Cooperative Method for Wireless Sensor Network Localization

Angel Stanoev[a], Sonja Filiposka[b], Visarath In[c], Ljupco Kocarev[a,b,d,*]

[a] *Macedonian Academy for Sciences and Arts, Skopje, Macedonia*
[b] *Faculty of Computer Science and Engineering, Ss Cyril and Methodius University, Skopje, Macedonia*
[c] *Space and Naval Warfare Systems Center Pacific*
Code 71730, 53560 Hull Street, San Diego, CA 92152-5001, USA
[d] *BioCircuits Institute, University of California, San Diego*
9500 Gilman Drive, La Jolla, CA 92093-0402

Abstract

In order to obtain an efficient wireless sensor network localization, several enhancements based on the decentralized approach are proposed. These features can be used in the cases when multiple distance measurements are used as input, where each node iteratively updates its estimated position using a maximum likelihood estimation method based on the previously estimated positions of its neighbors. Three novel features are introduced. First, a *backbone* is constructed, that is, a subset of nodes that are intermediaries between multiple beacon nodes, which guides the localization process of the other (non-backbone) nodes. Second, the space is perturbed more often during the earlier time steps to avoid reaching poor local minima in some cases regarding the localization optimization function. Third, for better localization of the non-backbone (or peripheral) nodes and avoidance of the rigidity problem, 2-hop neighboring distances are approximated. The introduced features are incorporated in a range-based algorithm that is fully distributed, shows good performance, and is scalable to arbitrary network size.

1. Introduction

Wireless sensor networks (WSNs) have recently attracted considerable research interest by providing unprecedented opportunities for monitoring and controlling homes, cities, and the environment. They consist of spatially distributed smart autonomous sensors, networked through wireless links and deployed in large numbers. Self-localization capability is a highly desirable characteristic of wireless sensor networks. Sensor network localization algorithms estimate the locations of sensors with initially unknown location information by using knowledge of the positions of a few sensors and inter-sensor measurements such as distance and bearing measurements. Sensors with known location information are called beacons or anchors and their locations can be obtained by using a global positioning system (GPS), or by installing them at points with known coordinates.

Many approaches for WSN localization have been studied in the literature. Overviews of WSN localization techniques are presented in [1, 2, 3, 4]. For a survey of localization in mobile WSNs, we refer to [5]. The spatio-temporal cooperation for localization and navigation is extensively studied in [6, 7].

1.1. Previous work

Localization techniques can be broadly classified into two categories: range-based and range-free. In large-scale WSNs, where signal range is limited, range-based schemes typically require a lot of beacon nodes to produce accurate results. On the other hand, range-free schemes estimate inter-node distances based on hop count information, thus all target nodes can be localized with fewer beacon nodes. Range-free techniques are those where node position estimation is not based on the distance estimation between nodes but on the solution to heuristic or optimization problems with a decentralized characteristic. The typical range-free algorithms include Centroid [8], CPE (Convex Position Estimation) [9], and DV-hop (Distance Vector-hop) [10]. Centroid and CPE are simple, having low complexity, but they require a normal node to have at least three neighboring beacons. DV-hop algorithm can handle the case where a normal node has less than three neighbor beacons.

Since the quality of localization is easily affected by node density and network conditions, range-free approaches typically provide imprecise estimation of node locations. Range-based approaches measure the Euclidean distances among the nodes with certain ranging techniques and locate the nodes using geometric methods, such as time of arrival (TOA), time difference of arrival (TDOA), and angle of arrival (AOA).

Here we focus on range-based designs for sparse networks. Several range-based algorithms that address the

*Corresponding author
Email addresses: astanoev@cs.manu.edu.mk (Angel Stanoev), sonja.filiposka@finki.ukim.mk (Sonja Filiposka), visarath.in@mail.sdsu.edu (Visarath In), lkocarev@ucsd.edu (Ljupco Kocarev)

network or beacon sparseness problem will be first reviewed. Savarese et al. [11] proposed a virtual-coordinate-based algorithm TERRAIN to address the sparse beacon problem. The algorithm constructs a virtual coordinate system on each beacon and takes the advantage of the property that the virtual coordinate holds the distance information between each node pair.

Other researchers utilized local maps to localize non-beacon nodes [12], [13]. They first use distance measurements between neighboring nodes to construct local maps, and then stitch them together to form a global map. Savvides et al. [14] used collaborative multilateration among neighbors to compensate the ranging information shortage, which localizes nodes by forming an over determined system of equations with a unique solution set. Collaborative multilateration performs better than trilateration in sparse networks. The disadvantage is that the collaboration is restricted in neighbors, so that the performance gain is limited.

Goldenberg et al [15] introduced the concept of finite localization, which holds all candidate positions of each node and prunes incompatible ones when other nodes join the procedure. Given proper beacon distribution, the algorithm is able to locate nodes in a globally rigid region, but may fail to localize the regions that contain few beacons. Wang et al [16] introduced the concept of component, by which nodes are grouped into components which are able to better share ranging and beacon knowledge. Operating on the granularity of components, the proposed design relaxes two essential restrictions in localization: the node ordering and the beacon distribution. Zhao et al [17] proposed a combined and differentiated localization approach for localization that exploits the strength of range-free approaches and range-based approaches using received signal strength indicator (RSSI). To achieve a better ranging quality, the proposed algorithm incorporates virtual-hop localization, local filtration, and ranging-quality aware calibration.

The cooperation between the network nodes, i.e., the exchange of messages, can be employed for different purposes. Depending on which nodes are used as reference points for each individual location estimation, the cooperative localization methods can be classified in two main classes. Multi-hop localization methods, such as the aforementioned ones, use the beacon nodes as individual reference points. There the cooperation takes place primarily to estimate the distances between the non-beacon nodes and the beacons, often located multiple hops away. Later, these distances are used for calculating individual position estimates by methods such as multilateration for example. On the other hand, methods such as [18, 6] and the one described in this paper perform iterative update of the position estimate of every node by using the position estimates of its neighbors as reference points.

Sensor network localization is a nonconvex optimization problem that, however, can be converted to a Semi-Definite Programming (SDP) problem by transforming the quadratic embedding constraints to a matrix inequality, which can be finally rewritten as a standard SDP problem. This approach has recently attracted considerable attention; for example, a technique for localization of sensors based on considering together local structures that are fit together in an as-rigid-as-possible manner is proposed recently in [19]. The local structures consist of reference patches (set of four sensors) and reference triangles, both obtained from inter-sensor distances.

In [20] the authors focused on the problem of sensor network localization in the plane by considering the group of algorithms that integrate local distance information (patches) into a global structure determination. Each patch is separately localized in a coordinate system of its own using either the stress minimization approach or by SDP. To every patch there corresponds an element of the Euclidean group Euc(2) of rigid transformations in the plane, and the goal is to estimate the group elements that will properly align all the patches in a globally consistent way. In [21], Shamsi and co-workers analyzed and determined sufficient conditions and formulations that guarantee that the SDP relaxation is exact, i.e., give the correct solution of the sensor localization problem. These conditions can be useful for designing sensor networks and managing connectivities in practice.

