

# An Efficient Algorithm for Mobile Localization in Sensor Networks

Xin-Min Xu<sup>1</sup>   Yao Chen<sup>1</sup>   Wen-Yao Xu<sup>2</sup>   Fang Gong<sup>2</sup>

<sup>1</sup>Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China

<sup>2</sup>Electrical Engineering Department, University of California, Los Angeles 90095, USA

**Abstract:** Node localization is a fundamental problem in wireless sensor network. There are many existing algorithms to estimate the locations of the nodes. However, most of the methods did not consider the presence of obstacles. In practice, obstacles will lead to blockage and reflection of communication signals between sensor nodes. Therefore, the presence of obstacles will greatly affect the localization result. In this paper, we implement an obstacle-handling algorithm based on the localization tool developed by MIT. The experimental result shows that the enhanced algorithm can reduce the average distance error by up to 46%, compared to the original algorithm.

**Keywords:** Wireless sensor network, automatic localization, intelligent computing, optimization, simulation.

## 1 Introduction

As localization in sensor networks<sup>[1]</sup> is applicable for miscellaneous of applications such as object tracking<sup>[2]</sup>, habitat monitoring<sup>[3]</sup>, urbanization control<sup>[4]</sup>, localization has always been a fundamental problem in sensor networks<sup>[5]</sup>. Much work is done on different localization techniques such as trilateration<sup>[6]</sup>, global optimization<sup>[7]</sup>, multi-lateration<sup>[8]</sup>, and probabilistic approach<sup>[9]</sup>. However, not much attention is paid to localizing nodes in the presence of obstacles.

Obstacles introduce error in localization by blocking the direct path between two nodes and causing a reflective path between them. This leads to a larger measured distance than the true distance, thus causing inaccuracies in position estimation. Additionally, this kind of uncertain interference makes the optimal path planning<sup>[10,11]</sup> in wireless sensor networks increasingly challenging due to unknown environment<sup>[12]</sup>. In this paper, we propose obstacle-handling algorithms to remove the reflective paths. We implement our algorithm on an existing trilateration localization system<sup>[6,13]</sup>. This localization algorithm differs from other algorithms in that it considers the possibility of flip ambiguities during trilateration due to measurement noise, and it does not require any position information. We run the enhanced localization algorithm onboard a sensor network. The experiment result shows that the localization result can be improved by up to 46% with the addition of our obstacle-handling algorithm.

The organization of this paper is as follows. Section 2 briefly describes the localization system which our algorithm is built on, and the impact of obstacles. Section 3 describes the proposed obstacle-handling algorithm. Section 4 describes the experimental results. And Section 5 concludes the paper.

## 2 Overview

### 2.1 Previous localization method

In this paper, we build our localization tool on a robust localization method<sup>[6]</sup>. This algorithm is mainly based on two properties of a quadrilateral which is fully connected with 6 edges (assuming there are not three nodes that are collinear):

1) The relative positions of the four nodes are unique up to a global rotation, translation, and reflection. This is also known as globally rigid.

2) Any two globally rigid quadrilaterals sharing three vertices from a 5-vertex subgraph is also globally rigid. By induction, any number of quadrilaterals chained in this manner form a globally rigid graph.

In addition, this algorithm further restricts the quadrilaterals to be robust, in short, to be a robust quad. Robust quad is defined as a fully-connected quadrilateral whose four sub-triangles satisfy

$$b \sin^2 \theta > d_{\min} \quad (1)$$

where  $b$  is the length of the shortest side,  $\theta$  is the smallest angle, and  $d_{\min}$  is a threshold chosen based on the measurement noise. This restriction reduces flip error, which is illustrated in Fig. 1. In Fig. 1, when  $\theta$  is small, if the measured distance of nodes A and D becomes slightly shorter, then the node D will be located at D'. This is called flip error. So (1) bounds the worst-case probability of a flip error for each triangle.

