International Journal of Hybrid Information Technology
Vol.2, No.2, April, 2009

Barycentric Location Estimation for Wireless Network Indoors Localization

GuoDong Kang[1,2], Tanguy Pérennou[2], and Michel Diaz[2]
[1] *School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China*
[2] *LAAS-CNRS ; ISAE ;Université de Toulouse ; F-31077 Toulouse, France*
guodong.kang@isae.fr

Abstract

The localization requirements for mobile nodes in wireless (sensor) networks are increasing. However, most research works are based on range measurements between nodes which are often oversensitive to the measurement error. In this paper we propose a location estimation scheme based on moving nodes that opportunistically exchange known positions. The user couples a linear matrix inequality (LMI) method with a barycenter computation to estimate its position. Simulations have shown that the accuracy of the estimation increases when the number of known positions increases, the radio range decreases and the node speeds increase. The proposed method only depends on a maximum RSS threshold to take into account a known position, which makes it robust and easy to implement. To obtain an accuracy of 1 meter, a user may have to wait at the same position for 5 minutes, with 8 pedestrians moving within range on average.

1. Introduction

Nowadays wireless networks are going deep into people's everyday life. And a lot of mobile nodes are carried by pedestrians for communication needs. Therefore how to have accurate knowledge of geographical positions of wireless nodes is becoming more and more important for the pedestrian users [1]. GPS-based solutions perform very well in outdoor environments but not in indoor or urban environments. Alternate solutions have been investigated in the context of Wireless Sensor Networks. However, at the present time most research about wireless sensor localization concentrates on triangulation approaches which estimate the user node's position with respect to a few beacons by means of received signal strength (RSS), time of arrival (TOA), angle of arrival (AOA) or hop count method [2] [3] [4] [5]. However, these methods either lack of localization accuracy (such as RSS) or are very sensitive to environment (such as TOA, AOA and hop count method) [6].

In this paper we simply exploit the information that a user node is located within radio range of other nodes or not. By using the linear matrix inequality (LMI) method, we compute the optimum value of the range intersection that the user lies in rather than estimate the geographic distance between nodes. To achieve precise enough location estimation, we assume the user has the ability to wait for a period of time at the same place. Furthermore, in the process of localization, this paper introduces a barycenter-based method: the barycenter of a set of optimum values collected by the user during the waiting time is regarded as the final location estimation. Our method is tested using Matlab simulation based on a specific pedestrian random mobility model.

The paper is organized as follows. In Section 2 we describe the main idea of our localization method involving corresponding mathematical model and error expression. The detailed description of our simulation is given in Section 3. We perform our analysis on the simulation in Section 4. Conclusions and future work are presented in Section 5.

2. Localization algorithm

In this paper we consider that the coverage of a node's signal in two dimensions can be bounded by a disc of radius R. Let's consider the following situation. If the user can detect the signals from two different nodes at the same time, the user can ensure that he is lying in the intersection of the two nodes' coverage, as shown in Figure 1. The more intersecting circles there are, the smaller the intersection is. So if a sufficient amount of nodes know their exact position and can be detected by the user, the possible area the user lies in will be small enough.

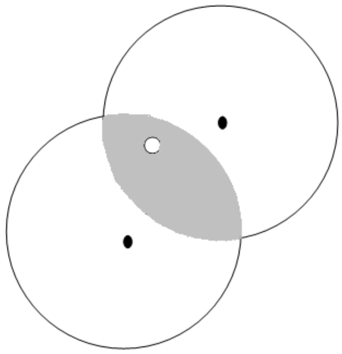


Figure 1. Black dots represent moving nodes knowing their exact position; the white dot represents the user who wants to know his position. Circles represent the maximum range of the moving nodes radio signal.

