Range-Free Localization in Sensor Networks: Handling Heterogeneous Scenarios

GIANNI GIORGETTI
Arizona State University
SANDEEP K.S. GUPTA
Arizona State University
and
GIANFRANCO MANES
Università degli Studi di Firenze

Implementing a localization service for a wireless sensor network is a challenging task. Sometimes the nodes are deployed in sparse topologies, while other times they are densely packed inside a building. Some environments are relatively uncluttered, while others have obstacles that impede the node placement and strongly affect the radio signal. To address the problem of localization in heterogeneous scenarios, we present a range-free scheme based on the neural network paradigm of *Self-Organizing Maps* (SOM). This method is lightweight, works with or without anchor nodes, and has proven effective in a variety of simulated scenarios. We propose three variants (SOM-V, SOM-A and SOM-R) that achieve accurate results in sparse topologies but are also suitable to localize nodes in dense networks or deployments with anisotropic layout. We evaluate the localization results using extensive simulations, comparisons with other range-free techniques, Cramér-Rao bound analysis and test cases with data from in-field measurements. Finally, we demonstrate analytically that the proposed scheme has low computation and communication overheads, making it suitable for resource-constrained networks.

Categories and Subject Descriptors: C.2 [**Computer Communication Networks**]: Network Protocols

General Terms: Algorithms, Performance, Design

Additional Key Words and Phrases: Self-Organizing Maps, Localization, Wireless Sensor Networks

1. INTRODUCTION

Network localization is a challenging research area that has been actively investigated over the last few years. Since not every node can be augmented with a GPS receiver, the node positions have to be computed using information collected within the network. The information available for node localization usually consists of a set of constraints, such as distance measurements or connectivity between nodes,

Author's address: G. Giorgetti and S.K.S. Gutpa, School of Computing and Informatics, Arizona State University, 699 S Mill Ave, Tempe, AZ 85287, USA.
G. Manes: Università degli Studi di Firenze, Via di S. Marta 3, 50139 Firenze, ITALY.

Scheme	Error	Simulation Scenario			Reference
		nodes	anchors	connectivity	
Convex opt.	0.5 R	200	50	5.7	[Doherty et al. 2001]
DV-HOP	0.25 R	100	30	7.6	[Niculescu and Nath 2003]
Amorphous	0.3 R	200	6	20	[Nagpal et al. 2003]
APIT	0.5 R	n/a	18	8	[He et al. 2003]
MDS	0.3 R	200	4	16	[Shang et al. 2003]

Table I: Typical performance of some range-free localization schemes.

and optionally, the absolute position of some nodes (*anchor nodes*). Localization is a difficult problem, both in theory and in practice. 1) From a theoretical perspective, localization is analogous to the problem of *embedding* a graph in a Euclidean space. If we exclude selected cases[1], this problem is NP-hard even assuming error-free range measurements or ideal connectivity [Saxe 1979; Breu and Kirkpatrick 1998]. 2) In addition to the computational complexity of finding a solution, the result may be ambiguous, i.e. multiple solutions are admissible, when not enough constraints are available [Eren et al. 2004]. This situation likely arises in Wireless Sensor Networks (WSNs) because nodes have limited communication/sensing range and can only interact with a few neighbors. 3) In practice, since nodes use inexpensive sensors and are deployed in uncontrolled environments, the measurements are not only difficult to obtain, but also corrupted by substantial noise that increases the uncertainty in the results [Niculescu and Nath 2004; Whitehouse et al. 2005].