The localization procedure uses robust quad as a starting point, and localize other nodes by chaining connected robust quads. Between two robust quads, if there are three common nodes, the relative location of the two quads can be computed. As shown in Fig. 2, quad {A, B, C, D} and quad {E, B, C, D} have 3 common nodes {B, C, D}, then we can compute the relative location of these two quads. And so do the quad {A, B, C, D} and quad {A, B, F, D}. In this way, the relative locations among {A, B, C, D, E, F} are determined. Such group of quads is called a cluster.

And if two clusters have 3 common nodes, we can compute the relative location of these two clusters, such as cluster 1 {A, B, C, D, E, F} and cluster 2 {A, M, C, N, O, F}. In this way, the relative location of all the sensor nodes can be computed.

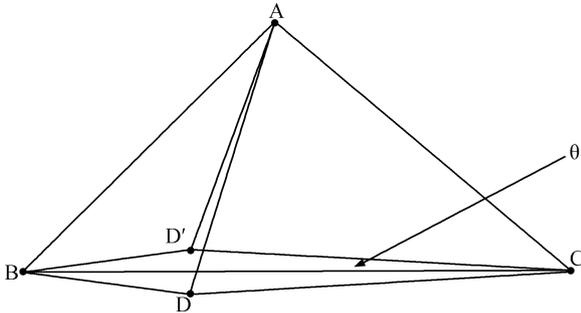


Fig. 1 Flip error

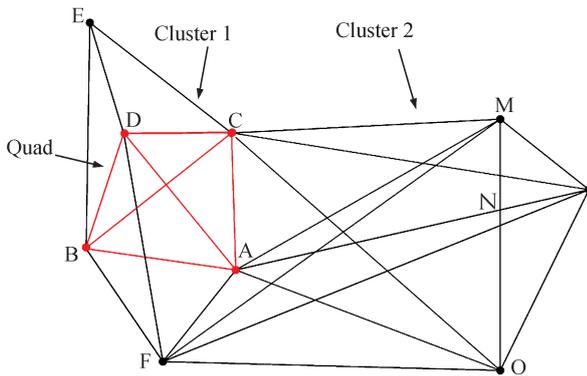


Fig. 2 Quadrilateral and cluster localization

### 2.2 Impact of obstacles

The localization algorithm proposed in [6] does not particularly consider the impact of obstacles. In practice, obstacles lead to the blockage and reflection of communication signals among sensor nodes. When there is an obstacle between two nodes, the nodes cannot measure the direct distance between each other, and they get much longer measurement caused by reflective path. Such wrong measurement leads to localization error, and the error will propagate during trilateration. Therefore, the existence of obstacles will greatly affect the localization result.

To reduce the impact of obstacles, we have to process the measured inter-node distances to remove the reflective paths. As long as there are enough measurements to localize all the nodes, missing one or more edges would not cause any problem. So the goal of our algorithm is to remove as many reflective paths as possible. The detail of our algorithm is discussed in Section 3. Our overall localization procedure is as follows:

- 1) Measure distance between sensor nodes.
- 2) Remove reflective paths.
- 3) Use robust quad method to estimate the location of sensors nodes.

## 3 Algorithm

In this section, we discuss two algorithms: shortest path algorithm (SPA) and most negative residue algorithm (MNRA) to remove reflection edges.

### 3.1 Shortest path algorithm

We use  $d(A, B)$  to denote the distance measured between nodes A and B,  $d_{\min}(A, B)$  to denote the length of the shortest path that connects A and B through zero (direct distance) or multiple nodes (multi-hop distance). Our approach is based on the following theorem, which is an extension from the law of triangles:

For nodes A and B, if

$$d(A, B) > d_{\min}(A, B) \tag{2}$$

then  $d(A, B)$  must be a reflective path.