Throughout the remainder of this paper we will denote the position of the user as $P_u = (x_u, y_u)$, the position of moving node N_i as $P_i = (x_i, y_i)$. The geometry relations among the nodes can be expressed in mathematical form as the following set of constraints for N moving nodes:

$$\|P_u - P_i\| \leq R \quad \forall i, 1 \leq i \leq N$$

Looking at Figure 1, the optimum of the shaded area appears to be a good estimation of the user's position. Searching for such an optimum can be taken as a linear optimum program which satisfies a series of constraints $\|P_u - P_i\| \leq R$. In [7] and [8], a method is introduced to deal with the optimum question. The preceding constraint is firstly transformed into a form of linear matrix inequality (LMI):

$$\|P_u - P_i\| \leq R \rightarrow \begin{bmatrix} R & 0 & x_u - x_i \\ 0 & R & y_u - y_i \\ x_u - x_i & y_u - y_i & R \end{bmatrix} \geq 0$$

Let LMI_u be the user location estimation computed by solving the above set of inequalities for $1 \leq t \leq N$ with a LMI solver.

In fact, when the user waits at the same place and nodes move around him, he progressively collects a set of optimum values. After a while, the barycenter of the previously collected optimum values provides an even better estimation. Let $B_u(t)$ be the barycenter estimation at time t:

$$B_u(t) = \frac{1}{t}\sum_{k=1}^{k=t} LMI_u(k)$$

We also define E_{LMI} and E_B as the estimation error for the LMI and the barycenter estimation respectively:

$$E_{LMI} = \|LMI_u - P_u\| ;\ E_B = \|B_u - P_u\|$$

3. Simulation setup

3.1. Simulation introduction

All our simulations are executed in the Matlab 2007b environment. Matlab provides the LMI Lab toolbox to solve LMI problems. Though some other efficient algorithms have been proposed to solve similar optimum problems, e.g. [9] which is based on an efficient SDP relaxation method, this paper will not adopt them here, since time-efficiency is not an issue for our simulations and the optimum toolbox in Matlab is convenient to use.

In all following simulations, we assume that the user locates at the origin point of the coordinate system while other nodes move around him. We assume the user can wait at the same position for a period of time long enough to obtain good location estimation, i.e. the final error can meet his requirement. In this paper the waiting time is set to be 5 minutes.

3.2. Random pedestrian mobility model

In this Section we introduce the Random Pedestrian Mobility Model to predict the positions of nodes carried by pedestrians at simulation time. This model does not aim at being a realistic model such as social mobility models [10] but rather it provides a simple random model with speed and direction properties closer to a pedestrian than a free particle.

In all our simulations, nodes are assumed to be carried by pedestrians. The user's 5-minute waiting time is divided into 300 small time slots of one second. During each time slot, the trajectory of the pedestrian carrying node N_i is considered as rectilinear and uniform: speed and direction do not change. At time slot $t \geq 1$ the position $P_i(t)$ of node N_i is a function of its previous position and its current walking speed $v_i(t)$ and direction $\theta_i(t)$:

$$P_i(t) = \begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix} = \begin{bmatrix} x_i(t-1) + v_i(t) \cdot \cos\theta_i(t) \\ y_i(t-1) + v_i(t) \cdot \sin\theta_i(t) \end{bmatrix}$$

Now the critical point lies in how to adjust the speed $v_i(t)$ and the direction $\theta_i(t)$ of every time slot automatically and randomly. According to statistical data about pedestrians in [11], a person's walking speed obeys to a normal probability distribution $N(\mu, \sigma)$. In this paper we set the mean value of walking speed μ equal to 1.2 m/s and its standard deviation σ is 0.2 m/s. A new speed $v_i(t)$ is produced at each time slot t according to the normal law:

$$v_i(t) \sim N(1.2, 0.2)$$

Nodes could simply choose their next time's direction θ uniformly in $[0, 2\pi]$ as in the case of a free particle movement. However, this would result in a very erratic trajectory. To obtain smoother pedestrian trajectories, we take into account the fact that in most cases a person moves forward with a field of vision of approximately π. This means a person going forward has a very large probability to choose his next step's direction in a small sub-range of $[0, \pi]$ in front of him. We propose to derive a new mobility model where the next time's direction is tightly related to the former time's direction and obeys to normal law. In such a model, the direction used during time slot t is centered on the direction at time slot $t-1$:

$$\theta_i(t) \sim N(\theta_i(t-1), \sigma)$$

By definition of the normal law, in our mobility model the probability that the user chooses his next direction $\theta(t+1)$ in the range of $[\theta(t) - 2\sigma, \quad \theta(t) + 2\sigma]$ in front of him is about 0.95. After some experiments, we decided to set σ to π/6 which gives smoother trajectories as shown in Figure 2.

We will hereafter refer to the mobility model described above and built by periodically drawing a new pedestrian speed in a normal law and a new direction in a normal law centered of the previous direction as the Pedestrian Random Mobility Model. We will use this model with the parameters of Figure 2.

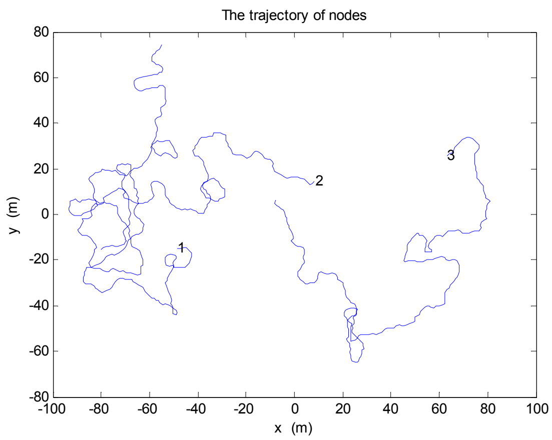


Figure 2. Three 5-minute trajectories generated with the Random Pedestrian Mobility Model (μ = former direction; σ = π/6).

3.3. Bounding the movements

In order to observe the variation of error conveniently when nodes move, each simulation uses the same number of nodes in the circle of the user's range. If at some time a node reaches the boundary of the range of user, it will be obliged to randomly draw its direction again and again to ensure it only moves in the range of user. In fact, here the circular boundary of the range plays a role of wall.

4. Simulation results

4.1. LMI location estimation

In this Section we focus on optimum values computed by the Matlab LMI toolbox and do not use the barycenter of the collected optimums. At each time step *t*, the location estimation LMI_{t_k} is the optimum computed by the LMI toolbox.
The range of nodes and user are both assumed to be 10 meters. At the beginning of the simulation, 50 nodes are distributed in the range of the user randomly and uniformly. These nodes then move independently for 5 minutes according to the Pedestrian Random Mobility Model defined above.
The user position is estimated every second as shown in Figure 3. The variation of the estimation positions are plotted with different RGB values which change from red to blue gradually over time.

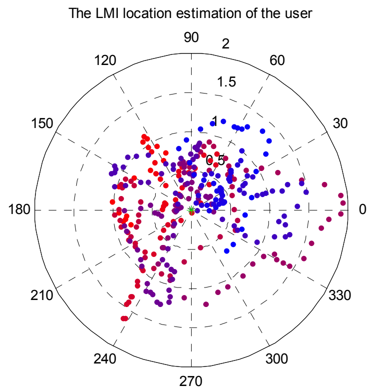


Figure 3. The distribution of LMI location estimations in polar coordinates. The user is located at the center.

We can observe that these estimations are rather close to the user position (< 2m). This is further illustrated by Figure 4, which shows that on average the LMI location estimation error E_{LMI} is less than 0.8 meters and in any case less than 2 meters.

