sDLS - Distributed Least Squares Localization for Large Wireless Sensor Networks

Ralf Behnke, Jakob Salzmann, Dominik Lieckfeldt, Dirk Timmermann
Institute of Applied Microelectronics and Computer Engineering
University of Rostock
18057 Rostock, Germany
{*ralf.behnke, jakob.salzmann, dominik.lieckfeldt, dirk.timmermann*}*@uni-rostock.de*

***Abstract*—Wireless Sensor Networks (WSNs) have been of high interest during the past couple of years. One of the most important aspect of WSN research is location estimation. As a good solution of fine grained localization Reichenbach et al. introduced the Distributed Least Squares (DLS) algorithm, which splits the costly localization process in a complex *precalculation* and a simple *postcalculation* which is performed on constrained sensor nodes to finalize the localization by adding locale knowledge. This allows to perform an originally complex calculation with high precision on constrained nodes. Besides this advantage, DLS lacks in two harmful constraints concerning practical appliance. On the one hand the algorithm does not scale, i.e. calculation and communication increases with the number of beacon nodes or with network size, respectively. On the other hand DLS even does not work for large networks. An important assumption of DLS is that each blind node can communicate with each beacon node to receive the precalculation and to determine distances to beacon nodes. In this work we present an adaptation of DLS, concerning major changes, which enables DLS to be used in large WSNs for the first time. At the same time computational and communicational cost of each node becomes independent from network size, while precision is kept on the same high level.**

***Keywords*-wireless sensor networks; localization; scalability**

I. Introduction

Recent technological advances have led to the development of tiny wireless devices, which are able to sense their environment, compute simple tasks and exchange data among each other. Interconnected assemblies of such devices, called Wireless Sensor Networks (WSNs), are commonly used to observe large inaccessible areas. In many applications of WSNs, knowledge of nodes' locations is mandatory for a meaningful interpretation of the data sensed. Location-awareness is not only necessary to assign a location to measured values but also to perform geographic routing [1][2] or location based clustering as described in [3]. Due to existing limitations in terms of size and energy consumption, local positioning within the network is preferred over utilizing common positioning systems like GPS. Therefore, the presence of location-aware sensor nodes, referred to as beacon nodes, is typically assumed. These nodes know their own position a priori or via common positioning systems. The remaining nodes, which we refer to as blind nodes, are assumed to use communication and any kind of distance estimation or neighborhood information to estimate their position with the help of beacon nodes.

Existing localization techniques can be divided into coarse-grained and fine-grained localization. Commonly this classification reflects the trade off between precision and resource consumption of the corresponding techniques. Coarse-grained approaches like Centroid Localization (CL) [4] and Adaptive Weighted Centroid Localization (AWCL) [5] often abstain from exact distances, require less communication and computation and provide lower precision estimates. On the other hand, fine-grained approaches strive to an exact localization with high precision, which is achieved through costly computations and estimation of distances or angles. Achievable precision of such approaches, commonly based on a set of linear equations, hardly depends on distance or angle estimation, respectively. In [6] Reichenbach et al. firstly described Distributed Least Squares (DLS) as a localization approach which combines high precision with relatively low complexity. DLS splits the costly localization calculation into two parts. The complex part, called precalculation, is performed on a high performance sink, independently from a specific blind node. The remaining part of calculations is less complex and performed on resource-constrained blind nodes.

The major drawback of DLS is that it inherently assumes communication of all blind nodes with all beacon nodes in the network. On the one hand, this does not consider break down of beacon nodes. Furthermore it presumes that each blind node in the network is able to communicate directly with the sink and is able to estimate its distance towards each beacon node. This makes the DLS infeasible for use in large multi-hop networks which represents one of the most interesting scenarios for WSNs. On the other hand, communication and computational effort on each blind node increases with the number of beacon nodes and, therefore, with the applied network size.

To overcome these drawbacks, we developed a new localization algorithm which bases on the idea of DLS. Some major changes enables our new algorithm to be used in large WSNs. The distinct features of our algorithm comprise that the computational cost becomes independent from network size and communication effort also scales better while constrained blind nodes are significantly relieved.

The remainder of the paper is organized as follows. Section II covers related work, especially the original DLS algorithm and existing derivatives. It also covers the drawbacks and emphasizes the resultant need for a new algorithm. In Section III, we present our new localization algorithm and emphasize the enhancements of scalable DLS (sDLS) compared to DLS. Section IV describes the simulation environment which was used to evaluate the new algorithm. The simulation results are presented in Section V. Section VI summarizes the presented work and covers topics of future work.