Reflection leads to a longer distance measurement. When there is no obstacle,  $d(A, B)$  is the direct distance between A and B, it is obviously the shortest possible path connecting A and B, so  $d(A, B) = d_{\min}(A, B)$ . However, in the presence of obstacles, the direct path between two nodes may be blocked and only the reflective path can be measured, as shown in Fig. 3. In this case, there may exist other path(s) connecting A and B in a multi-hop fashion which is shorter than  $d(A, B)$ , e.g., the path  $A \rightarrow C \rightarrow D \rightarrow B$  in Fig. 3. So when  $d(A, B) > d_{\min}(A, B)$ ,  $d(A, B)$  must be a reflection. In our implementation, the shortest path between every pair of nodes is found using the Floyd-Warshall algorithm. Each inter-node measured distance is then compared against its corresponding computed shortest path. If the measured distance is longer than the shortest path, such edge measurement must be a reflective path and thus is removed.

In any physical environment, the distance measured between two nodes will be affected by the presence of noise. To prevent our algorithm from removing a non-reflective edge, we introduce a noise tolerance factor  $\alpha$ . This parameter is user-defined, depending on how noisy the environment is. With noise tolerance, (2) can be rewritten as

For nodes A and B, if

$$d(A, B) > (1 + \alpha)d_{\min}(A, B) \tag{3}$$

then  $d(A, B)$  must be a reflective path.

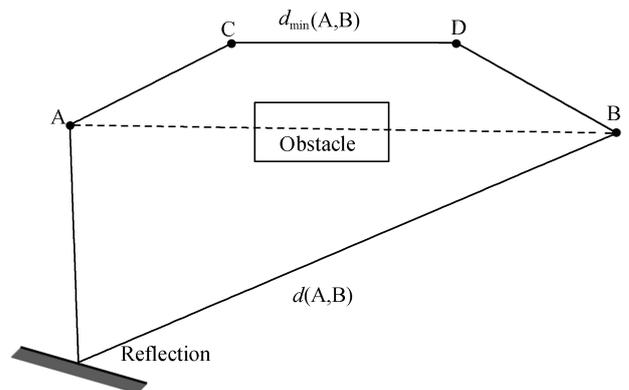


Fig. 3 Shortest path and reflective path

Our shortest path algorithm does not guarantee that all the reflective paths are removed. However, it can remove the few most “significant” reflective paths, i.e., the path that is much longer (about two times) than the direct distance. Moreover, the shortest path algorithm will not remove any direct edges, thus, it could only improve but not harm the existing algorithm.

### 3.2 Most negative residue algorithm

To reduce more reflection, we further introduce the most negative residue algorithm. After SPA, we first estimate the locations of all nodes, then calculate the residual of each measured distance. We define  $R(A, B)$  as the residual of  $d(A, B)$ ,  $R(A, B)$  reflects the difference between measured distance and estimated distance, shown as

$$R(A, B) = d(A_e, B_e) - d(A, B) \quad (4)$$

where  $A_e$  and  $B_e$  are estimated locations of A and B, and  $d(A_e, B_e)$  is distance between  $A_e$  and  $B_e$ . As discussed above, the outliers come from reflection, then the distance must be larger than the actual distance. Therefore, the residual of a reflection edge must be negative. The measured distance with most negative residual is very likely to be a reflection. Therefore, we may iteratively remove the measured distance with most negative residual and recalculate the locations until the most negative residual is smaller than a desired bound. In practice, the bound can be defined by users. Fig. 4 shows the flow of negative residual algorithm.

- |  |
|--|
| <ol style="list-style-type: none"> <li>1. Estimate location</li> <li>2. while (most negative residual &lt; desired bound)</li> <li>3.   remove distance with most negative residual</li> <li>4.   remove distance with most negative residual</li> <li>5.   estimate location</li> </ol> |
|--|

Fig. 4 Most negative residual algorithm flow

## 4 Experimental results

### 4.1 Simulation results

To validate our algorithms, we randomly generate a number of sensors and obstacles. Figs. 5 and 6 show the results of our algorithms. We can find that our algorithms remove the most of reflections.