The difficulties above explain the large number of localization schemes proposed over the past few years. Existing solutions, based either on formal approaches or heuristics, can be classified broadly in two families: *range-based* schemes that use distance or angle estimates, and *range-free* schemes that use proximity information such as radio connectivity. In dense networks, range-based localization produces accurate results when the measurements have a high *signal to noise ratio* (SNR). However, range estimation generally relies on additional hardware, which adds to the cost and size of the nodes, and it is less suitable for ad-hoc deployments. For example, the ultrasound transceivers commonly used to measure distances have a range of only few meters unless properly aligned, and the antenna or microphone arrays used for *Angle of Arrival* (AoA) estimates need to be aligned perpendicularly to the plane of the incoming signal. On the other hand, range-free schemes offer more coarse-grained resolution but are cheaper, easier to deploy, and potentially better suited for use in noisy environments [Niculescu and Nath 2004; Beutel 2005].

1.1 Range-Free Localization

Schemes such as *Centroids* [Bulusu et al. 2000], convex position estimation [Doherty et al. 2001], DV-HOP [Niculescu and Nath 2003] and MDS [Shang et al. 2003] are popular localization techniques that use only the connectivity data derived from radio communication. As shown in Table I, these schemes achieve a localization

[1]The localization problem can be efficiently solved in networks where a high number of error-free inter-node distances and angles are known. See the work using semidefinite programming by Biswas and Ye [2004] or the definition of *trilateration* graphs in [Aspnes et al. 2006].

Variant	Input Data			Applications
	Hop Dist.	Anchors	RSS	
SOM-V	•	–	–	virtual coordinates, geo-routing
SOM-A	•	•	–	absolute coord., sparse and anisotropic networks
SOM-R	•	•	•	abs. coord., sparse, dense and anisotropic networks.

Table II: Proposed localization schemes and their applications. Dots (•) denote the information used by each variant of the scheme.

error of 0.25 R to 0.5 R, that is 25% to 50% the maximum communication range. If these figures meet the application requirements, the estimated positions are useful not only to locate the sensors, but also to implement services such as geographic routing and target tracking. In fact, He et al. [2003] have shown the performance of these services do not significantly decrease provided that the localization error is below 0.4 R–0.5 R.

Since many applications tolerate approximate positions, range-free approaches adapts well to localization in low-cost sensor networks. However, given the variety of applications targeted by WSNs, the localization results may degrade significantly depending on the deployment scenario at hand. Network configurations that may cause the localization error to increase to values substantially higher than those in Table I include: i) networks with limited connectivity or low percentage of anchor nodes, ii) dense networks where most of the nodes are in the radio range of each other, iii) networks where large obstacles interfere with wireless communication and/or cause anisotropic layouts. We address the challenges of implementing localization services in these heterogeneous scenarios by revising and extending a previously presented range-free scheme that had already been shown to work well in sparse topologies [Giorgetti et al. 2007].

1.2 Contributions and Paper's Outline

The contribution of our research is three-fold. First, we describe a novel range-free scheme based on a neural network formalism known as the *Self-Organizing Map* (SOM). We present few variants of the solution that are designed to work with or without anchor nodes, and additionally to use *Received Signal Strength* (RSS) information (see Table II). Second, we perform extensive simulation to characterize the performance of the solution and its variants in a variety of simulation scenarios. The results are compared with those of the popular DV-HOP and MDS schemes, showing significant improvements. Third, we analytically evaluate algorithm's computation and communication complexity, demonstrating that the solution is compatible with the limited hardware resources of typical sensor nodes.

The proposed schemes compute the node positions by using the hop count distances between sensors and exploiting the topological ordering properties of the SOM paradigm (see Section 2). Although other SOM-based localization schemes have been presented, to the best of our knowledge, our work is the first to use SOM for localization based on radio-connectivity (see Section 3) and the first to systematically evaluate the performance of SOM as a tool for sensor network localization. The result of our approach is a simple yet powerful technique that we tested on a wide range of localization scenarios. A summary of the main results follows.

—**Anchor-Free or Anchor-Based localization.** The proposed method works with or without anchor nodes. In Section 4 we show that the anchor-free version (SOM-V) generates virtual coordinates that are effective when used for geographic routing. When anchor nodes are available, this information is used during the training phase of the map (SOM-A) to generate absolute coordinates (see Section 5). Using only four anchor nodes, SOM-A achieves a localization error as low as 0.3 R for networks with an average connectivity of five.