II. Related Work

The DLS algorithm was developed to alleviate trade off between precision and cost of localization and provides localization with high precision and low cost [6]. The original approach can be divided into two parts. We start with considering the arithmetical part and continue with the algorithmic part thereafter.

A. Math behind DLS

The system of equations which have to be solved for localization of a blind node is originally build by distance equations as given in equation (1).

$$(x - x_i)^2 + (y - y_i)^2 = r_i^2 \quad (i \in \{1, 2, \ldots, m)\}) \tag{1}$$

Here x and y denotes the unknown position of a blind node. The known position of a beacon node is denoted as x_i and y_i, while the distance between both nodes is denoted as r_i. The number of beacon nodes, used for localization is given as m.

To linearize this system of equations a linearization tool [7] is used. Therefore an arbitrary beacon node is selected as linearizer and utilized as given in equation (2). This reduces the number of equations by 1.

$$\begin{gathered}(x - x_L + x_L - x_i)^2 + (y - y_L + y_L - y_i)^2 = r_i^2 \\ (L \in \{1, 2, \ldots, m\}, i \in \{1, 2, \ldots, m\} \setminus L)\end{gathered} \tag{2}$$

Restructuring the equations leads to equation (3), where r_L denotes the distance between blind node and linearizer, r_i is the distance between blind node and beacon node and d_{iL} denotes the distance between linearizer and beacon node.

$$\begin{aligned}&(x - x_L)(x_i - x_L) + \\ &\quad (y - y_L)(y_i - y_L) = \frac{1}{2}\left[r_L^2 - r_i^2 + d_{iL}^2\right] \\ &\quad = b_{iL}\end{aligned} \tag{3}$$

The restructured system of equations can be written in matrix form as

$$\mathbf{Ax} = \mathbf{b} \tag{4}$$

with

$$\begin{gathered}\mathbf{A} = \begin{pmatrix} x_2 - x_L & y_2 - y_L \\ x_3 - x_L & y_3 - y_L \\ \vdots & \vdots \\ x_m - x_L & y_m - y_L \end{pmatrix}, \\ \mathbf{x} = \begin{pmatrix} x - x_L \\ y - y_L \end{pmatrix}, \mathbf{b} = \begin{pmatrix} b_{2L} \\ b_{3L} \\ \vdots \\ b_{mL} \end{pmatrix}\end{gathered} \tag{5}$$

As it is already implied in equation (5), DLS uses the first beacon as linearization tool, i.e. $L = 1, i = 2, \ldots, m$.

At this point matrix $\mathbf{A}$ only consists of beacon position data, while $\mathbf{b}$ contains distances between beacon nodes and blind nodes. Therefore calculations on $\mathbf{A}$ can be performed once for all blind nodes at a sink outside the WSN. The localization will be finalized on each blind node by performing the remaining part of the calculation. To solve the linear equations system is was proposed to use normal equations, qr-factorization or singular-value decomposition. Using normal equations, which is favored by the authors, equation (4) is restructured as given in equation (6). In this case $\mathbf{A}_p = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$ and $\mathbf{d}_p = \mathbf{d}^2$ present the precalculation, performed on the sink.

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\frac{1}{2}\left[r_L^2 - \mathbf{r}^2 + \mathbf{d}^2\right] \tag{6}$$

B. The DLS Algorithm

The algorithm is based on the precondition that each beacon node and each blind node is able to communicate with the sink. Furthermore each blind node have to be able to determine its distance to all beacon nodes, which requires direct communication to all beacon nodes.

The algorithm consists of four steps:

Step 1 - *Initialization Phase:*
All beacons send their position to the sink.
Step 2 - *Precalculation Phase:*
Sink computes $\mathbf{A}_p$ and $\mathbf{d}_p$.
Step 3 - *Communication Phase:*
Sink sends precalculated data to all blind nodes.
Step 4 - *Postcalculation Phase:*
Blind nodes determine distance to every beacon node, receive precalculated data and estimate their location by solving the postcalculation.

C. Limitations of DLS

The described DLS algorithm, as well as its derivates iterative DLS (iDLS) [8] and mobile DLS (mDLS) [9], demands for the precondition that each beacon node has to be able to communicate with the sink and each blind node has to be able to directly communicate with each beacon node for distance estimation. That means that the described algorithm is limited to one-hop networks. The problem of communication might be solved using multi-hop strategies. However, this strategy does not solve the problem of distance estimation.