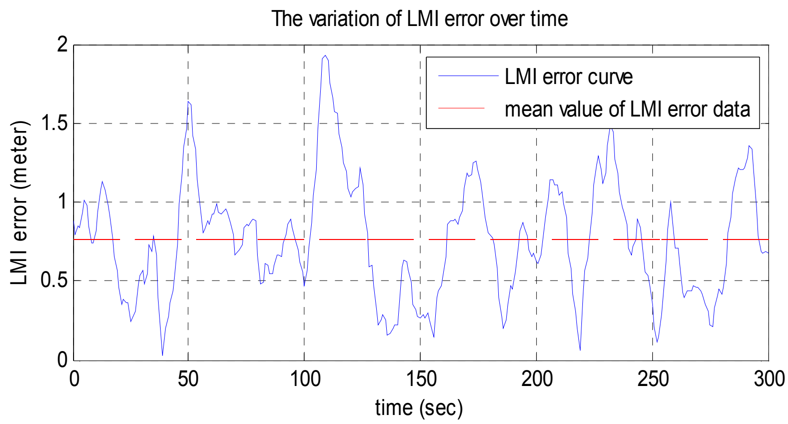


Figure 4. The LMI error variation curve over time.

4.2. Barycenter location estimation

As explained in Section 2, the LMI estimations are progressively collected by the user who can compute the barycenter of the LMI estimations collected so far. The results of this barycenter-based location estimation at each time t are far less scattered than the LMI estimations, as shown in Figure 5.

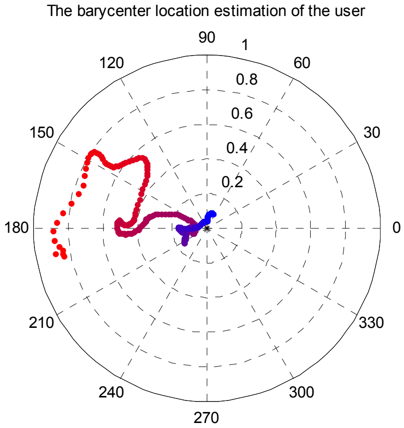


Figure 5. The distribution of barycenter location estimations in polar coordinates. The user is located at the center.

The barycenter error E_B converges to a very satisfying value of less than 0.2 meters after a period of time smaller than 2 minutes, as shown in Figure 6. The accuracy of the barycenter estimation is better than the LMI estimation.

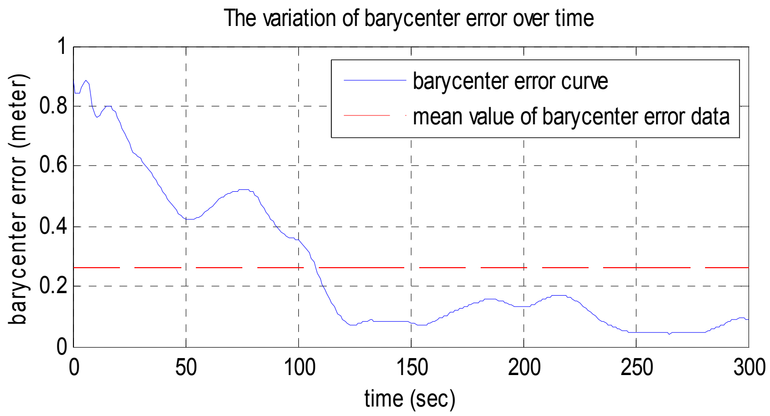


Figure 6. The barycenter error variation curve over waiting time.

To further survey the barycenter error, we carried out another 49 simulations which are also based on 50 nodes and use the same parameters but use different random seeds. All of these experiments achieve similar results as the first experiment. Figure 7 gives the mean values of the LMI and barycenter errors in the 50 experiments. We observe that the barycenter method systematically gives better location accuracy.