1.2. Our contribution

In this paper, we focus on introducing new features that will improve the performances of decentralized algorithms for localization that do not use centralized units for computation and data gathering. The localization process, however, is based on known global positions of some of the network nodes. These are necessary for identifying the absolute positions of the other nodes. Iterative position estimation update is executed by each node using the approximated distances from its neighbors. Hence, the marginal distribution of each node is optimized since the nodes' locations are the component variables comprising the joint distribution that describes the state space. However, optimizing the marginal distributions, as convenient as it is, does not always optimize the joint distribution. The aim of this paper is to present means by which one can drive the optimization of the marginals in a direction that will produce a better sample value of the joint distribution, i.e., a set of individual locations that has high joint probability. We apply these improvement features to an example algorithm, and illustrate how they enhance its efficiency. They can also be applied on other algorithms that optimize the marginals, such as [6] that uses graphical models for example.

Comparing to other algorithms for localization, the algorithm enhancements presented here introduce three novel features. First, we take into account the fact that some nodes are one to few hops away from mutliple beacons, while others are connected to the beacons via longer multi-hop routes (Fig. 1). More precisely, by taking advantage of the nodes that interconnect beacon nodes, we

propose the construction of a backbone, that is, a subset of nodes that are intermediaries between multiple beacon nodes, which can then be used to guide the localization process of other non-backbone nodes. Second, marginals explore the state space more excitedly during the earlier iterations in order to target better joint distribution sample value in the steady state, regarding the localization optimization. Third, for better localization of the non-beacon nodes and avoidance of the rigidity problem, the 2-hop neighboring distance approximation is introduced. By using only local communication, the enhancements based algorithm proves to be naturally fast (discussed further in Section 3).

The outline of the remainder of the paper is as follows: section 2 presents the properties of the proposed features and explains the steps of an example algorithm that incorporates them, section 3 presents the results of the example algorithm simulated on the computer generated networks and its performance measures, and section 4 provides the conclusion.

2. Improved features for localization: an example

Three improvement features are the main contribution of this work and, therefore, we construct, as an example, an algorithm on which these features have been applied. It is range-based decentralized iterative algorithm that uses known locations of the beacons. In this section we describe the characteristics of the algorithm and the basic input that it requires. Motivated by the typical settings where the algorithm is applied, the contributions of the corresponding novel features are introduced.

2.1. Range measurement

Range-based algorithms utilize locally measured neighboring distances for better estimation. There are various methods for measuring distances between sensor nodes: received signal strength indicator (RSSI), time of arrival (ToA), time difference of arrival (TDoA), round-trip time of arrival (RToA), ultra-wide bandwidth (UWB) [2, 6], among others. However, most of these methods are prone to errors due to various reasons, like obstacles or the environment anisotropy itself. A single distance measurement (point estimate) is potentially highly inaccurate and can influence the localization results heavily. Hence, assuming noisy distance measurements, probability estimation of a distance is obtained by taking multiple point estimates. The mean and variance of the set of these point estimates between each neighboring pair are recorded during the initialization step. 20 point estimations are used to approximate the normal distribution of the mean. This information is used as an input to the algorithm subsequently.

2.2. Beacon nodes

For absolute localization the locations of the beacon nodes are needed as an input. They can be predefined or obtained by GPS units. As it might be expensive to attach a GPS unit to every node in a network of hundreds or thousands of nodes, only a small percentage of the nodes is assumed to be equipped with such capability. It is also assumed that there are at least three non-collinear beacons in the observed network, as that is the lowest number of beacons required for unambiguous localization in a two-dimensional space. There has not been made any other assumption about the distribution of their locations, because they are typically unknown or uncontrollable during the deployment process.

2.3. Graph backbone

Consider a wireless sensor network described with a graph $G = (V, E)$, where for convenience we write $V = \{1, 2, \ldots, n\}$; we assume that the first l nodes $1, \ldots, l$ are beacons. We further assume that the graph is simple (unweighted and undirected); let $A = [a_{ij}]$ be its corresponding adjacency matrix. Edge between two nodes in the network exists if their distance is lower than a predefined range R_{max}.

From the graph G we construct directed graph $G_1 = (V, E_1)$ as follows. Assume that i and j are neighbors (that is, $a_{ij} = 1$), then: if both i and j are non beacon nodes, both arcs ij and ji exist; if one node is a beacon node and the other one is not, only arc from the beacon node to non beacon node exists; and finally, if both i and j are beacons, no arcs between these two nodes exist. In other words, the undirected links of the beacons are replaced by outgoing links in the graph G_1, so that beacon state is not influenced by the states of other nodes. The resulting graph G_1 is directed graph containing at least $l+1$ strongly connected components: l leader nodes (the beacons) with no in-links and other components containing the rest $n-l$ nodes. Let $A_1 = [a_{ij}^{(1)}]$ be the adjacency matrix of the graph G_1.

As a preprocessing step, we run a linear process based on the leader nodes [22]. Suppose that the l beacon nodes are leaders. We associate a membership vector $\mathbf{m}_i$ with each node i: a probabilistic vector of size l that states to what degree the node is related to each of the leader nodes. Hence $\sum_j m_{i,j} = 1$. During the execution of the process, each beacon node $i \leq l$ has constant membership vector $\mathbf{m}_i = \mathbf{e}_i$, where $\mathbf{e}_i$ is the unit vector with its i-th component equal to 1. At each time step, every non beacon node $i > l$ updates its membership vector according to the update rule

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \epsilon \sum_{j=1}^{n} a_{ij}^{(1)} (\mathbf{m}_j(t) - \mathbf{m}_i(t)) \quad (1)$$

where $\epsilon \in (0, 1/\Delta)$ is the step size, and Δ is the maximum degree in the network. Eq. (1) holds for the beacon nodes

as well. In matrix form we can write

$$M(t+1) = (I - \epsilon L_1)M(t) \tag{2}$$

where the membership vectors $\mathbf{m}_i^T(t)$ are rows of the row-stochastic membership $n \times l$ matrix $M(t)$ and L_1 is the Laplacian matrix of the graph G_1. In an explicit form the matrix $M(t)$ can be written as:

$$M(t) = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & & & \\ 0 & 0 & \dots & 1 \\ m_{l+1,1}(t) & m_{l+1,2}(t) & \dots & m_{l+1,l}(t) \\ \vdots & & & \\ m_{n1}(t) & m_{n2}(t) & \dots & m_{nl}(t) \end{bmatrix}.$$

The process (2) converges exponentially fast, as it represents linear difference equation [23]. The equilibrium membership matrix is

$$M^* = \begin{bmatrix} \mathbf{v}_1 \dots \mathbf{v}_l \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & & & \\ 0 & 0 & \dots & 1 \\ m^*_{l+1,1} & m^*_{l+1,2} & \dots & m^*_{l+1,l} \\ \vdots & & & \\ m^*_{n1} & m^*_{n2} & \dots & m^*_{nl} \end{bmatrix}$$

where each $\mathbf{v}_i$, $i = \overline{1,l}$, is an eigenvector of the Laplacian L_1 corresponding to the zero eigenvalue and $\sum_j m^*_{i,j} = 1$.