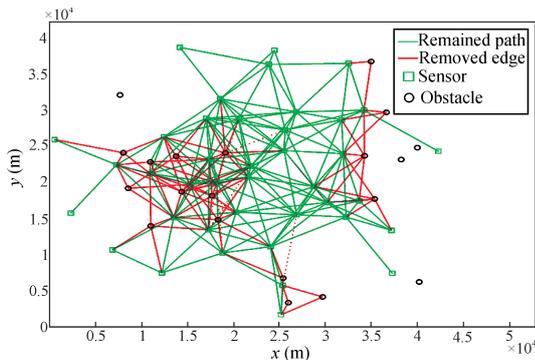


Fig. 5 Results after SPA

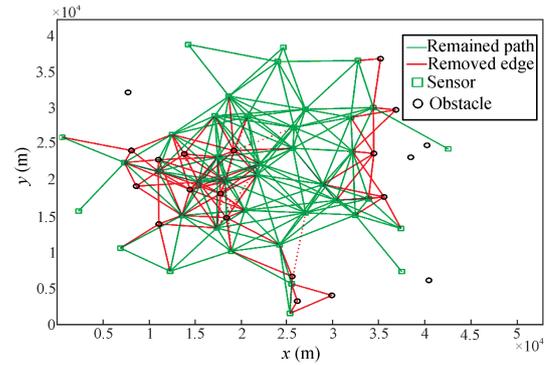


Fig. 6 Results after MNR

Table 1 shows the number of removed reflection by our algorithm for different number and size of obstacles. We can see that our algorithms remove the most reflections with only a small number of wrong removals (remove a measured distance which is not a reflection).

### 4.2 Real measurement

We further verified our algorithm by implementing it onboard a network of up to 8 crickets, which is a hardware platform developed and supplied by MIT<sup>[6]</sup>. Due to the limited memory space, we could only implement one algorithm onboard. Crickets have the same hardware as the Mica2 Motes developed by Berkeley with the addition of an ultrasonic transmitter and receiver. This additional hardware enables the motes to measure inter-node distances using the time difference of arrival between Ultrasonic and radio-frequency signals. Fig. 7 shows one of the set-up of our experiment. We run our enhanced algorithm onboard a cricket at the bottom right hand corner. This cricket communicates with the host through a serial cable. The rest of the crickets are arranged in a semi-circular order, which is a setting that gives the best localization result without obstacles. We first used the original algorithm<sup>[6]</sup> to measure the motes' locations without obstacle present. This is our baseline, against which the performance of each algorithm is compared. We choose this instead of using the known ground truth as our baseline because here we are not focused on how well the original localization algorithm is, but how much the result could be improved by the additional obstacle-handling algorithm. So, simply comparing the enhanced algorithm to the original algorithm in a setting which gives the best localization result without obstacles would be sufficient for our purpose.

After finding the baseline localization result, we add obstacles to the center of the semi-circle and measure the data using the original algorithm developed by MIT, then the enhanced algorithm which handles obstacles. We also try to use different noise tolerance factor  $\alpha$ , ranging from 0 to 0.3. When  $\alpha$  is too small, our algorithm removes more edges so that some of the sensor nodes cannot be localized. When  $\alpha$  is too big, the algorithm does not remove any reflective path and no improvement is shown. We find that  $\alpha = 0.1$  provides the best result in our set-up, therefore we adopted this value. Our set-ups could be grouped into two test cases:

Table 1 Number of removed reflections

# Sensors	# Obstacles	Obstacle size	# Reflections	# Correct removals	# Wrong removals
200	10	300	17	17	2
200	10	700	74	74	2
200	50	300	266	264	3
200	50	700	191	190	2
200	70	300	282	276	4
200	70	700	245	236	7

Table 2 Comparison of absolute error and improvement between original and enhanced algorithm

Case	Edges removed	Localized nodes	Original error	Enhanced error	Improvement (%)	Average improvement (%)
1	2	8	13.9	7.53	46.0	
1	2	8	8.64	6.26	27.6	38.8
1	2	8	8.33	4.78	42.7	
2	1	6	8.29	7.49	9.59	
2	1	6	11.7	9.22	21.1	11.0
2	1	6	4.44	4.33	2.40	

- 1) With 8 notes;
- 2) With 6 notes.