—**Robustness to shadowing effects.** In addition to the ideal radio model widely used in the literature, we consider a more realistic propagation model that accounts for the variability of the RF signal due to shadowing effects. We compare the results to the Cramèr-Rao bound, and we show that, for low connectivity values, SOM-A's error is close to the theoretical bound (see Section 6). Furthermore, SOM-A's accuracy degrades more gracefully than that of MDS and DV-HOP when strong shadowing of the signal alters the network connectivity.

—**Localization in dense networks.** When most of nodes are in the radio range of each other (e.g. indoor applications or smart spaces), connectivity data carry little information about the node positions, and range-free schemes achieve poor performance[2]. In Section 7, we address this problem by using RSS information, and we propose a solution (SOM-R) whose accuracy is independent from the network connectivity. We validate the scheme using extensive simulations and data from three in-field measurements with node in 2D and 3D spaces.

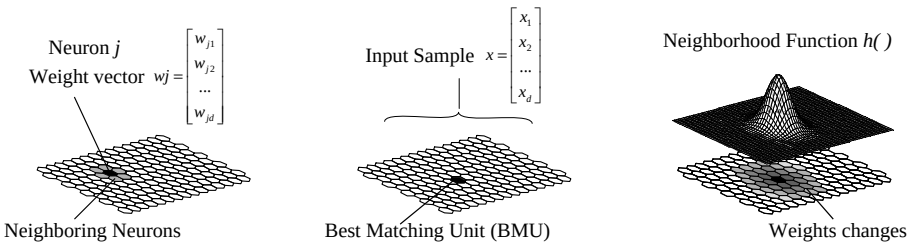
—**Localization in anisotropic networks.** We evaluate our SOM schemes on topologies with anisotropic layout, which occur, for example, when sensors are deployed in regions with obstacles (e.g. tall buildings). Localization in anisotropic networks have been studied by several authors and have motivated the use of specialized techniques [Shang and Ruml 2004; Lim and Hou 2005; Wang and Xiao 2008]. In Section 8, we show that the SOM paradigm leads to a solution that is naturally suited to localize anisotropic layouts. Without modification to the base schemes, SOM-A and SOM-R achieve localization results comparable to those obtained for regular topologies.

In Section 9, we evaluate the computational and communication complexity of the solution. The SOM technique is a centralized method, but the result of our analysis and benchmark tests on real hardware show that the lightweight solution is suitable for execution on simple nodes with limited computational and memory resources. We discuss related work in Section 10, with particular emphasis on the theoretical aspects of the localization problem and the Self-Organizing Map technique.

2. SELF-ORGANIZING MAPS

Introduced in the early 80's, the *Self-Organizing Map* (SOM) [Kohonen 1982; 1993] is a neural network where each neuron contains a weight vector, also called *model* or *reference* vector, that is updated during the training phase of the map. The neurons are arranged in regular geometric structures, typically two-dimensional lattices with rectangular or hexagonal patterns like the one in Figure 1a. As we will see in

[2]In the extreme case of a fully connected network, since all the nodes have the same neighbors, they will eventually be associated to the same, or a similar, position.



(a) SOM (b) Competition (c) Adaptation

Fig. 1: a) Self-Organizing Map with hexagonal pattern; b, c) two steps of the training algorithm.

the following sections, the structure of the map and the learning algorithm result in a versatile architecture that has found numerous applications in the context of exploratory data analysis, pattern recognition and vector quantization. An extensive bibliography of SOM papers has been initially compiled by Kaski et al. [1998] and successively updated by Oja et al. [2003].