Another restriction is the unique precalculation, which is the same for each blind node. However, the precalculated matrix has to be updated whenever beacon nodes, included in precalculation, are not within the communication range. Recognizing that one beacon node is used to linearize the system of equations and, therefore, contributes to each row of the matrix, it is clear that leaving out this beacon node causes significant computations. Specifically, the whole matrix has to be updated whenever the beacon node used for linearization is outside the communication range. Even mDLS, which actually updates precalculation, lacks in this point and works only near the linearizing beacon node.

Besides the mentioned drawbacks, the original algorithm lacks scalability. Due to the unique precalculation, all beacon nodes contribute to the precalculated matrix, which leads to an increasing complexity in communication an computation on each blind node with increasing network size.

III. Scalable DLS

To overcome the restrictions of DLS as mentioned in the previous section, the following requirements need to be met:

1) enable blind nodes to determine their distance to all beacon nodes, needed for their localization
2) make the linearizing beacon node reachable for all blind nodes
3) reduce number of data, transferred to a blind node
4) reduce computation per blind node to a necessary minimum

The basic idea is to provide each blind node with an individual precalculation that includes exactly those beacon nodes which are in its communication range. Since this will not be achievable in most cases, it is a good solution to provide a precalculation that includes more or less beacon nodes than accessible by the blind node. The blind node then needs to delete parts of the precalculation, i.e. beacon nodes outside its communication range, or insert additional data, i.e. beacon nodes insight its communication range. Our approach to reduce the number of data updates is to provide each beacon node with a precalculation consisting only of beacon nodes in its own communication range. A blind node is expected to use the precalculation of the closest beacon node. As illustrated in figure 1 the amount of beacons that have to be added by the blind node as well as those that have to be deleted from precalculation is relatively small. To reduce the update cost, a blind node first performs deletions and then adds new data.

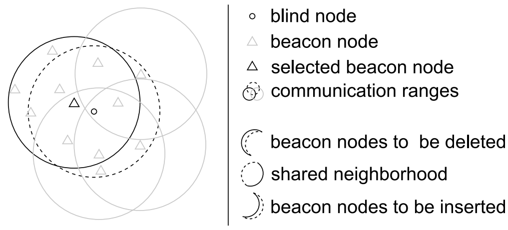

Figure 1. Blind node selecting precalculation of nearest beacon node to minimize update operations

Since the blind node has to update the precalculated data set, sDLS uses qr-decomposition instead of normal equations to solve the linear equations system. This allows updating and downdating [10]. Doing so, with $\mathbf{A} = \mathbf{QR}$ and $\mathbf{R}$ upper triangular, the system of equations, given in equation (4), becomes restructured as given in equation (7). The precalculation is now presented by $\mathbf{Q}^T$, $\mathbf{R}$ and $\mathbf{d}_p = \mathbf{d}^2$, individually determined for each beacon node.

$$\mathbf{Rx} = \mathbf{Q}^T \frac{1}{2} \left[r_L^2 - \mathbf{r}^2 + \mathbf{d}^2 \right] \tag{7}$$

DLS is not able to perform a similar step, since the beacon node used for linearization is part of the precalculation and affects each row of the matrix. sDLS choses the linearizing beacon node individually for each blind node. The precalculation sent to a beacon node uses the beacon node itself as linearizer, so at least the linearizer of the chosen precalculation is in a blind nodes' communication range.

Following the above described strategy the new sDLS algorithm consists of the following steps:

Step 1 - *Discovery Phase:*
All beacons send a local broadcast to discover neighboring beacon nodes.

Step 2 - *Initialization Phase:*
All beacons send their position and a list of their neighbors to the sink.

Step 3 - *Precalculation Phase:*
Sink computes $\mathbf{Q}^T$, $\mathbf{R}$ *and* $\mathbf{d}_p$ *individually for each beacon node.*

Step 4 - *Distribution Phase:*
Sink sends precalculated data to beacon nodes.

Step 5 - *Communication Phase:*
Beacon nodes send precalculated data to blind nodes.

Step 6 - *Postcalculation Phase:*
Blind nodes determine distance to accessible beacon nodes, receive precalculated data, update precalculation and estimate their own position by solving the postcalculation.