The final membership vectors of the non-beacon nodes $\mathbf{m}_i^*$ do not depend on their initial values $\mathbf{m}_i(0)$, but only on the eigenvectors of the Laplacian, hence on the network/graph topology, and on the initial values of the leaders, which are trivially equal to $\mathbf{m}_i(0) = \mathbf{e}_i$ for all $i = 1, \dots, l$. As shown in [22], for a given non-beacon node i, the values of m^*_{ij} in the row i of the matrix M^* correspond to 'influence' the beacons $j = 1, \dots, l$ have on the node i. Therefore, all non-beacon nodes can be grouped/arranged in l communities by using the following rule: the node i belongs to community k if $m_{ik} > m_{ij}$ for all $j = 1, \dots, l$. However, since in general $m^*_{ij} \neq 0$, the membership vector describes the node's involvement in each community.

The membership vectors determine the communities that are formed around the beacons driven by their influences [22]. As a result, most of a beacon's community nodes will have similar membership vectors, with its respective component as the dominant one. Therefore, the (Euclidian) distance between the membership vectors of two neighboring nodes provides an additional information regarding the structure of the network. For example, if the distance is large it indicates that those two nodes belong to two different communities and their shared edge intermediates in between. When considering only the components of the respective beacons, this distance would be the largest on average for edges that are placed on the shortest path between the beacons.

We consider a new undirected weighted graph $G_2 = (V, E, w)$, whose nodes and edges are the same as in G, and the weight function $w : E \to R$ is defined as $w_{ij} = \|\mathbf{m}_i^* - \mathbf{m}_j^*\|$, where $\|\cdot\|$ is the Euclidian distance. Let N_i be the set of all neighbors of the node i. The node strength (weighted degree) defined as

$$s_i = \sum_{j \in N_i} \|\mathbf{m}_i^* - \mathbf{m}_j^*\|,$$

serves as a (simple) node centrality metric of G_2. A non-beacon node with high strength has large weight of at least one of its neighbors' links and therefore the node intermediates between at least two communities (or two beacons). Finally, we define *graph backbone* as a set of non-beacon nodes for which the node strength is greater than the average, that is

$$\left\{ k : s_k > \frac{1}{n-l} \sum_{i=l+1}^{n} s_i, k = l+1, \dots n \right\}.$$

Figure 1 illustrates the main concepts introduced in this section. As an example we consider a network with 3 beacons (represented by squares) and 47 non beacon nodes. Colors of the non beacon nodes are mixtures of the beacon nodes' colors in correspondence to their membership vectors. Thus, the color of a non beacon node i is a mixture of 3 beacon nodes' colors in a ratio $m_{i1} : m_{i2} : m_{i3}$ such that $\sum_{j=1}^{3} m_{ij} = 1$. The largest value of m_{ij} determines the dominant color but also the community to which node i belongs. Link widths are proportional to $\|\mathbf{m}_i^* - \mathbf{m}_j^*\|$. Node size reflects its strength. The graph backbone has 16 nodes; these nodes are represented with diamonds.

2.4. Probabilistic iterative approach

As mentioned earlier, the proposed algorithm is iterative, i.e., each node iteratively updates its own location estimation using the neighbors' location estimates from the previous iteration. As the set of location estimates from every node represents a variable of the joint distribution, this process corresponds to optimization of the marginal distributions with their maximums taken as the final outcome.

The algorithm is formulated as follows: Given a set of location estimates $\{\mathbf{x}_i | i \in N_k\}$ in time t and a set of means and variances of measured distances $\{(\mu_{ki}, \sigma^2_{ki}) | i \in N_k\}$, where N_k is the set of neighbors of node k, the algorithm searches for the best location estimate of node k in time step $t+1$. To determine this, for each point $\mathbf{x}$ of the considered space, it calculates the likelihood for the location of node k by the formula

$$P(\mathbf{x} = \mathbf{x}_k) \propto \prod_{i \in N_k} \mathcal{N}(\|\mathbf{x} - \mathbf{x}_i\|; \mu_{ki}, \sigma^2_{ki}), \tag{3}$$

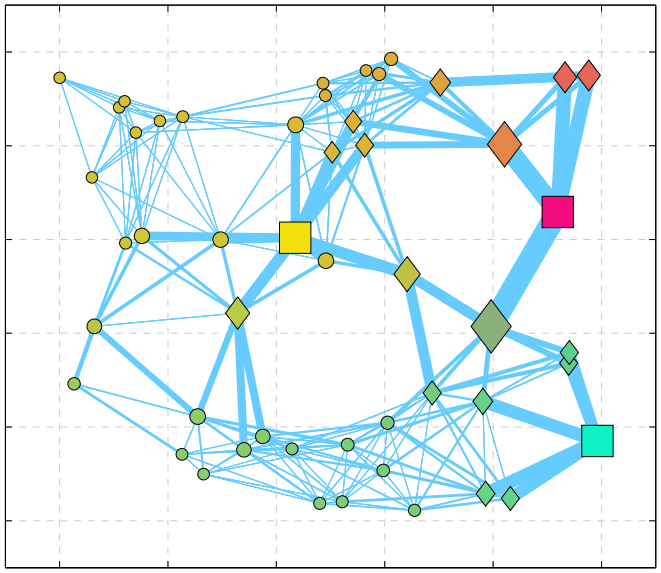
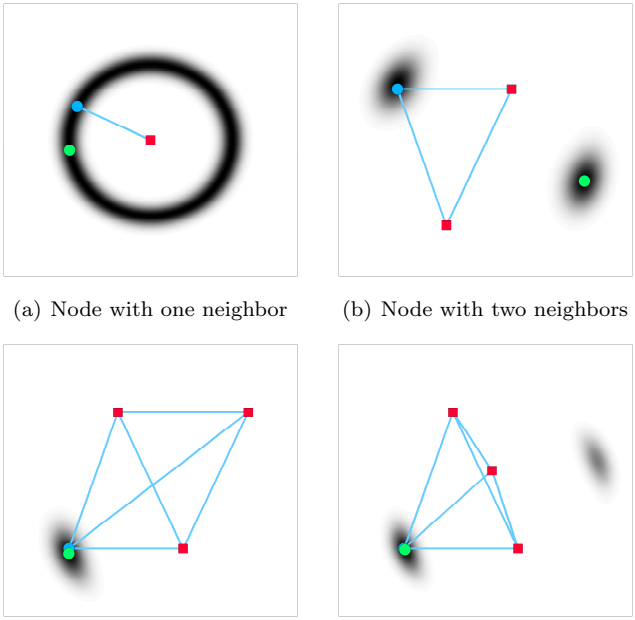


Figure 1: Membership vectors and the graph backbone. The beacons are represented by squares. Colors of the non beacon nodes are mixtures of the beacon nodes' colors in correspondence to their membership vectors. Link widths are proportional to $\|\mathbf{m}_i^* - \mathbf{m}_j^*\|$. Node size reflects the node strength of a non-beacon node. Beacon sizes are chosen arbitrarily. The identified backbone nodes are represented with diamonds.