2.1 Learning Algorithm

SOM implements *unsupervised* learning, meaning that the map is able to learn the properties of the training set without the aid of labeled samples or reward functions. Assuming that the information to learn is contained in a large and potentially continuous input set with elements $\mathbf{x_i} \in R^d$, the map produces a compact representation of the training set using a finite number of reference vectors $\mathbf{w_j} \in R^d$. The weights $\mathbf{w_j}$ are initialized with random values and updated by executing multiple iterations of the following three-step sequence:

(1) **Sampling**: A sample is extracted from the training set and presented to the network. We let $\mathbf{x}(n)$ denote the sample at the current iteration.

(2) **Competition**: The sample $\mathbf{x}(n)$ is compared with the map weights using a distance function. The neuron whose weight is closer to $\mathbf{x}(n)$ wins the competition and become the *Best Matching Unit* (BMU) (Figure 1b). If the distance function is implemented using the Euclidean distance, the election rule is:

$$c = \arg\min_j \|\mathbf{x}(n) - \mathbf{w_j}(n)\|\,, \tag{1}$$

where c denotes the index of the BMU, and $\|\cdot\|$ is the Euclidean norm.

(3) **Adaptation**: The weight vector $\mathbf{w_c}$ associated with the BMU is updated to increase its similarity with the input sample. During the adaptation process, the BMU activates nearby neurons (*cooperative learning*) allowing them to learn some of the information contained in $\mathbf{x}(n)$. The update rule is:

$$\mathbf{w_j}(n+1) = \mathbf{w_j}(n) + \eta\, h_{cj}[\mathbf{x}(n) - \mathbf{w_j}(n)], \tag{2}$$

where η is the global *learning rate* parameter and h_{cj} is the value of *neighborhood function* that controls the adaptation for neurons close to the BMU (see Figure 1c).

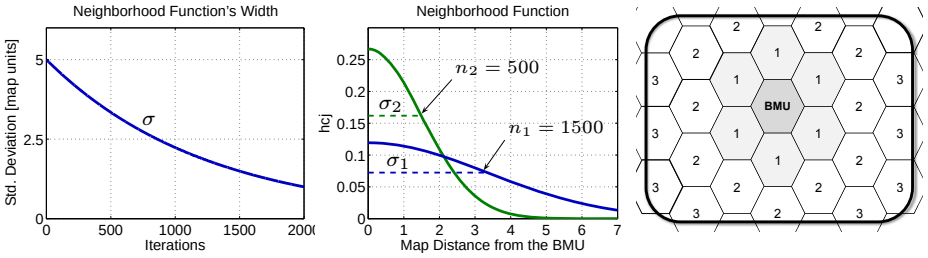


Fig. 2: a) Exponential scheduling for the parameter σ; b) neighborhood function at different number iterations; c) portion of a SOM showing the map distances from the BMU.

2.1.1 *Learning Parameters.* For ensuring convergence, the learning rate η should be computed using a function $\eta(n)$ that decreases monotonically with the number of iterations. Optimal choices of $\eta(n)$ have been discussed in the literature [Mulier and Cherkassky 1994; Kohonen 2001], however, in practice, the exact form of $\eta(n)$ is not a critical factor in the SOM technique.

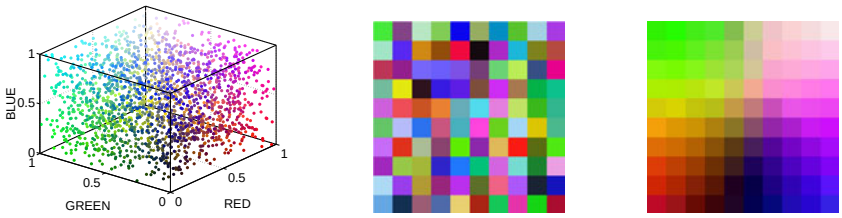
The update rule (2) is also controlled by the value h_{cj} that determines the amount of information learned by neurons close to the BMU. The value h_{cj} can be constant for all the neurons within a given distance from BMU when a *step* neighborhood function is used, or it can be computed using a smoothing kernel. As shown in Figure 1c, a common choice is to calculate h_{cj} by using a Gaussian function:

$$h_{cj} = \exp\left(-\frac{d_{\text{map}}(c,j)^2}{2\sigma^2}\right), \tag{3}$$

where $d_{\text{map}}(\cdot,\cdot)$ measures the distance on the map between two neurons, and the parameter σ controls the *width* of the smoothing kernel. As in the case of the learning rate, the parameter σ should be computed using a function $\sigma(n)$ that decreases monotonically. Large values of σ during the initial iterations result in a wide neighborhood function that allows the map to quickly organize the neurons, while the smaller values at the end of the training determine localized changes, allowing the map to describe different input features. Figures 2a,b show an example of exponential scheduling for the parameter σ and the resulting neighborhood function at iteration $n_1 = 500$ and $n_2 = 1500$. Figure 2c shows part of a hexagonal SOM with labels indicating the distances between the BMU and nearby neurons. Whenever the neurons are updated, the map distance and the current value of σ are used to compute the value h_{cj} that controls the adaptation level of each weight.

2.2 Properties

We show a few SOM's properties by using an example where samples from the RGB color space in Figure 3a are used to train a 10×10 map. Both the samples and the map weights are represented by vectors $[r_i, g_i, b_i]$ containing the the red, blue and green color components. Figure 3b shows the initial SOM with weights randomly assigned. After training the map with a few thousands samples from the input space, the final weights assume the values shown in Figure 3c. The results illustrate the following properties:



(a) Input Space (b) Initial Weights (c) Final Weights

Fig. 3: 10×10 **SOM trained with samples from the RGB color space.**

(1) SOM implements a *projection* technique: the three-dimensional input space is mapped onto a two-dimensional surface.
(2) SOM implements a *Vector Quantization* (VQ) technique. In this case, 100 vectors were selected as representative values of a much larger input set.
(3) SOM generates *topologically ordered* results, in the sense that similar information is mapped to nearby locations. This property emerges as a consequence of the update rule: since adjacent neurons are subjected to similar weight changes, they eventually converge to similar values.

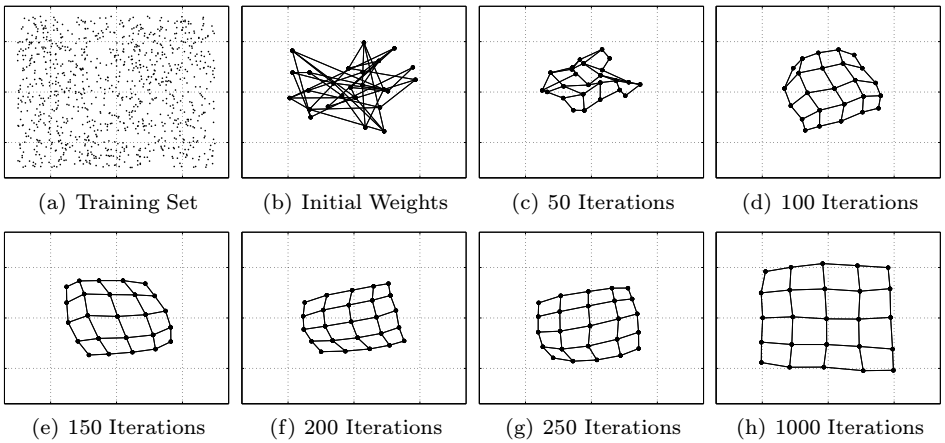
3. LOCALIZATION USING SOMS

As seen in the previous section, a SOM can be used to process a large amount of multi-dimensional information and represent it using a compact, low-dimensional model. After training a map, the same election rule (1) discussed in Section 2.1 can be used to translate new samples into their corresponding *codebooks* (vector quantization) or to project points onto a two-dimensional surface.