While steps 1 to 4 are expected to be performed once, steps 5 and 6 are expected to be repeated. Beside the precalculated data a list of beacon nodes is transmitted as well in sDLS as in DLS to indicate which line of the precalculation corresponds to which beacon node.

IV. SIMULATIONS

To verify performance of the described algorithm, the MatLab based network simulator Rmase is used [11]. The simulator provides a realistic radio communication model, given in equation (8), including spatial (α) and temporal (β) normal distributed fading, random transmission errors, collisions and a CSMA-CA MAC layer. The Rmase layer structure provides layers, implementing addressing, queuing, aggregation and routing. The original Rmase also includes an application simulation as a core component, which generates messages on each node.

$$P_{rec} = P_{send} * \frac{1}{1 + d_{rs}} * (1 + \alpha(x, y)) * (1 + \beta(t)) \quad (8)$$

For our simulations Rmase has been partly modified. The Rmase application layer received the most significant change. It represents an extra layer which can hold real applications now. Furthermore, we implemented a static bidirectional spanning-tree routing for an objective comparison of data traffic. Since routing performance is not in the focus of this work, an additional layer which turns collided packets into received packets has been implemented to avoid overhead.

A random deployment of n^2 nodes within a field of $n * n$ arbitrary distance units (adus) was used. The first node is always used as sink, while the remaining nodes are randomly chosen as blind nodes (50%) or beacon nodes (50%). The field size parameter n was varied from 5 to 30. The average communication range, given by the radio model was 3 adus. For each field size the average over 100 simulations has been determined. In each simulated network our new sDLS algorithm as well as the original DLS algorithm has been performed concurrently. In contrast to real applications the simulations includes only one localization per blind node after the initial precalculation and data distribution.

Existing restrictions of DLS necessitate the use of all beacon nodes in the network and therefore a distance estimation to all beacon nodes. Therefore, the simulator's radio model, given in equation (8), is used to provide a signal strength for distance estimation even though the calculated strength undergoes the reception threshold.

V. RESULTS

The new sDLS algorithm is compared to the original DLS in terms of localization accuracy, computation cost on blind nodes and cost of data transmission. Additionally the number of beacon nodes used by a blind node as well as the update-rate of sDLS is analyzed.

A. General Results

Figure 2 shows the number of beacon nodes used per blind node and the percentage of blind nodes which have been able to estimate their position using DLS and sDLS, respectively. It is shown that sDLS achieves a success rate of nearly 100% using about 12.4 beacon nodes. Only few blind nodes fail to self-localize which is caused by the deployment of some blind nodes in regions not covered by sufficient numbers of beacon nodes. In contrast, DLS forces blind nodes to use a raising number of beacon nodes, i.e. all beacon nodes in the field, which causes most blind nodes to fail even in relative small networks. In other words, sDLS works excellent in large networks while DLS completely fails.

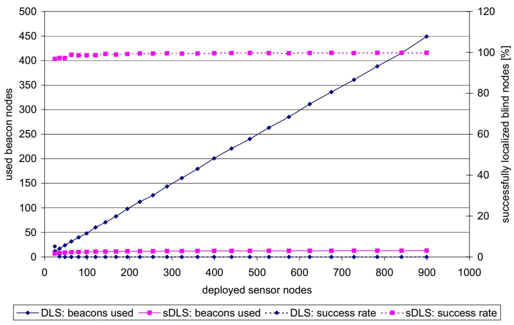

Figure 2. Number of used beacon nodes (left) and success rate (right) over total number of deployed nodes

B. Localization

To compare the performance of localization, each blind node estimates its position with both algorithms. To perform DLS localization which actually fails in most cases due to a missing direct link for distance estimation, a corresponding signal strength was emulated as described in Section IV. After successful localization, the distance between real position and estimated position is regarded as localization error. The illustration in figure 3 shows that the averaged localization error of sDLS outperforms that of DLS even though less beacon nodes are taken into account using sDLS. As we found out the outliers are not systematic but result from heavy outliers of single nodes in single simulations. The high localization error of DLS may be partly caused by the emulated signal strength. Alternatively, in real networks a hop-based distance estimation can be used which comes with an increase of complexity and data transmission.