(a) Node with one neighbor (b) Node with two neighbors

(c) Node with three neighbors (d) Node with three near-collinear neighbors

Figure 2: Location likelihood function of a node, given its neighbors' locations. Its actual location is marked with blue and its direct neighbors are marked with red. Its estimated position is marked with green.

where $\mathcal{N}$ is the normal density function. Furthermore, the negative log likelihood function can be written as

$$-lnP(\mathbf{x} = \mathbf{x}_k) = \frac{1}{2} \sum_{i \in N_k} \frac{(\|\mathbf{x} - \mathbf{x}_i\| - \mu_{ki})^2}{\sigma_{ki}^2} + const, \quad (4)$$

whose gradient is

$$\frac{d(-lnP(\mathbf{x} = \mathbf{x}_k))}{d\mathbf{x}} = \sum_{i \in N_k} \frac{(\|\mathbf{x} - \mathbf{x}_i\| - \mu_{ki})}{\sigma_{ki}^2} \frac{\mathbf{x} - \mathbf{x}_i}{\|\mathbf{x} - \mathbf{x}_i\|} \quad (5)$$

Employing this gradient in a typical gradient descent algorithm, the maximum likelihood estimation for the location of node k can be calculated. Visualization of the likelihood function in a two-dimensional plane is shown in Fig. 2. In the case when there are not enough neighbors, the localization estimate cannot be done unambiguously (Fig. 2(a) and 2(b)). On the other hand, the maximum likelihood can produce good results with 3 or more neighbors, as illustrated in Fig. 2(c). However, resulting from the probabilistic approach and the potential near-collinearity, the location likelihood function can have local maxima, see Fig. 2(d), whose basins of attraction may span significant area. Therefore, performing the maximum likelihood estimation multiple times from different initial positions increases the likelihood of finding the global maximum. In the simulations, the estimations are performed from 5 evenly distributed initial positions gathered from the local information.

As mentioned earlier, the purpose of the cooperation in our algorithm is position estimation utilizing neighboring estimates. In every iteration each node produces a point estimate of its location and transmits it to its neighbors. The information needed for a node's estimate is the locations of its neighbors received in the previous iteration.

Typically, distributed gradient descent is performed in most algorithms for analytically guaranteed convergence of the algorithm. However, the fixed point at which the system converges is not guaranteed to be the globally optimal state as the optimization function is usually not convex. Distributed gradient descent is highly prone to local minima of the joint distribution, since the steady convergence does not allow extensive exploring of the space. Additionally, it performs large number of iterations. In our approach, instead of using the gradient of the negative log likelihood for new position estimate in each iteration, we rely on the maximum likelihood optimal value. Furthermore, the optimal value is calculated using potentially different neighbors in each iteration (sampled according to the node strengths) during the earlier stages, for better exploration of the state space (further discussed in Section 2.7). After certain number of iterations (35 in our simulations), each node continues with gradient descent-like position estimation, which provides convergence. In most cases, however, the algorithm converges prior to this switching.

2.5. 2-hop neighborhood

Decentralized localization algorithms usually suffer from using only locations from the direct neighbors. When not

enough neighbors are connected, likelihood function can have multiple global maxima. Fig. 2(a) and Fig. 2(b) illustrate this case. Clearly, as a way to solve the global maxima problem, one solution would be to introduce more neighbors in some manner. Presuming the nodes use their maximal range to communicate, i.e., new direct neighbors cannot be added, the 2-hop neighbors are used (neighbors of the direct neighbors that are not also neighbors).

More generally, given a randomly deployed wireless sensor network, its global rigidity is not guaranteed. Global rigidity is considered to be a condition for the solvability of the network localization problem [3]. The rigidity problem is illustrated in Fig. 3. For the given graph, two configurations are possible, shown on Fig. 3(a) and Fig. 3(b). However, considering the wireless sensor network scenario, it should be taken into account that two non-neighboring nodes should not be in close proximity. Reason for this presumption is the communication range of the units, that is, a non-neighboring node must not be within the communication range of a given node. Consequently, for the given example, correct representation should only be the configuration in Fig. 3(b).

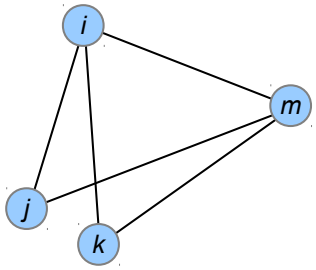
Information about the 2-hop neighboring locations can be received by message forwarding from the direct neighbors in the same iteration. As a result, the nodes broadcast information twice during a single iteration. We denote the union of 1-hop and 2-hop neighbors of node i with N_i^2. However, since direct distance measurements are not possible, they need to be approximated, i.e., the means and variances of their distributions need to be estimated.

Given one direct neighbor i to a referent node k, the task is to approximate the distance to node j, a direct neighbor of node i that is not neighbor with node k. The normal distributions of the distances between both neighboring pairs, characterized by their means and variances, are present as available information. Taking into account the rigidity argument, it is desirable nodes k and j to be localized further away from each other than they are from node i. Therefore, Gaussian addition, or convolution that also results in normal distribution, is introduced. So, the initial estimations for the mean and variance of the distance between k and j through the node i are

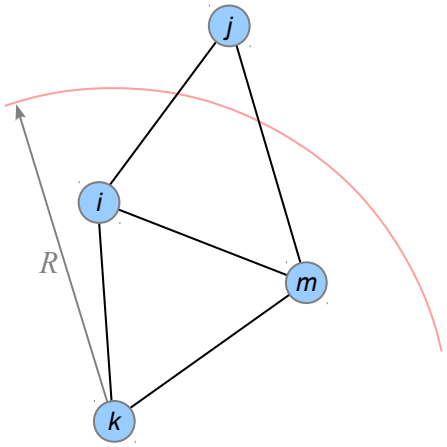
$$\mu_{kj,i} = \mu_{ki} + \mu_{ij} \tag{6}$$

$$\sigma^2_{kj,i} = \sigma^2_{ki} + \sigma^2_{ij} \tag{7}$$

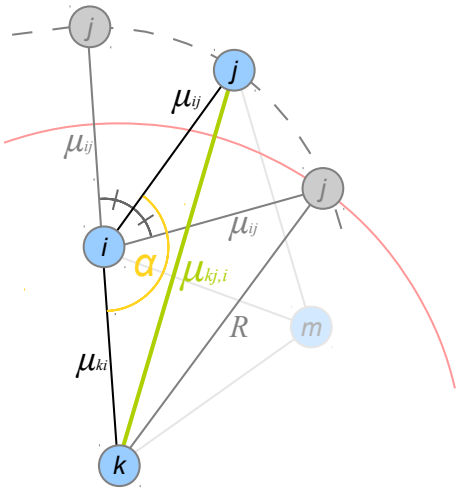
However, an underlying assumption here is that these three nodes are collinear, which most often is not the case. Hence, we should presume a better angle between the two neighboring segments ($\overline{ki}$ and $\overline{ij}$) than π (Fig. 3(c)). As there is also low probability that node j is positioned on the circle with center k and radius R (with minimal possible angle between the neighboring segments), we presume that the neighboring segments occupy the middle angle between the minimal and the maximal possible. This is illustrated in Fig. 3(c). Therefore, the formula for the