These properties has been used in the past to implement localization schemes for mobile robots in unknown environments [Janet et al. 1997; Gerecke and Sharkey 1999]. As the robot explores a new space, the multisensory data collected by on-board sensors are fed to a SOM that organizes them on the basis of their similitude. Assuming that sensor readings are correlated with their positions, the SOM defines a virtual map for the space just explored: the robot's location is given by the BMU that matches the current sensor readings. Ertin and Priddy [2005] have applied the same concept to the localization problem in WSNs. In their work, synchronous snapshots gathered from the sensors are used to train a SOM, producing a set of weights that define a grid of so-called *virtual sensors*. The node coordinates are approximated by the grid position of the virtual sensor that matches the actual sensor measurements. The authors suggest possible application to target tracking. A similar approach has been used by Sakurai et al. [2005] for human tracking in an indoor space and by Xu et al. [2007] for localization of mobile users using signal strength values from cellular base stations. Finally, Takizawa et al. [2006] have proposed a range-based scheme based on an update rule similar to the one used by SOM. These approaches are discussed in more details in Section 10.

3.1 Proposed Approach

Our work differs from previous SOM approaches because we do not assume the availability of sensors readings or range estimates, and we do not use the concept



(a) Training Set (b) Initial Weights (c) 50 Iterations (d) 100 Iterations

(e) 150 Iterations (f) 200 Iterations (g) 250 Iterations (h) 1000 Iterations

Fig. 4: 5×5 **SOM trained with random samples from a 2D training set.**

of virtual sensors. *Our scheme uses proximity information derived by radio communication and explicitly compute each node's position during the map's training.* The proposed solution is based on two simplifying assumptions:

(1) The sensor distribution is (approximately) uniform in the deployment area.
(2) Nodes that are radio neighbors are relatively close to each other.

We will successively relax these two assumptions by considering non uniform deployments and more realistic propagation models, however 1) and 2) are useful to illustrate how the SOM technique leads to an intuitive localization scheme.

Imagine that the deployment area is the square region in Figure 4a and that a large number of points $[x_i, y_i]^t$ are sampled inside this area and used to train a SOM, say a 5×5 square map. Since the training samples and the map vectors have the same structure, each weight defines a position in the 2D plane. Figure 4b shows the values of the random weights, where segments of line are used to link the positions of adjacent neurons. As the map is trained, the weights assume the values shown in Figure 4c-f. Similarly to the example of Figure 3, the SOM weights approximate the input distribution, and the weights of neurons that are close on the map converge to similar values. We note that the coordinate assignment in Figure 4f is compatible with the positions of an hypothetical 25 node WSN that meets the two assumptions stated at the beginning of this section. More in general, we observe that the weights of a SOM trained with points from a 2D uniform distribution can be used as an approximation for the positions of a set of wireless nodes. For this purpose, the number of neurons in the SOM needs to match the number of nodes in the WSN, and the map has to be organized in such a way that neighbor nodes are associated to adjacent neurons. In the next sections we formalize the use of SOM as a tool for sensor network localization.

3.1.1 *System Model.* We consider a connected network with N nodes placed at unknown locations. None of the nodes is equipped with hardware for position, range or angle estimation, and no assumption is made regarding the availability of sensors. We only assume that the nodes can determine their radio neighbors. We

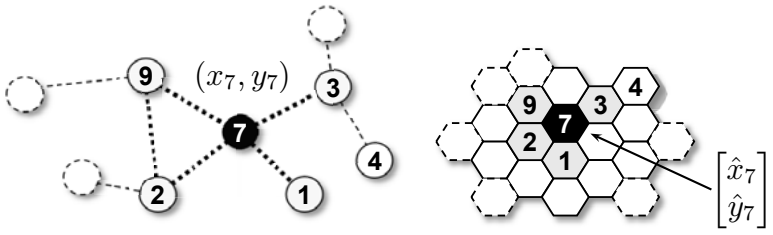


Fig. 5: Correspondence between the nodes in a WSN (left) and the neurons in a SOM (right). The arrangement of neurons in the SOM describes the neighborhood relationship in the WSN. The weight vector of each neuron contains an estimate of the corresponding node coordinates.

let $d_{\text{hop}}(i, j)$ denote the *hop distance*, i.e. the minimum number of transmission required to transfer a message from a node i to a node j.

