q}_z\rangle\,\mathbf{Q}_z^{-1} \\ &\quad - 2(\nabla_z m(z)\mathbf{Q}_z\,\langle \dot{z}.\nabla_z m(z)\rangle) \\ &= \delta t \mathbf{R}_z \dot{z} + O(\delta t^2), \end{aligned} \tag{8}$$

where $\mathbf{R}_z$ is a 2x2 matrix with expression:

$$\begin{aligned} \mathbf{R}_z = \frac{1}{2}(&\mathrm{tr}(\nabla_z \mathbf{Q}_z.\mathbf{Q}_z^{-1}\nabla_z \mathbf{Q}_z.\mathbf{Q}_z^{-1}) \\ &- 2\mathrm{tr}(\mathbf{Q}_z \nabla_z m(z)\otimes\nabla_z m(z))). \end{aligned} \tag{9}$$

Localization algorithm

In order to achieve the correct drift, a good algorithm consists in taking at each step C_z equal to the inverse of the 2x2 matrix R_z, since $E(\mathbf{R}_z^{-1}\nabla_z \ell(z(t),\mathbf{v}(t))) = \dot{z}\delta t$. This implies that the expectation of the difference between the estimated and the real path is zero. Hence, the localization algorithm we propose is the following:

$$z(t+\delta t) = z(t) + \mathbf{R}_z^{-1}\nabla_z \ell(z(t),\mathbf{v}(t)), \tag{10}$$

which computes the new node's position $z(t+\delta t)$, based on the previously estimated position $z(t)$. Consequently, in order to perform localization of the mobile node, we only need to calculate at each step $\mathbf{R}_z$ and $\nabla_z \ell(z(t),\mathbf{v}(t))$, according to (2) and (9).

3 Analytical Examples

In this section, we provide some analytical examples, in order to better illustrate how the previously derived formulas can be used in practice. More precisely, we develop simpler formulas for the quantities R_z and $\nabla_z \ell(z(t),\mathbf{v}(t))$, which are used in the localization algorithm according to (10). In all the following examples, we assume that there is no co-variance among the signals received from the different APs.

3.1 Constant variance

We consider the simple case where the covariance matrix is of the form $\mathbf{Q}_z = \lambda \mathbf{I}$, which means that the variance is constant and the same for every AP. In this case, we have:

$$\begin{aligned}\nabla_z \ell(z, \mathbf{v}) &= -\lambda \sum_{i=1}^{i=k} \nabla_z m_i(z).(\mathbf{v}_i - m_i(z)) \\ &= -\lambda \sum_{i=1}^{i=k} \begin{bmatrix} \frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial x} & \frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial y} \\ \frac{\partial m_i}{\partial x}\frac{\partial m_i}{\partial y} & \frac{\partial m_i}{\partial y}\frac{\partial m_i}{\partial y} \end{bmatrix},\end{aligned} \tag{11}$$

and

$$\begin{aligned}\mathbf{R}_z &= -\lambda \sum_{i=1}^{i=k} \nabla_z m_i(z) \otimes \nabla_z m_i(z) \\ &= -\lambda \sum_{i=1}^{i=k} \begin{bmatrix} (u_i - m_i)\frac{\partial m_i}{\partial x} \\ (u_i - m_i)\frac{\partial m_i}{\partial y} \end{bmatrix}.\end{aligned} \tag{12}$$

3.2 Non-constant variance

We now consider the case where $\mathbf{Q}_z = \lambda(z)\mathbf{I}$, *i.e.*, the variance is the same function of z for every AP. In this case:

$$\begin{aligned}\nabla_z \ell(z, \mathbf{v}) = \frac{1}{2}(\nabla_z \lambda(z) & \sum_{i=1}^{i=k} (\mathbf{v}_i - m_i(z))^2 \\ & - 2\lambda(z) \sum_{i=1}^{i=k} \nabla_z m_i(z).(\mathbf{v}_i - m_i(z)) \\ & - k \frac{\nabla_z \lambda(z)}{\lambda(z)}),\end{aligned} \tag{13}$$

and

$$\begin{aligned}\mathbf{R}_z = \frac{1}{2}(& \frac{\nabla_z \lambda(z)}{\lambda(z)} \otimes \frac{\nabla_z \lambda(z)}{\lambda(z)} \\ & - 2\lambda(z) \sum_{i=1}^{i=k} \nabla_z m_i(z) \otimes \nabla_z m_i(z)).\end{aligned} \tag{14}$$