(a) Possible configuration of a non globally rigid graph



(b) Non globally rigid graph configuration, constrained by the communication range



(c) Approximation of 2-hop distance through one direct neighbor. The neighboring segments, $\overline{ki}$ and $\overline{ij}$, occupy the middle angle (α) between the minimal possible (yielding 2-hop distance R) and the maximal possible angle (π)

Figure 3: Illustration of the global rigidity problem. While a non globally rigid graph can be represented by multiple configurations (two in this example), the communication range constraints the possible locations of the 2-hop neighbors.

mean looks like

$$\mu_{kj,i}^2 = \mu_{ki}^2 + \mu_{ij}^2 - 2\mu_{ki}\mu_{ij}\cos\alpha \tag{8}$$

where

$$\cos\alpha = -\sqrt{\frac{R^2 - (\mu_{ki} - \mu_{ij})^2}{4\mu_{ki}\mu_{ij}}}; \tag{9}$$

In this manner, the 2-hop distance given one mutual neighbor between two non-neighboring nodes k and j is approximated. However, node k and node j can have more than one mutual neighbor. In this case, the 2-hop means and variances produced from every mutual neighbor are combined to refine the final approximation. Product of Gaussian probability density functions is used for that purpose, so the final formulas look like

$$\mu_{kj} = \frac{\sum\limits_{i \in N_k \cap N_j} \mu_{kj,i}/\sigma_{kj,i}^2}{\sum\limits_{i \in N_k \cap N_j} 1/\sigma_{kj,i}^2} \tag{10}$$

$$\sigma_{kj}^2 = \frac{1}{\sum\limits_{i \in N_k \cap N_j} 1/\sigma_{kj,i}^2} \tag{11}$$

Additionally, in many practical cases the communication range R is not fixed and equal for every node in the network. Although R usually depends on the transmission power of the node, the environmental conditions can affect the communication channel and consequently decrease R. We denote with R_{max} the maximal communication range that can be achieved in ideal conditions, i.e., that depends only on the transmission power. Further, we assume that the actual communication range $R \in (0, R_{max}]$ is distributed with some known distribution $P_{R_{max}}$, parameterized with R_{max}. Therefore, Eq. (9) can be formulated by

$$\cos\alpha = -\int_0^{R_{max}} P_{R_{max}}(R)\sqrt{\frac{R^2 - (\mu_{ki} - \mu_{ij})^2}{4\mu_{ki}\mu_{ij}}}\mathrm{d}R \tag{12}$$

Similarly, we can further approximate k-hop distances, for $k > 2$, causing k broadcasts in a single iteration, and even more - causing increase of the message lengths. With sufficiently large k, each node can receive information from significant percentage of the network nodes. Given the huge communication increment, a question arises whether decentralization is actually achieved. Additional question is how much the localization results are improved by increasing k. Here, we only use 2-hop distances; they do not cause communication overload and they address the rigidity problem adequately.

2-hop distance approximations have particular importance for the localization of the peripheral nodes, i.e., nodes with low connectivity to the beacon nodes. During the early iteration steps it is desirable that they are able to localize themselves further away from the *backbone* nodes. In that sense, 2-hop distance approximations give "direction" of the localization of a peripheral node.

2.6. Initialization and execution

Frequently used initialization approach ([24, 25]) is the localization of nodes that have multiple beacon nodes as neighbors at first, and then iterative localization of the rest of the nodes. This requires the beacon nodes to be placed close to one another centrally in the network. Other approaches require the beacon nodes to be placed on the boundary of the network [18, 26]. However, in most cases beacon assignment or beacon placement cannot be controlled, hence our algorithm does not depend on such assumptions. As stated earlier, the beacon nodes are randomly selected from the set of nodes at the beginning of the simulation process.

In our approach a node generates a position estimate if any of its neighbors (1-hop or 2-hop) have previously made position estimates. Initially only the beacon nodes have estimates, and in a few iterations every node in the network would have initial position estimate.

The algorithm then runs until every node's change in position estimate becomes negligibly small compared to the one from the previous iteration (Euclidean distance). However, there is no guarantee that the system will end up in the most optimal state. As most of the nodes will find their globally optimal position estimates, some clusters of nodes might end up in local optima. Certainly, there is interdependence of the outcomes of the position estimation process of the nodes. An inaccurate position estimate has the potential to inflict corruption to the position estimates of the other nodes, as the estimation process spreads from the beacon nodes to the rest of the network. Clearly, the nodes that are fairly connected, i.e., backbone nodes, are less likely to produce inaccurate estimates than the non-backbone nodes (peripheral nodes). Therefore, it is important to give more significance to the estimates of the backbone nodes than to the peripheral ones; particularly, the backbone nodes should not be influenced by the peripheral nodes during the initial time steps. The objective of this selective interdependence is better overall localization of the backbone and subsequent improved localization of the peripheral nodes; using 2-hop distance approximations they will localize themselves further away from the backbone.

2.7. Activation time and space exploration

The localization process is structured around the backbone: non-backbone nodes (that is, nodes with a smaller node strength) start the process later and are localized by using (already localized) backbone nodes. For this purpose, we introduce an activation function which is a linearly decreasing function that assigns activation time to each node in accordance to its node strength. Backbone nodes are activated in the first iteration. We also set the maximal activation time step, t_{max}^{act}, at which even the nodes with zero node strength have to be activated. The

activation function has the following form:

$$f^{act}(s_i) = \frac{1 - t^{act}_{max}}{\frac{1}{n-l}\sum_{j=l+1}^{n} s_j} s_i + t^{act}_{max}.$$

Further, during the earlier time steps, it is likely that some of the nodes will produce inaccurate position estimates. To lower their influence, we limit the number of neighbors that each node uses for producing its estimate. In this way, during every iteration each node explores different sets of neighbors (instead of executing the maximum likelihood from multiple initial positions) and chooses the one which results with the highest maximum likelihood value for the new position estimate. Since nodes with high strength are less likely to produce inaccurate estimations as opposed to other nodes, node strength is used during the weighted sampling process. The set $N_i^{act}(t) \subseteq N_i^2$ of which the node i samples in time t is the set of activated neighbors of node i. The number of used neighbors increases linearly with the time step. Consequently, nodes with high strength are most influential during the initial time steps. Backbone nodes are preferred over the other nodes.

Avoidance of local minima of some groups of nodes is possible with this feature. The minima usually occurs at highly connected groups of nodes with small strength. If during the initial steps, most of the nodes belonging to such group have inaccurate position estimates, they are more likely to get stuck in local minima due to the inner influence (high connectivity) in the group.

The good position estimates of the high strength nodes do not influence them enough to get the whole group out of the local minima. However, by introducing the limitation of used neighbors and activation times, we tackle the local minima problem as well. Since the low strength nodes are more likely to choose a subset of their neighbors consisted of high strength nodes, the inner group influence decreases and the group can escape local minima during the initial iterations.