Table 2 summarizes the error comparison in both cases. We see that in Case 1, the average localization improvement compared to baseline is 38.8%, and in Case 2 it is 11%. In the following sections, we pick one scenario from each case and provide a detailed analysis on it.

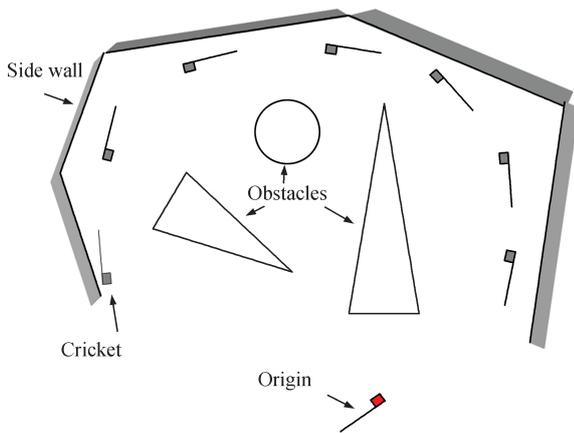


Fig. 7 Localization of crickets and obstacles in Case 1

### 4.3 Case 1: 8 notes

The location plot is presented in Fig. 8 and the measurement data is summarized in Table 3. Using the enhanced algorithm, the localization result is improved by 46%. Table 4 presents the inter-node distance ratio of Case 1 to baseline. In the presence of obstacles, edge (2, 6) and edge (3, 6) are removed by the enhanced algorithm. Edge (2, 6) is increased by 2.56 times in the presence of obstacles, and edge (3, 6) is increased by 2.03 times. Clearly, they are both reflection paths, thus need to be removed. Besides,

there are three other edges (1, 3), (3, 4) and (0, 2) that are increased by over 1.3 times after the placement of obstacles, but were not removed. This is because the multi-hop shortest path algorithm does not guarantee that all the reflections are removed, as mentioned in Section 3. However, we find that removing too many edges leads to a lack of information to localize all the nodes, so what our algorithm does is sufficient.

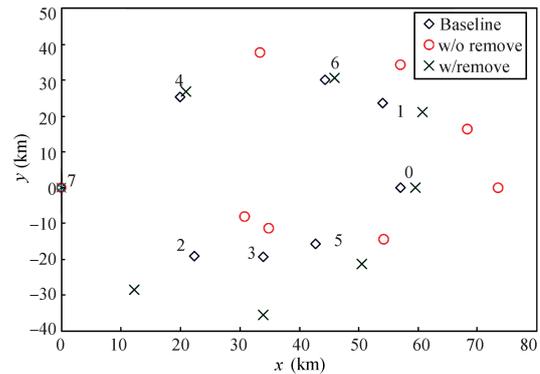


Fig. 8 Localization results compared with original method and the enhanced algorithm in Case 1

### 4.4 Case 2: 6 notes

Fig. 9 presents the location plot of Case 2. The numerical position measurement is summarized in Table 5. We find that the enhanced algorithm could improve the localization result by about 21.1%. Table 6 presents the inter-node distance ratio of Case 2 to baseline. The enhanced algorithm removes edge (1, 2), from all the measured edges. This edge is increased by 2.75 times after the placement of obstacles, therefore, it is legitimate to be removed. Similar to Case 1, there are two more edges that are increased by over 1.3 times and yet are not removed. Their ratio is 1.35 and 1.74.

Table 3 Residual comparison results between the original and enhanced algorithm in Case 1

	Node	0	1	2	3	4	5	6	7
Base line	$x$	57.1	54.1	22.4	34.0	20.0	42.8	44.4	0.00
	$y$	0.00	23.6	-19.1	-19.3	25.3	-15.7	30.1	0.00
Original algorithm	$x$	73.5	68.3	30.9	34.9	33.5	54.3	57.1	0.00
	$y$	-0.04	16.3	-8.06	-11.3	37.7	-14.4	34.3	0.00
	error	16.4	16.0	13.9	8.02	18.3	11.5	13.4	0.00
Enhanced algorithm	$x$	59.6	60.8	12.3	34.0	21.0	50.6	46.0	0.00
	$y$	0.00	21.1	-28.5	-35.5	26.8	-21.3	30.6	0.00
	error	2.50	7.15	13.8	16.2	1.80	9.60	1.68	0.00

However, we find that if an additional edge is removed by our algorithm in this test case, it will lead to at least one node not being localized. Therefore, what we have is sufficient. From the experiment result, we find that the proposed algorithm works better with more motes present. This is because the more motes we have, the more edges available to perform the multi-hop shortest path algorithm. Thus, a false path is easier to be detected.