Finally, we consider that we have different functions of the variance for every AP. This is the most realistic assumption presented in this section, where we have assumed that there is no co-variance among the signals of different APs. When the APs are placed in different locations, the latter can be considered a sufficiently realistic condition, and the formulas can be used for path tracking in real environments. Different variance functions for every AP mean that $\mathbf{Q}_z$

would be a diagonal array of the form:

$$\mathbf{Q}_z = \begin{bmatrix} \lambda_1(z) & 0 & \dots & 0 \\ 0 & \lambda_2(z) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_k(z) \end{bmatrix}.$$

Hence, we have:

$$\begin{aligned} \nabla_z \ell(z, \mathbf{v}) = \frac{1}{2}(&\sum_{i=1}^{i=k} \nabla_z \lambda_i(z)(\mathbf{v}_i - m_i(z))^2 \\ &- 2\sum_{i=1}^{i=k} \lambda_i(z) \nabla_z m_i(z).(\mathbf{v}_i - m_i(z)) \\ &- \sum_{i=1}^{i=k} \frac{\nabla_z \lambda_i(z)}{\lambda_i(z)}), \end{aligned} \tag{15}$$

and

$$\begin{aligned} \mathbf{R}_z = \frac{1}{2}(&\frac{\nabla_z \lambda_i(z)}{\lambda_i(z)} \otimes \frac{\nabla_z \lambda_i(z)}{\lambda_i(z)} \\ &- 2\sum_{i=1}^{i=k} \lambda_i(z) \nabla_z m_i(z) \otimes \nabla_z m_i(z)). \end{aligned} \tag{16}$$

4 Numerical Simulations

In this section, we study the performance of our approach via numerical simulations, in different cases, following the analytical examples presented in the previous section. Again, we assume that there is zero co-variance among the signal strengths of the different APs. We also consider here that the initial node position is known (however, this constraint is not essential, since simulations show that the algorithm converges to the real path after a few steps).

4.1 Constant variance

We first perform simulations assuming that the signal strength variance is constant and equal to λ. We consider that we have three APs, so we have three different functions for the mean of every AP. As an example we take the following linear functions of z for the mean signal strength of the APs: $m_1(x, y) = x + y$, $m_2(x, y) = x + 2y$ and $m_3(x, y) = 2x + 3y$. More particularly, from (12):

$$\mathbf{R}_z = -\lambda \begin{bmatrix} 6 & 9 \\ 9 & 14 \end{bmatrix},$$

and from (11):

$$\nabla_z \ell(z, \mathbf{v}) = -\lambda \begin{bmatrix} (u_1 - m_1) + (u_2 - m_2) + 2(u_3 - m_3) \\ (u_1 - m_1) + 2(u_2 - m_2) + 3(u_3 - m_3) \end{bmatrix}.$$

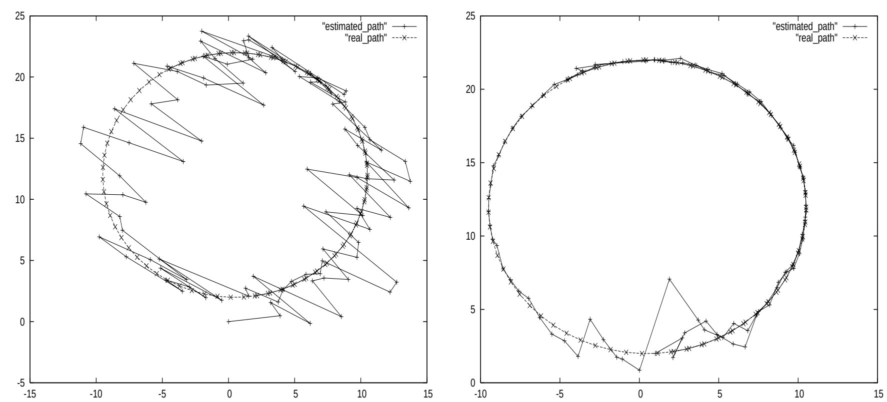
Consequently, at each step of the algorithm we take the following correction in the mobile node's position:

$$\mathbf{R}_z^{-1}\nabla_z \ell(z,\mathbf{v}) = \begin{bmatrix} \frac{5}{3}(u_1 - m_1) - \frac{4}{3}(u_2 - m_2) + \frac{1}{3}(u_3 - m_3) \\ -(u_1 - m_1) + (u_2 - m_2) \end{bmatrix}$$

where m_1, m_2 and m_3 are computed based on the previous estimated position's coordinates, while u_1, u_2 and u_3 are the signal strength measurements. Notice that the value the result does not depend on the value of λ.