Similar to the simulated annealing, during the earlier iterations, the algorithm is unstable. It is more sensitive to estimation changes, especially to the high strength nodes. Thus, by exploring the space more extensively, it is capable of escaping local minima in some cases. Later, when every node uses all of its neighbors for position estimation, the algorithm converges towards a good configuration. The rate at which this happens, however, depends on the average node strength. Networks in which there is high percentage of beacon nodes, hence high average node strength, naturally require less time for localization. On the other hand, networks with low percentage of beacon nodes or their bad topological placements (in terms of uniform placement and influence), hence with low average node strength, require more time to explore the state space. So, as a rule of thumb for the number of used neighbors of node i in each time step t we use

$$\min\left(3 + \frac{t}{3(n-l)} \sum_{j=l+1}^{n} s_j, \mid N_i^{act}(t) \mid\right).$$

3. Simulation results

Simulations were performed on computer generated networks in order to measure the precision of the algorithm. The settings that we used are commonly used in the literature - consisting of 50 nodes placed on a unit square with communication range of 0.3 or 0.25 with average degree of 10.5 and 7.6 in the network, respectively. We also performed simulations on networks containing obstacle of size 0.3×0.3 placed in the center with communication range of 0.3 and average degree of 9.9. The distance approximations were calculated from 20 normal distribution samples with mean equal to the actual distance and standard deviation equal to 10 percent of the distance. We varied the number of beacons from the minimal of 3 to 25. The results are obtained with respect to the non-beacon nodes only. Although the beacons were chosen randomly, for better performance assessment the beacon sets that consisted of nearly collinear beacons were resampled. That happened most often for the case with three beacons. Principal component analysis was applied to check whether the selected beacons are approximately collinear: if the set of beacon coordinates retained 95 percent of its variance by reducing the dimensionality to one dimension it is resampled. We do this to exclude the cases where there can be no unambiguous localization due to the collinearity of the chosen beacons, i.e., due to the symmetry that they produce. Basically, we exclude the cases that generate unfair negative impact on the performance of any localization algorithm. We do not take into account any other constraints of the beacon deployment. The figures produced for each number of beacons were averaged over 100 sampled networks. The results are presented in Fig. 4.

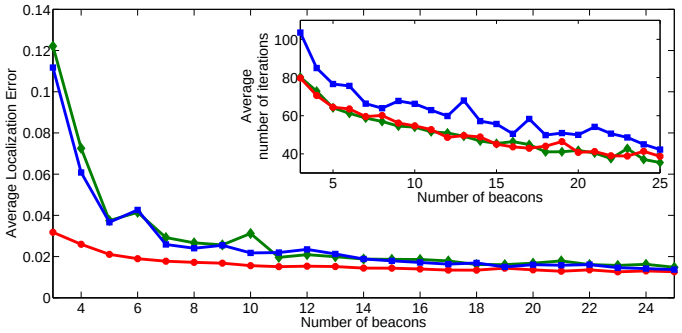


Figure 4: Average localization error as a function of the number of beacons. Performance of the networks with communication range of 0.3 is marked with red color, while the results for the networks with range 0.25 are marked with blue. Green denotes performance of the networks with communication range of 0.3 and an obstacle placed in the center. The inset image shows the average number of time steps needed for convergence of the algorithm.

We calculate the average localization error, that is, the average Euclidean distance between the actual and estimated locations. Other localization metrics are presented in [27]. As can be seen from the figure, the results show good performance. Even for the case with three beacons, the average localization error is below 0.03 for networks with communication range 0.3, while it is around 0.08 for the ones with range of 0.25. The networks that contain obstacle show worse performance with small number of beacons, as expected. We can also assert from the figure that the performance improves significantly when the number of beacons become greater or equal to 10 percent of the network size. The inset shows the average number of iteration steps carried out per number of beacons. As can be seen, this number gradually decreases from 80-100 to 40 iterations. Hence there is a trade-off between the number of beacons in the network and the communication needed for the algorithm to terminate. In addition, there are two broadcast messages sent from each node in every iteration, due to the 2-hop neighboring information exchange. The networks with greater communication range show better performance as expected, due to the greater average degree of the nodes.

The outage probability $P_{out}(e_{th})$ was computed as well (introduced in [6]): the average percentage of mislocalized nodes is calculated for a specified allowable error e_{th} (tolerance threshold). This says should an acceptable error is specified, what percentage of nodes would have localization error larger than that acceptable error. Simulations were ran on both scenarios described in [6], with 100 and 50 non-beacon nodes and 13 beacon nodes, communication range of 0.2, averaged over 20 sampled networks with average degrees of 11.8 and 6.5, respectively. The results are presented in Fig. 5. For networks with 100 non-beacon nodes, 99 percent of the nodes are correctly localized (1 percent are in outage) with allowable error of only 0.033, whereas the same success rate is achieved with 10-fold higher allowable error of 0.3 in the case of 50 non-beacon nodes. On the other hand, 10 percent of the nodes are in outage should the allowable error is 0.02 and 0.045 for the case with 100 and 50 non beacon nodes, respectively. Hence, 90 percent of the non beacon nodes are correctly localized for small allowable error, even in the sparse network case.

To examine whether the algorithm can be scaled, we compare results obtained from networks with 50, 100, 500 and 1000 nodes with number of beacons equal to 10 percent of the number of nodes. The network radius ranges are set to produce average node degree of about 10. We calculate the average localization error, averaged over 10 sampled networks. As can be seen from the results in Fig. 6, the localization error drops as the network size grows. Increasing the network size while maintaining the average degree constant can be interpreted as viewing the nodes on a different resolution scale. Hence, a decrease in the global error can be expected. However, the relative (to the communication range) error does not decrease significantly. Also note that the threshold at which the algorithm terminates produces resolution limit; beyond that, the error does not decrease with the network size. On the other hand, the average number of performed iterations is roughly constant. We can conclude that the algorithm scales good with the network size, in terms of performance as well as time steps needed for convergence. It also performs well with low number of beacon nodes, 10 percent in this case.

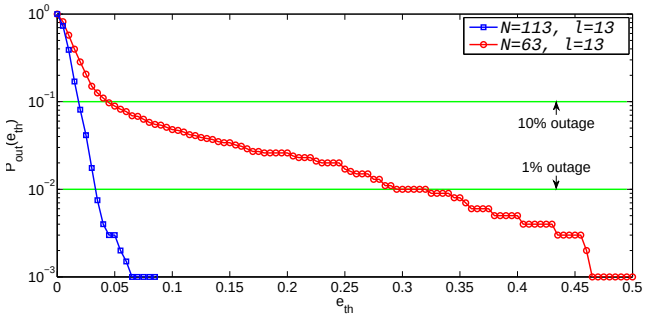


Figure 5: Outage probability per allowable error for networks of 100 and 50 non beacon nodes with 13 beacons and radius range 0.2, deployed on a unit square.