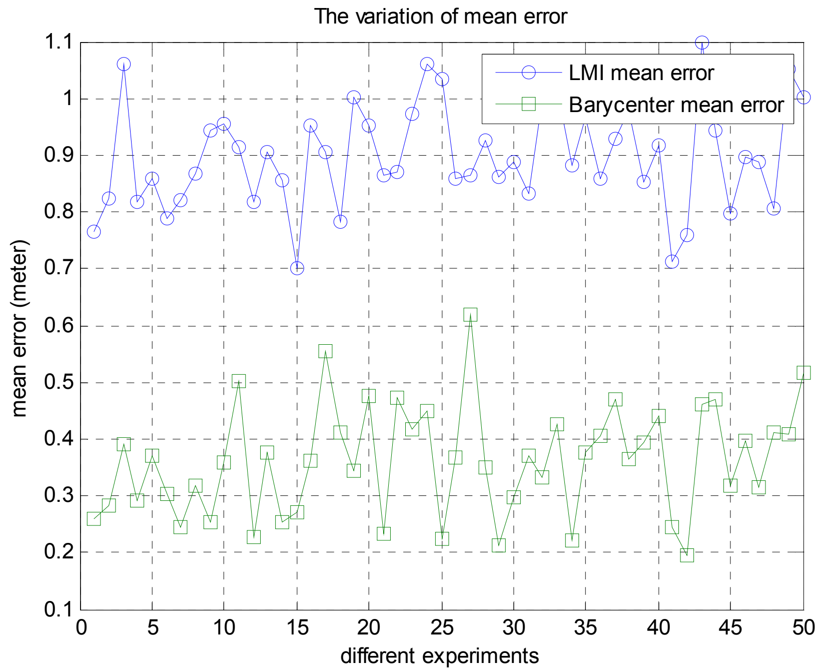


Figure 7. Comparing LMI and barycenter errors across 50 experiments.

4.3. Impact of the simulation parameters

In terms of the above simulation model, there are several parameters that may influence the final location estimation accuracy, that is, the node density, the range level of the nodes and the moving speed of the nodes. These parameters are varied and discussed separately throughout the remainder of this Section.

4.3.1. Node density impact. Here, the node density is equivalent to the number of nodes in the range of the user. This definition is consistent with [12], in which the network density in a region of area A is express as:

$$\mu(R) = \frac{N \cdot \pi \cdot R^2}{A}$$

where R is the range of the nodes and N is the number of nodes in the region of area A. When A is the range of the user, the density is equal to the node number.

As in Section 4.1 and 4.2, we keep a specific number of nodes moving in the range of user all the time so that the node density in the presented simulation doesn't change. The node speed obeys a normal law $N(1.2, 0.2)$ as explained in Section 3.2.

As expected, we can see in Figure 8 that as the number of nodes increase, the mean barycenter error decrease gradually. When there are 8 nodes that can be used, most of the mean barycenter errors in 50 experiments are below one meter. When there are 50 known nodes (typically in a crowded area), almost all the mean barycenter errors are smaller than 0.5 meter.

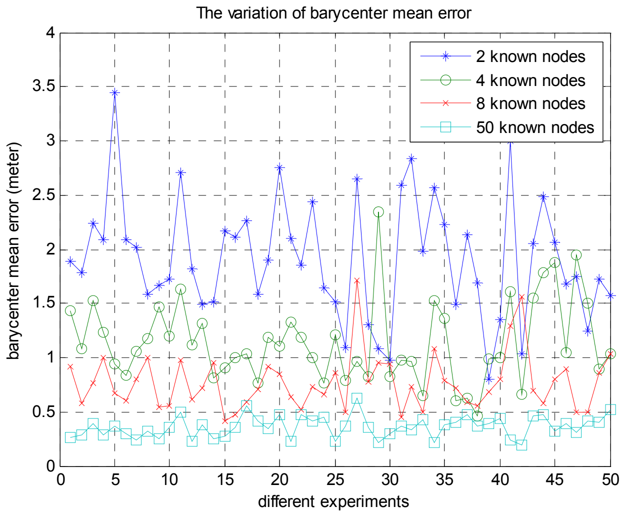


Figure 8. Node density impact on 50 experiments: the location accuracy increases with the number of nodes within range.