As a second example, we take the following non-linear functions for the signal strength mean: $m_1(x,y) = x^3 + y$, $m_2(x,y) = \sin(x) + \cos(y)$ and $m_3(x,y) = x^3 + y^2$. For a circular path, Fig. 1(a) and Fig. 1(b) compare the estimated path with the real path, for the two respective examples.

In order to improve the path estimation, we use an exponential moving average (which can be computed in real time without keeping a history of the path). We plot the path estimate for the first example in Fig. 2, which clearly improves on the estimate of Fig. 1(a).



(a) First example. (b) Second example.

Figure 1: Real path vs. estimated path, constant signal strength variance.

4.2 Non-constant variance

We now assume that the signal variance is the same for all APs, but it varies with space. We take the function $\lambda(x,y) = 1000/(1000+x)$, where the covariance matrix is $\lambda(x,y)\mathbf{I}$. Fig 3 compares the real and estimated paths in a numerical example where we used the non linear functions of the first example of Section 4.1 for the signal mean.

Finally, we consider the case of different signal variance functions for each AP. We take again the same functions for the mean, and for the different variances: $\lambda_1(x,y) = 4000000$, $\lambda_2(x,y) = 1000 + 100y^{0.5}$ and $\lambda_3(x,y) = 1000/(1 + x/100)$. The real and estimated paths are depicted in Fig 4.

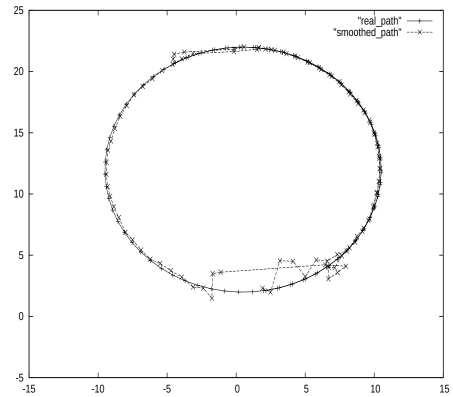


Figure 2: Estimated path using an exponential smoothing average.

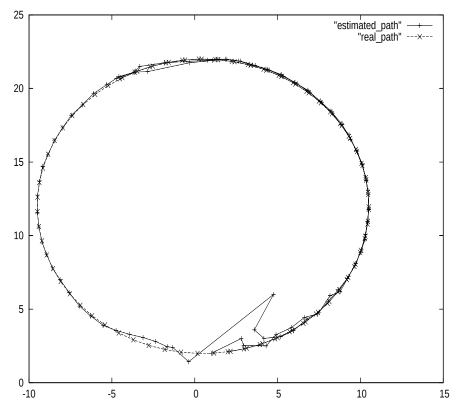


Figure 3: Real path vs. estimated path, same non-constant signal strength variance for every AP.

4.3 Signal characteristics known only at specific locations

In this section, we consider that we have a map of signal strength characteristics in some points only. In other words, we have a database of the signal mean and variance on these selected positions. This situation corresponds to a practical localization system.

Therefore, we use a Delaunay triangulation [4], in order to estimate the mean and variance of every other position, based on linear interpolation and the known values of the signal characteristics on the triangulation points. The Delaunay triangulation gives satisfactory (close to the optimal) interpolating estimations. Since we have the signal strength measurements of the nodes of every triangle, the triangulation is used to interpolate the values of every other point inside the triangle.

For our simulations, we use the CGAL [1] library to compute the triangulation of a given set of points. We use the functions of Section 4.2 for the signal mean and variance, but we assume that the values are known only on the