C. Data Transmissions

In contrast to DLS which generates only one large set of precalculated data, sDLS produces a small individual data set for each beacon node. The results, given in figure 4, show the mean amount of data transmission per node. Blind nodes, that did not participated in data transmission, i.e. did not send any data, have not been taken into account. Each transmitted float value, e.g. position or precalculation, have been rated as four bytes, while a transmitted integer have been rated as two bytes. The figure shows, that for small sensor networks sDLS uses more data transmission than

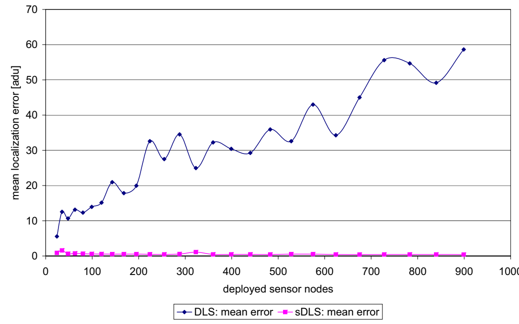

Figure 3. Mean error of localization over total number of deployed nodes

DLS. But in contrast to DLS the slope of sDLS decreased with the number of nodes and the data transmission per node tends to become constant for large sensor networks, while those of DLS increases with the number of nodes. Additionally, it has to be considered, that the simulations only cover the initialization phase and one round of localization. It is assumed that in real WSN applications blind nodes will estimate their position in recurring intervals. Due to the small number of beacon nodes used for sDLS, data transmission in localization phase is much smaller for sDLS than for DLS. Therefore, sDLS outperforms DLS in terms of data transmission. Furthermore , in contrast to DLS, sDLS would greatly benefit from multiple sinks, reducing routing activity, because of the small individual data packets.

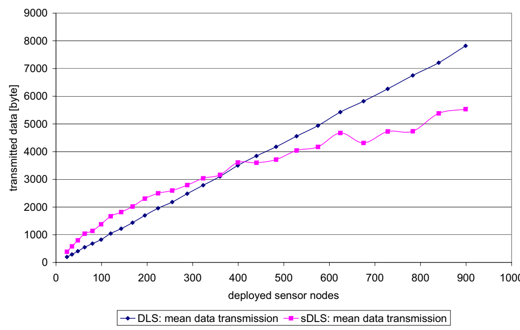

Figure 4. Mean data transmitted per node

D. Cost of Computation

To quantify cost of computation, the number of operations has been counted on each blind node. Due to the different complexity of operations three kinds of operations have been analyzed. Additions and subtractions has been summed up as additions; multiplications and divisions are combined in multiplications, and powers include squares and square roots. While computation in DLS only consists of the final determination of the blinds' positions, sDLS previously performs an additional data update. In figure 5 the overall computations of both algorithms are illustrated.

It can be seen, that for large sensor networks, the number of additions used by sDLS undergoes the number of additions used by DLS. Also the use of powers and square roots is much smaller for sDLS than for DLS. While the number of additions and powers is only slightly affected by sDLS data update, it has much stronger impact on the number of multiplications. Although, it remains nearly constant for large networks and will, therefore, stay smaller than that of DLS for very large networks, it should be reduced in future work to optimize the data update.

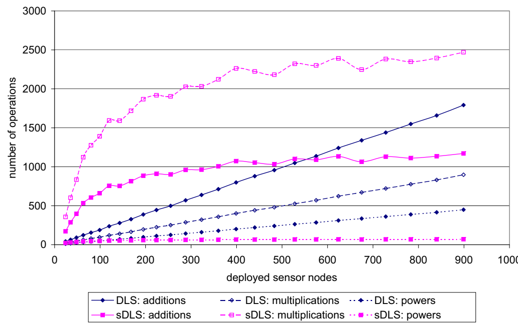

Figure 5. Mean number of operations performed on a blind node

In contrast to figure 5, only the final calculation part of sDLS, i.e. except the data update, as well as DLS are illustrated in figure 6. It is shown that, regarding only this part of calculation, sDLS notably outperforms DLS in all aspects of computation even for small networks. It is, therefore, strongly expected that cost of computation of sDLS can be improved by a modified update behavior.