4.3.2. Node range impact. In order to observe the effect of node range on the location accuracy, another 50 experiments were carried out changing the range of nodes to 30 meters. The node density is set to 50 and the mean speed is drawn in a $N(1.2, 0.2)$ normal law. Figure 9 compares the results for the original 10-meter range and the new 30-meter range. As shown in the figure, the mean barycenter errors increase with the range in every experiment. Experiments with different node densities exhibit the same trend.

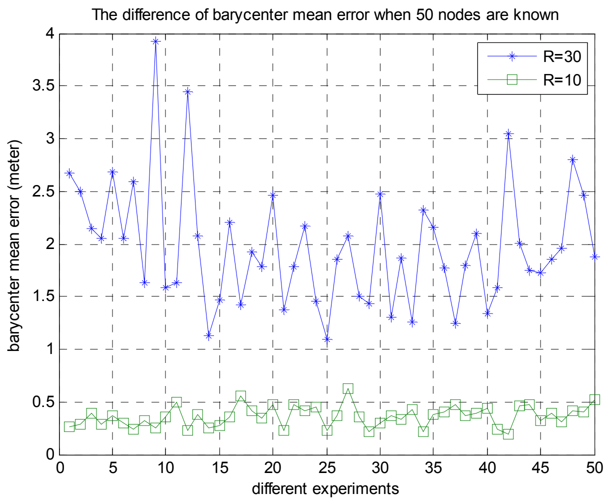
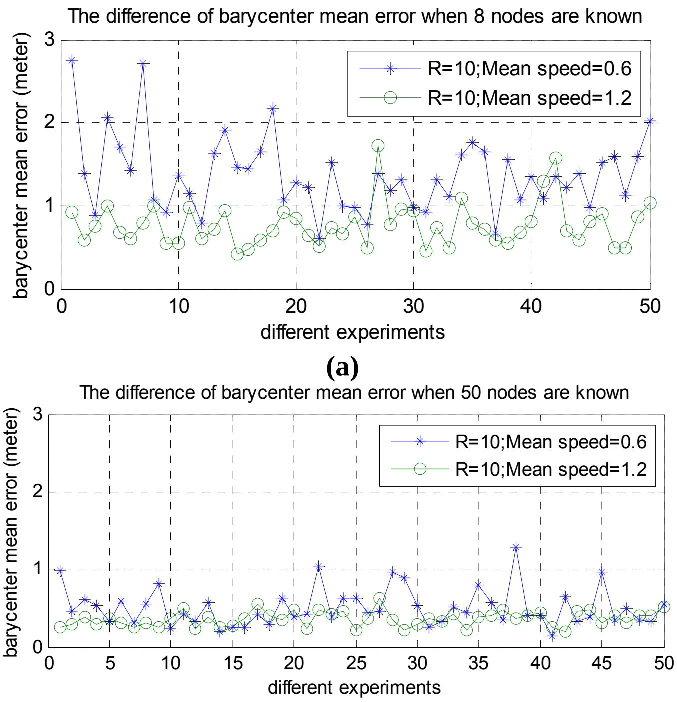


Figure 9. Node range impact on 50 experiments: the location accuracy decreases as the range increases.

4.3.3. Node speed impact. We now study the impact of the mean speed of the movement model, setting it to 0.6 m/s (a slow pedestrian). The other parameters remain unchanged: a speed variance of 0.2 m/s, a range of 10 meters and a node density of 8 known nodes and 50 known nodes. As shown in Figure 10, the mean barycenter error tends to decrease as the nodes move faster. This indicates that the speed of nodes will influence the final barycenter error. If the node moves too slowly, the barycenter error curve will take more time to converge to a better result.



(b)

Figure 10. Node speed impact on 50 experiments: on average, the location accuracy increases with the speed. (a) is the 8-node case; (b) is the 50-node case.

5. Conclusions and future work

In this paper we have proposed a location estimation scheme based on collaborating moving nodes that opportunistically exchange known positions. Using a linear matrix inequality (LMI) method coupled with a barycenter computation, a user can compute an estimation of its own position. Simulations have shown that the accuracy of the location estimation increases when the number of known positions increases, the radio range decreases and (to a smaller extent) when nodes speeds increase.