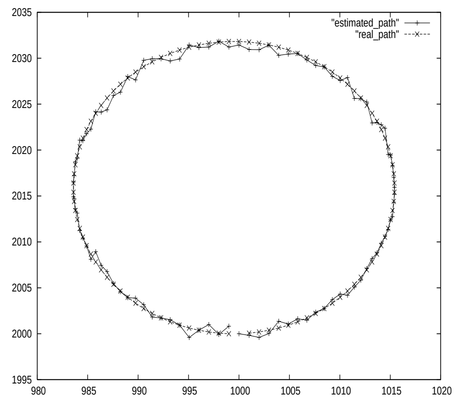
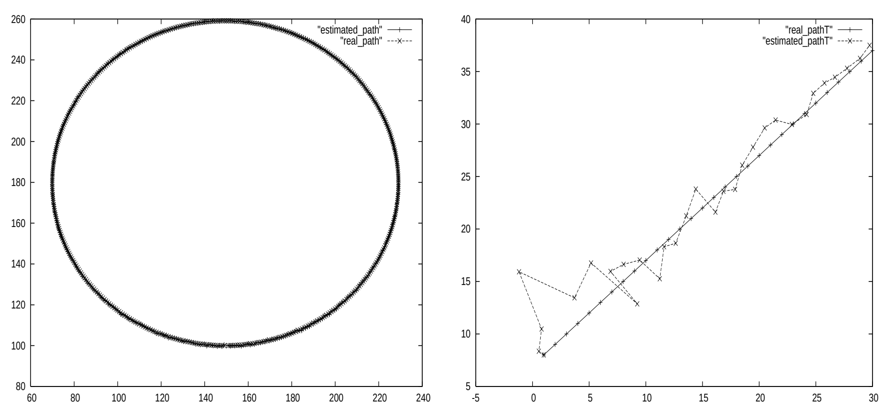


Figure 4: Real path vs. estimated path, different non-constant signal strength variance for every AP.

points of the triangulation. The average edge length of the triangulation is 25 units. We illustrate the comparison of the real and estimated paths in Fig. 5(a) and 5(b), for a circular and linear trajectory respectively (the error magnitude is similar in both cases, but more visible in Figure 5(b)). We note that the algorithm again uses (2) and (9) from Section 2; however, the values of the signal characteristics at the estimated node location are based on an interpolation using the known values of the triangulation.



(a) First example (b) Second example

Figure 5: Real path vs. estimated path based on a Delaunay triangulation interpolation.

5 Conclusion and Future Work

In this paper, we presented a filtering algorithm for the localization of mobile nodes, without any constraint or assumption about the node mobility model. Our approach relies on an analytical evaluation of the perturbation of the received signal strength from IEEE 802.11 access points. Based on signal strength measurements and a signal map, the algorithm aims to correct the estimated node position in real time. The proposed method has the advantage of being based on analytical calculations, which give closed formulas for the position estimates. Therefore, the implementation of the algorithm is simple and suitable for nodes of limited computational power, while still sufficiently accurate for use in indoor environments. We evaluated the performance of the algorithm with numerical simulations. In future work, we plan to extend this method to take under consideration the orientation of the antenna in order to derive more precise location estimates. Moreover, it would be interesting to design a system where data collected during the system operation can dynamically refine the initial training. Finally, we intend to investigate the system performance using real measurements.

References

[1] CGAL. http://www.cgal.org/.

[2] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-based location sensing using wireless Ethernet. In *Proceedings of the Eight ACM International Conference on Mobile Computing and Networking (MOBICOM 2002)*, Atlanta, GE, September 2002. ACM Press.

[3] Lauri Aalto, Nicklas Gothlin, Jani Korhonen, and Timo Ojala. Bluetooth and wap push based location-aware mobile advertising system. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 49–58. ACM Press, 2004.

[4] B. Delaunay. Sur la sphère vide. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7:793–800, 1934.

[5] Eiman Elnahrawy, Xiaoyan Li, and Richard Martin. The Limits of Localization Using Signal Strength: A Comparative Study. In *SECON*, Rutgers, October 7th 2004.

[6] P Faurre. *Navigation inertielle optimale et filtrage statistique*. Dunod, 1971.

[7] J. Hightower, R.Want, and G. Borriello. SpotON: An indoor 3D location sensing technology based on RF signal strength, February 2000. UW CSE.

[8] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the AMSE, Part D, Journal of Basic Engineering*, 82:35–45, 1960.

[9] N. B. Priyantha, A. K. L. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 1–14, Rome, Italy, July 2001.

[10] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil. Landmarc: indoor location sensing using active rfid. *Wirel. Netw.*, 10(6):701–710, 2004.

[11] Nissanka Bodhi Priyantha and Anit Chakraborty and Hari Balakrishnan. The Cricket Location-Support System. In *6th ACM MOBICOM*, Boston, MA, August 2000.

[12] P. V. Bahl and Padmanabhan. Radar: An In-Building RF-based User Location and Tracking System. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Tel Aviv, Israel, March 2000.

[13] Roy Want and Andy Hopper and Veronica Falcao and Jon Gibbons. The Active Badge Location System, 1992. ACM Transactions on Information Systems.

[14] S. Arulampalam and S. Maskell and N. Gordon and T. Clapp. A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. IEEE Transactions of Signal Processing, Vol. 50(2), pages 174-188, February 2002.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399