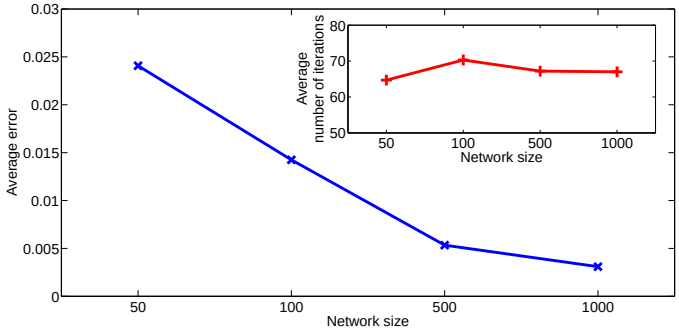
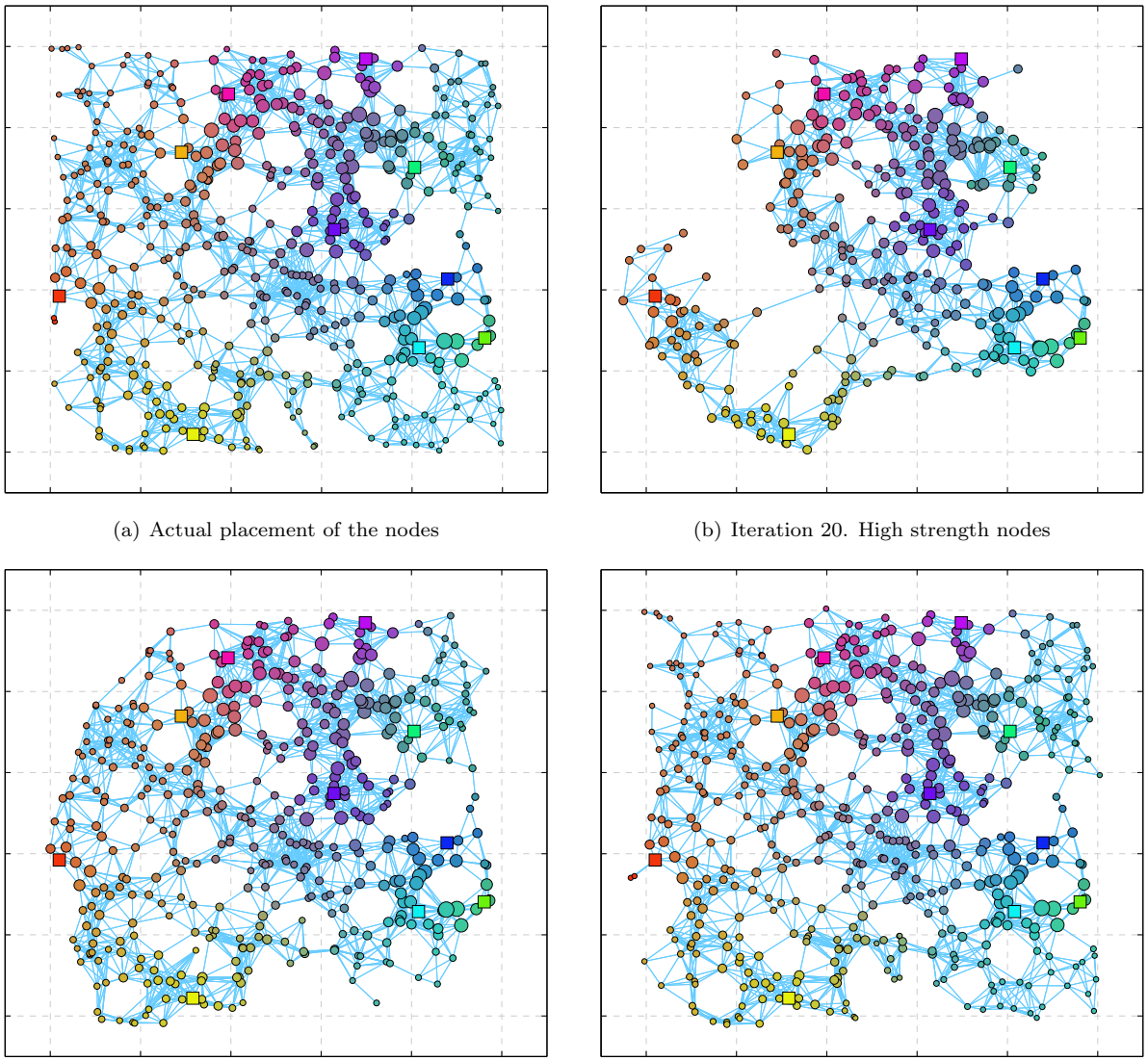


Figure 6: Comparison of average localization error and number of iterations for networks with different sizes. The network radius ranges are set to produce average node degree of about 10. 10 sampled networks are used for averaging.

Sample execution of the algorithm is presented on Fig. 7 on a network with 500 nodes and 10 beacons (Fig. 7(a)). The progress illustrates the properties of the algorithm. The early activation of the backbone nodes can be observed in iteraction 20 (Fig. 7(b)). Subsequent activation of the peripheral nodes and their localization away from the backbone, aided by the 2-hop distance approximations, is shown in iteration 40 (Fig. 7(c)). Finally, almost complete recovery of the original positions can be observed before termination of the algorithm, in iteration 80 (Fig. 7(d)).

4. Conclusion

In this paper we introduced several novel features incorporated in an example of distributed, cooperative algorithm for WSN localization. Based on these features, the



(a) Actual placement of the nodes

(b) Iteration 20. High strength nodes

(c) Iteration 40. 2-hop distance approximation

(d) Iteration 80. Fill network position estimation

Figure 7: Sample execution of the algorithm on network with 500 nodes, 10 beacons and communication range 0.09. In (a) the actual network deployment is depicted. In (b), (c) and (d) the estimation states at iteration 20, 40 and 80 are depicted, respectively. Beacon nodes are denoted with squares and distinguished by color. Non beacon nodes are denoted with circles; their size depicts node strength and color is mixture of the beacon nodes' colors in correspondence to their membership vectors.

process of localization, in addition to known beacon locations, uses multiple distance measurements (point estimates) in order to calculate a mean and variance that represent the distance distribution more accurately. No additional assumptions, such as certain beacon distribution are necessary. Node strength is introduced as a measure of the connectivity of a node toward the beacons. High strength nodes are topologically located between the beacons, i.e., are placed on the relatively short paths between them. The expanding localization estimation is structured around these nodes that comprise the network backbone. Each node estimates its location by using its neighbors' locations from the previous iteration. Initially not every neighbor is used in the estimation process. In this way, we successfully tackle the local optima problem and localize the backbone of the network more efficiently. Hence, the neighbors that are used are sampled from the set of all active neighbors according to their node strength. Also, nodes are activated in accordance to their node strength. Furthermore, each node also uses its 2-hop neighbors' locations to avoid the rigidity problem, i.e., to deal with the situations where there is not enough information from the direct neighbors to produce unambiguous localization. For that purpose, we introduced 2-hop neighboring distance approximation. The simulation results from computer generated networks suggest that algorithms that incorporate the proposed features cope well with the known challenges of WSN localization. The obtained results are good even in the case of small number of beacons, while the performances are scalable with the network size.

Acknowledgements

The authors are grateful for the supports provided by the Office of Naval Research (ONR, Code 30) and ONR-Global (London Office).

References

[1] A. Boukerche, H. Oliveira, E. Nakamura, A. Loureiro, Localization systems for wireless sensor networks, wireless Communications, IEEE 14 (6) (2007) 6–12.