The advantage of the proposed method is that it does not need to cope with over-sensitive measures such as RSS or TOA. Rather, it can accept a maximum RSS threshold to take into account a given position information, which is easy to implement. To obtain an accuracy as good as 1 meter with a good probability, a node may have to wait at the same position for up to 5 minutes, with 8 pedestrians moving around him on average.

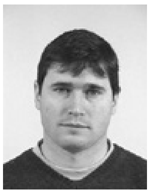
The method can be applied to situations where nodes are densely distributed and do not have specific positioning equipment aside from wireless radios. This includes a wide range of social networks involving smart phones or PDAs, as well as mobile adhoc or sensor networks. Future work in this direction includes taking into account more realistic pedestrian movements, in indoors as well as urban scenarios mixing pedestrians and vehicles (typically buses). We also want to relax the “known position” assumption: in the present work, a moving node provides its exact position to the user. In practical situations, an exact position may be available from time to time, using well known beacons for instance, but most of the time, moving nodes may only be able to send estimates of their positions.

References

[1] C.-Y. Chong, F. Zhao, S. Mori, and S. Kumar, "Distributed tracking in wireless ad hoc sensor networks," in *Proceedings of the Sixth International Conference of Information Fusion,* 2003, pp. 431-438.
[2] P. Bahl and V. N. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *Proceedings of IEEE INFOCOM 2000*, 2000, pp. 775-784.
[3] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of ACM MOBICOM 2000,* 2000, pp. 32-43.
[4] K. Sayrafian-Pour and D. Kaspar, "Source-assisted direction estimation inside buildings," in *Proceedings of IEEE INFOCOM 2006,* 2006, pp. 1-8.
[5] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA '02)*, 2002, pp. 112-121.
[6] E. Elnahrawy, X. Li, and R. P. Martin, "The limits of localization using signal strength: a comparative study," in *Proceedings of the 1st IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004)*, 2004, pp. 406-414.
[7] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of INFOCOM 2001*, vol. 3, 2001, pp. 1655-1663.
[8] L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System & Control Theory (Studies in Applied Mathematics, Volume 15).* Society for Industrial and Applied Mathematics (SIAM), 1994.
[9] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Third International Symposium on Information Processing in Sensor Networks (IPSN 2004)*, 2004, pp. 46-54.
[10] M. Musolesi and C. Mascolo, "Designing mobility models based on social network theory," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 3, pp. 59-70, July 2007.
[11] B. H. Kaye, *A random walk through fractal dimensions*. New York, NY, USA: VCH Publishers, 1989.
[12] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann, "Scalable coordination for wireless sensor networks: Self-configuring localization systems," in *Proceedings of the 6th International Symposium on Communication Theory and Applications (ISCTA 2001)*, 2001, pp. 1-6.

Authors

Guodong KANG was born in Heze, China, in 1981. He is a PhD candidate of Université de Toulouse, ISAE, France as well as Northwestern Polytechnical University, Xi'an, China. His research interest is wireless communication and wireless sensor network localization for indoors.



Tanguy Pérennou is a professor in the Applied Mathematics and Computer Science Department at ENSICA. His research interests range from distributed computing to wireless networks. His teaching activities also include object-oriented and 3D programming as well as Web services. He is currently involved in characterization of losses in wireless networks and IP-level wireless networks emulation.

Michel DIAZ is Director de Research at CNRS, leads the OLC Research Group at LAAS, and is working on methodologies, techniques and tools for designing distributed systems. In 1990, he was a one year visiting staff at the University of Delaware at Newark and of California at Berkeley. He was the manager of the EC ESPRIT project SEDOS on the development of formal description techniques for protocols and of the EC IST project GCAP on new architecture and end-to-end services and protocols for the future internet. He now leads the RTP French research coordination structure on Communication Networks. He served as Chairman or member of many Program Committees, edited or co-edited 11 books and published more than 150 technical papers.