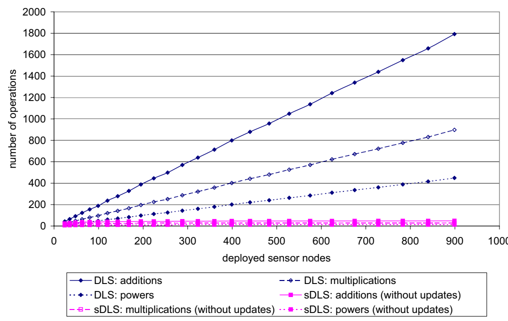

Figure 6. Mean number of operations performed for final determination on a blind node

E. sDLS Update Performance

The new sDLS algorithm uses matrix updates to adapt the precalculated data set to the beacon nodes in range of a blind node. The aim of the algorithm was to keep the number of updates down by choosing the precalculated data set of the nearest beacon node. The graph in figure 7 shows how many beacon nodes have been used by a blind node, i.e.

have been in range of a blind node, how many beacon nodes have to have been deleted from the precalculation and how many beacon nodes have been added to the precalculation. The results show, that for large networks, about a third of the beacon nodes have been swapped. Although the number of updates is small, there is also a potential to improve the algorithm by reducing the updates in future work. While the number of updates has the strongest impact on the computational cost, the whole algorithm can profit from an improved update performance.

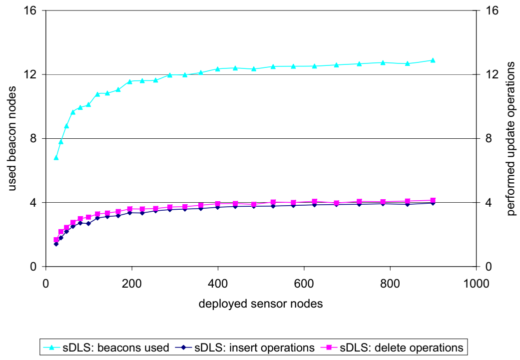

Figure 7. Mean number of update operations performed by a blind node

F. General Remarks

Regarding before mentioned results, it can be observed that the number of beacon nodes, used with sDLS raises up to a certain value, after which it slightly drops again. A similar effect can be observed regarding the update performance as well as the computational complexity. It is assumed that this effect is an impact of the border area which becomes less influencing for large networks.

VI. Conclusion

The presented sDLS algorithm made the idea of DLS applicable for large WSNs. Additionally it outperforms the original DLS in terms of localization and cost of communication and computation. In contrast to DLS all costs become constant for large WSNs while those of DLS increase with network size. The used data update, which strongly influences the cost of computation, provides prospects of future improvements.

There are several opportunities to further improve sDLS in future work. On the one hand, cost of computation can be improved by a modified update behavior or a modification of the precalculated data sets provided to the blind nodes. Also a well-directed selection of precalculated data by the blind node may lead to an improvement in cases when blind nodes fails while selecting the nearest beacon node. On the other hand data transmission can be significantly reduced if beacon nodes would share precalculated data or parts of it. Using a cluster based structure like 2-MASCLE [3] may be one suitable solution.

Acknowledgment

This work was supported by the German Research Foundation under grant number BI467/17-2 (keyword: Geosens2)

References

[1] K. Akkaya and M. F. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[2] J. Al-Karaki and A. Kamal, "Routing techniques in wireless sensor networks: a survey," *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 6–28, Dec. 2004.

[3] J. Salzmann, R. Behnke, D. Lieckfeldt, and D. Timmermann, "2-MASCLE - A Coverage Aware Clustering Algorithm with Self Healing Abilities," *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pp. 739–744, Dec. 2007.

[4] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, Oct. 2000.

[5] R. Behnke and D. Timmermann, "AWCL: Adaptive Weighted Centroid Localization as an efficient Improvement of Coarse Grained Localization," *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, pp. 243–250, March 2008.

[6] F. Reichenbach, A. Born, D. Timmermann, and R. Bill, "A distributed linear least squares method for precise localization with low complexity in wireless sensor networks," *Distributed Computing in Sensor Systems*, pp. 514–528, 2006.

[7] W. S. Murphy and W. Hereman, "Determination of a position in three dimensions using trilateration and approximate distances," Tech. Rep., 1999.

[8] F. Reichenbach and D. Timmermann, "On improving the precision of localization with minimum resource allocation," in *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, Aug. 2007, pp. 1093–1098.

[9] F. Reichenbach, D. Lieckfeldt, and D. Timmermann, "Using qr-updating with reduced complexity for precise localization in mobile sensor networks," in *Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual*, May 2008, pp. 432–439.

[10] D. S. Watkins, *Fundamentals of matrix computations*, 2nd ed., ser. Pure and Applied Mathematics. New York: Wiley-Interscience [John Wiley & Sons], 2002.

[11] Y. Zhang, M. Fromherz, and L. Kuhn, "Rmase: Routing modeling application simulation environment," 2009, http://www2.parc.com/isl/groups/era/nest/Rmase/.