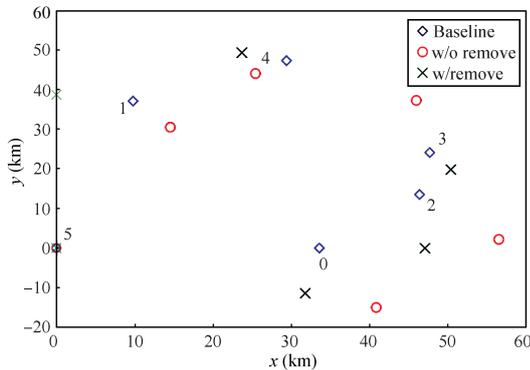


Fig. 9 Localization results compared between the original algorithm and the enhanced algorithm in Case 2

Table 4 Inter-node distance ratio of Case 1 (with obstacles) to its baseline (without obstacles)

Node	0	1	2	3	4	5	6	7
0	1.00	0.97	1.40	1.11	1.04	1.05	1.00	1.04
1	0.97	1.00	0.00	1.62	1.12	1.00	1.02	1.08
2	1.40	0.00	1.00	1.00	1.19	1.04	2.56	1.00
3	1.11	1.62	1.00	1.00	1.43	1.00	2.03	1.02
4	1.04	1.12	1.19	1.43	1.00	1.08	1.01	1.01
5	1.05	1.00	1.04	1.00	1.08	1.00	1.00	1.02
6	1.00	1.02	2.56	2.03	1.01	1.05	1.00	1.02
7	1.04	1.08	1.00	1.02	1.01	1.17	1.02	1.00

## 5 Conclusions

In this paper, we proposed obstacle-handling algorithms to remove reflective paths introduced by the presence of obstacles. We implemented our proposed algorithm on an existing trilateration localization method, and run it on-board a sensor network of crickets. We verified that with the addition of our obstacle-handling algorithm, the local-

ization result is improved by up to 46%. The improvement is larger with more motes present in a network.

Table 5 Residual comparison results between the original and enhanced algorithm in Case 2

	Node	0	1	2	3	4	5
Base line	$x$	33.6	9.80	46.4	47.7	29.4	0.00
	$y$	0.00	37.1	13.5	24.5	47.3	0.00
Original algorithm	$x$	40.9	14.6	56.5	46.0	25.4	0.00
	$y$	-15.0	30.5	2.18	37.3	44.0	0.00
	error	16.7	8.17	15.2	13.3	5.12	0.00
Enhanced algorithm	$x$	31.8	0.00	47.1	50.4	23.7	0.00
	$y$	-11.4	38.7	-0.03	19.8	49.3	0.00
	error	11.5	9.93	13.5	5.08	6.04	0.00

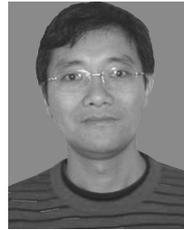
Table 6 Inter-node distance ratio of Case 2 (with obstacles) to its baseline (without obstacles)

Node	0	1	2	3	4	5
0	1.00	1.35	1.01	1.03	1.74	1.01
1	1.35	1.00	2.75	1.09	1.00	1.01
2	1.01	2.75	1.00	1.00	1.14	1.00
3	1.03	1.09	1.00	1.00	1.00	1.16
4	1.74	1.00	1.14	1.00	1.00	1.01
5	1.07	1.01	1.00	1.16	1.01	1.00