[2] N. Patwari, J. Ash, S. Kyperountas, A. Hero III, R. Moses, N. Correal, Locating the nodes: cooperative localization in wireless sensor networks, Signal Processing Magazine, IEEE 22 (4) (2005) 54–69.

[3] J. Aspnes, T. Eren, D. Goldenberg, A. Morse, W. Whiteley, Y. Yang, B. Anderson, P. Belhumeur, A theory of network localization, Mobile Computing, IEEE Transactions on 5 (12) (2006) 1663–1678.

[4] G. Mao, B. Fidan, B. Anderson, Wireless sensor network localization techniques, Computer Networks 51 (10) (2007) 2529–2553.

[5] I. Amundson, X. Koutsoukos, A survey on localization for mobile wireless sensor networks, Mobile Entity Localization and Tracking in GPS-less Environnments (2009) 235–254.

[6] H. Wymeersch, J. Lien, M. Win, Cooperative localization in wireless networks, Proceedings of the IEEE 97 (2) (2009) 427–450.

[7] M. Win, A. Conti, S. Mazuelas, Y. Shen, W. Gifford, D. Dardari, M. Chiani, Network localization and navigation via cooperation, Communications Magazine, IEEE 49 (5) (2011) 56–62.

[8] R. K. Patro, Localization in wireless sensor network with mobile beacons, in: Electrical and Electronics Engineers in Israel, 2004. Proceedings. 2004 23rd IEEE Convention of, IEEE, 2004, pp. 22–24.

[9] L. Doherty, K. S. Pister, L. El Ghaoui, Convex position estimation in wireless sensor networks, in: INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 3, IEEE, 2001, pp. 1655–1663.

[10] D. Niculescu, B. Nath, Dv based positioning in ad hoc networks, Telecommunication Systems 22 (1-4) (2003) 267–280.

[11] C. Savarese, J. M. Rabaey, K. Langendoen, Robust positioning algorithms for distributed ad-hoc wireless sensor networks, in: USENIX technical annual conference, USENIX Association, 2002, pp. 317–327.

[12] Y. Shang, W. Rum, Improved mds-based localization, in: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, Vol. 4, IEEE, 2004, pp. 2640–2651.

[13] 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, ACM, 2004, pp. 50–61.

[14] 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, 2002, pp. 112–121.

[15] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. Anderson, A. S. Morse, Y. R. Yang, Localization in sparse networks using sweeps, in: Proceedings of the 12th annual international conference on Mobile computing and networking, ACM, 2006, pp. 110–121.

[16] X. Wang, J. Luo, Y. Liu, S. Li, D. Dong, Component-based localization in sparse wireless networks, IEEE/ACM Transactions on Networking (ToN) 19 (2) (2011) 540–548.

[17] J. Zhao, W. Xi, Y. He, Y. Liu, X.-Y. Li, L. Mo, Z. Yang, Localization of wireless sensor networks in the wild: Pursuit of ranging quality, IEEE/ACM Transactions on Networking (TON) 21 (1) (2013) 311–323.

[18] U. Khan, S. Kar, J. Moura, Distributed sensor localization in random environments using minimal number of anchor nodes, Signal Processing, IEEE Transactions on 57 (5) (2009) 2000–2016.

[19] L. Zhang, L. Liu, C. Gotsman, S. J. Gortler, An as-rigid-as-possible approach to sensor network localization, ACM Transactions on Sensor Networks (TOSN) 6 (4) (2010) 35.

[20] M. Cucuringu, Y. Lipman, A. Singer, Sensor network localization by eigenvector synchronization over the euclidean group, ACM Transactions on Sensor Networks (TOSN) 8 (3) (2012) 19.

[21] D. Shamsi, N. Taheri, Z. Zhu, Y. Ye, Conditions for correct sensor network localization using sdp relaxation, in: Discrete Geometry and Optimization, Springer, 2013, pp. 279–301.

[22] A. Stanoev, D. Smilkov, L. Kocarev, Identifying communities by influence dynamics in social networks, Physical Review E 84 (4) (2011) 046102.

[23] A. Stanoev, D. Smilkov, Consensus theory in networked systems, in: Consensus and Synchronization in Complex Networks, Springer Berlin Heidelberg, 2013, pp. 1–22.

[24] J. Albowicz, A. Chen, L. Zhang, Recursive position estimation in sensor networks, in: Network Protocols, 2001. Ninth International Conference on, IEEE, 2001, pp. 35–41.

[25] H. Oliveira, A. Boukerche, E. Nakamura, A. Loureiro, Localization in time and space for wireless sensor networks: An efficient and lightweight algorithm, Performance Evaluation 66 (3) (2009) 209–222.

[26] G. Wu, S. Wang, B. Wang, Y. Dong, S. Yan, A novel range-free localization based on regulated neighborhood distance for wireless ad hoc and sensor networks, Computer Networks.

[27] H. Aksu, D. Aksoy, I. Korpeoglu, A study of localization met-

rics: Evaluation of position errors in wireless sensor networks, Computer Networks 55 (15) (2011) 3562–3577.

Angel Stanoev holds BSc and MSc degrees in Computer Science and Electrical Engineering from the Faculty of Electrical Engineering and Information Technologies at the Ss. Cyril and Methodius University, Skopje. As a researcher in the Laboratory for Complex Systems and Networks at the Macedonian Academy of Sciences and Arts, his research activities included modeling of multi-agent cooperation, complex systems and networks. He is currently a PhD candidate at the Max Planck Institute of Molecular Physiology in Dortmund, working in the field of systems biology.

Sonja Filiposka is an assistant research and teaching professor at the Faculty of Computer Science and Engineerind at the Ss. Cyril and Methodius University. Her research interests include Wireless ad hoc and sensor networks, System on chip design, Parallel processing, Complex systems and networks, Electromagnetic compatibility and Social impact on information theory.

Visarath In

Visarath In graduated from Georgia Institute of Technology with a Ph.D. in Physics in 1996. He works in the field of nonlinear dynamics in both theoretical and experimental areas and has in depth experience in the fields relating to mechanical, electronic, and nonlinear sensor systems. Currently, Dr. Visarath In is working on the coupled nonlinear systems with special emphasis on the fluxgate magnetic sensors, electric field sensors, coupled nonlinear gyroscopes for navigation, signal generation & multiple frequency generation systems, and Nonlinear Channelizer System for RF communication. He has over 50 publications and 24 patents.

Ljupco Kocarev is Professor at the Faculty of Computer Science and Engineering, Ss Cyril and Methodius University, Skopje, Macedonia and Researcher (full professor rank) at the BioCircuits Institute, University of California San Diego, USA. His scientific interests include nonlinear systems and circuits, coding and information theory, computer and complex networks, cryptography, bioinformatics and computational biology. He has coauthored more than 100 journal papers in more than 20 different international peer-reviewed journals ranging from mathematics to physics and from electrical engineering to computer sciences. According to the Science Citation Index, his work has been cited more than 5000 times. He is a member of the Macedonian Academy of Sciences and Arts and Fellow of IEEE.

Angel Stanoev

Sonja Filiposka

Visarath In

Ljupco Kocarev