3.1.2 *Modified SOM Model.* We compute the unknown node positions by using a SOM with N neurons. Each neuron j corresponds to a sensor node and contains a weight vector $\mathbf{w_j}=[\mathrm{x}_j, y_j]^t$. *This vector, initially picked at random, will eventually contain the estimated location for the corresponding node.* The map is trained using the same algorithm described in Section 2.1, but we modify the neighborhood function to account for the spatial relationships among the sensor nodes. In particular, we use a new neighborhood function where the map distance d_{map} is replaced by d_{hop} as shown below:

$$h_{cj} = \exp\left(-\frac{d_{\text{hop}}(c, j)^2}{2\,\sigma^2}\right). \tag{4}$$

The use of d_{hop} in place of d_{map} implicitly defines a lattice of neuron with a structure that reflects the hop-count distance between each pair of sensor nodes (see Figure 5).

Having described the structure of the weights and the map, the last step involves the choice of a proper training set. To understand how to generate the training samples, we first note that since no reference points or range information are used, the SOM's results will be correct up to global translations, rotations, flipping or scaling. This is a consequence of input used and not of the SOM technique; in other words, any range-free, anchor-free scheme will generate similar results. While these ambiguities might appear as a potential complication, in reality they simplify the algorithm's implementation. In fact, since the result will be expressed in an arbitrarily coordinate system, we can train the SOM with random samples from an arbitrary distribution (e.g. $0 \leq x, y \leq 1$).

3.1.3 *Localization Algorithm.* Our algorithm is centralized, therefore each node needs to communicate the list of its radio neighbors to the unit in charge of the computation. Using this information, the hop-count distances between each pair of nodes are computed by first representing the network as a graph, and then applying the Dijkstra or Floyd algorithm. We assume that the distances are stored in a matrix D_h with elements $[d_h]_{ij} = d_{\text{hop}}(i, j)$. The matrix D_h is the only input parameter to the algorithm.

Algorithm 1 contains the pseudo-code of the localization scheme. The learning

Algorithm 1: 2D SOM-V Localization

Input: matrix D_h: hop count distances among nodes
Output: $[x_j, y_j]$ for $j = 1, \ldots, N$: node positions

% Parameter Initialization
1: $\eta_{\max} = 0.1$; $\eta_{\min} = 0.01$;
2: $\sigma_{\max} = \max_{i,j}\{D_h\}/2$; $\sigma_{\min} = 0.001$

3: **for all** nodes n **do**
4: $[x_n, y_n]^T = \textbf{random}()$
5: **end for**

% Main Loop
6: **for** $n = 1$: to N_ITER **do**
7: $\eta = \eta_{\max} - n(\eta_{\max} - \eta_{min})/(\text{N\_ITER} - 1)$
8: $\sigma = \sigma_{\max} - n(\sigma_{\max} - \sigma_{min})/(\text{N\_ITER} - 1)$

9: $(x, y) = \textbf{random}()$
10: $c = \arg\min_j \|(x, y) - (x_j, y_j)\|$
11: **for all** network nodes j **do**
12: $h_{cj} = \exp\left(-D_h(c, j)^2/2\sigma^2\right)$
13: $[x_j, y_j] \mathrel{+}= \eta\, h_{cj}([x, y] - [x_j, y_j])$
14: **end for**
15: **end for**