## References

- [1] L. Zhang, L. G. Liu, C. Gotsman, S. J. Gortler. An as-rigid-as-possible approach to sensor network localization. *ACM Transactions on Sensor Networks*, vol. 6, no. 4, pp. 1–21, 2010.
- [2] R. Peng, M. L. Sichitiu. Probabilistic localization for outdoor wireless sensor networks. *Journal Mobile Computing and Communications Review*, vol. 11, no. 1, pp. 53–64, 2007.
- [3] Q. C. Wang, Y. H. Gong, C. H. Yang, C. H. Li. Robust object tracking under appearance change conditions. *International Journal of Automation and Computing*, vol. 7, no. 1, pp. 31–38, 2010.
- [4] R. Sezwczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, D. Estrin. Habitat monitoring with sensor networks. *ACM Transactions on Wireless Sensor Networks*, vol. 47, no. 6, pp. 34–40, 2004.

- [5] B. D. Qu. Game modeling research for urbanization and epidemic control. *International Journal of Automation and Computing*, vol. 2, no. 1, pp. 13–19, 2005.
- [6] D. Moore, J. Leonard, D. Rus, S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, IEEE, New York, NY, USA, pp. 50–61, 2004.
- [7] L. Doherty, K. S. J. Pister, L. El Ghaoui. Convex position estimation in wireless sensor networks. In *Annual Joint Conference of the IEEE Computer and Communications Societies*, IEEE, vol. 3, pp. 1655–1663, 2004.
- [8] A. Savvides, H. Park, 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*, ACM, New York, NY, USA, vol. 6, pp. 112–121, 2002.
- [9] V. Ramadurai, M. L. Sichitiu. Localization in wireless sensor networks: A probabilistic approach. In *Proceedings of International Conference on Wireless Networks*, CSREA Press, Las Vegas, Nevada, USA, vol. 4, pp. 275–281, 2003.
- [10] J. Z. Lin, X. Zhou, Y. Li. A minimum-energy path-preserving topology control algorithm for wireless sensor networks. *International Journal of Automation and Computing*, vol. 6, no. 3, pp. 295–300, 2009.
- [11] L. He, Z. Chen, J. D. Xu. Optimizing data collection path in sensor networks with mobile elements. *International Journal of Automation and Computing*, vol. 8, no. 1, pp. 69–77, 2011.
- [12] T. K. Wang, Q. Dang, P. Y. Pan. Path planning approach in unknown environment. *International Journal of Automation and Computing*, vol. 7, no. 3, pp. 310–316, 2010.
- [13] A. Ward, A. Jones, A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, vol. 4, no. 5, pp. 42–47, 1997.



**Xin-Min Xu** is an associate professor at Zhejiang University, China.

His research interests include system-on-chip design and verification methodologies and implementation, dynamic reconfiguration theory and technology research, application of the embedded system, intelligent instruments.

E-mail: xuxm@zju.edu.cn (Corresponding author)



**Yao Chen** is a master student in the Information and Electronic Engineering Department at Zhejiang University, China, majored in electronic circuit and information system.

His research interests include system-on-chip design and development, and application of the embedded system.

E-mail: chenyz@zju.edu.cn



**Wen-Yao Xu** received his B.Sc. and M.Sc. degrees in electrical engineering from Zhejiang University, Hangzhou, China in 2006 and 2008, respectively. From 2008, he started his graduate study at the Department of Electrical Engineering and Computer Sciences, University of California at Los Angeles, USA, and is currently a Ph. D. candidate.

His research interests include embedded system design for wireless health, power management and smart grid.

E-mail: wxu@ee.ucla.edu



**Fang Gong** received his B.Sc. degree from Computer Science Department, Beijing University of Aeronautics and Astronautics, China in 2005. Also, he graduated from Computer Science Department at Tsinghua University with M.Sc. degree in 2008. After that he became a Ph. D candidate, in the Electrical Engineering Department, University of California, Los Angeles, USA.

His research interests include numerical computing and stochastic techniques for CAD, fast circuit simulation, yield estimation and optimization, numerics parallel and distributed computing.

E-mail: gong08@ee.ucla.edu