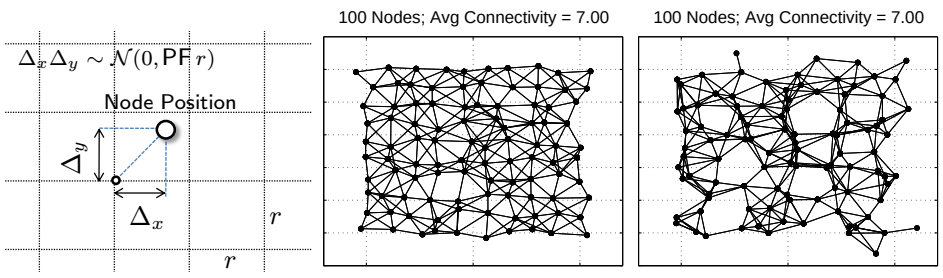
parameter $\eta(n)$ and the radius $\sigma(n)$ are scheduled using a linear function that decreases with the number of iterations (see lines 7 and 8). We refer to Algorithm 1 as SOM-V, because, as discussed in Section 4, it generates *virtual coordinates*. The SOM-A and SOM-R versions are discussed in Section 5 and Section 7.

3.2 Simulation Model

The proposed localization schemes have been validated using extensive simulations that were generated in the attempt to model realistic network configurations. Before presenting the results of our experiments, we describe the simulation model used.

3.2.1 *Placement Model.* The simulation scenarios are generated according to a *noisy grid* deployment model where the node positions correspond to the intersection points of a regular grid with rows and columns spaced by a factor r (see Figure 6a). To capture the random nature of an ad-hoc deployment, each coordinate is perturbed with samples from normal random variables: $\Delta_x, \Delta_y \sim \mathcal{N}(0, \sigma_N)$. We control the amount of noise by computing the value of σ_N as follows:

$$\sigma_N = \mathsf{PF}\, r, \tag{5}$$



(a) Noisy Grid Model (b) PF = 20% (c) PF = 50%

Fig. 6: a) The noisy grid model; b, c) two 100-node networks with different perturbation factors.

where PF is the *Perturbation Factor* parameter ($\mathsf{PF} \geq 0$) that defines the magnitude of the noise relative to the grid spacing r. Figures 6b,c show two 100 node topologies with increasing values of the parameter PF.

In Section 5.4 we also consider topologies with node positions sampled from *independent and identically distributed* (i.i.d) random variables, but we find the noisy grid model more appropriate to describe typical WSN deployments. For example, in applications such as environmental monitoring and precision agriculture some control is usually exerted to ensure an approximate uniform coverage of the monitored area. The noisy grid model also makes it easier to generate connected networks with low connectivity (e.g. 4 or 5), while in the random model, the probability of having connected networks rapidly goes to zero as we reduce the communication range [Krishnamachari et al. 2002].

3.2.2 *Connectivity Model.* We initially define which nodes are radio neighbors using an *ideal radio model*. If R denotes the maximum communication radius, then two nodes are considered "connected" if their separation distance is less than R, and "disconnected" otherwise. Although this model over-simplifies the nature of wireless communication, the use of ideal connectivity is intuitive and facilitates comparison with previously published results. A more realistic connectivity model will be considered in Section 6.

3.2.3 *Error Metric.* We evaluate the performance of the proposed schemes by computing the average localization error relative to the communication range:

$$\text{Avg. Error (R)} = \frac{1}{R} \sum_{i=1}^{N} \frac{\sqrt{(\tilde{x}_i - x_i)^2 + (\tilde{y}_i - y_i)^2}}{N}, \tag{6}$$

where N is the number of nodes, $(\tilde{x}_i, \tilde{y}_i)$ are the estimated coordinates, (x_i, y_i) are true node positions, and R is the communication range. In Section 6 we will use a different error metric to facilitate comparison with the Cramér-Rao bound.

4. LOCALIZATION WITHOUT ANCHOR NODES: VIRTUAL COORDINATES

We refer to basic version of the algorithm described in Section 3.1.3 as SOM-V. Since only connectivity information are used, SOM-V generates *Virtual Coordinates* [Moscibroda et al. 2004] that describe the relative locations of nodes, in the

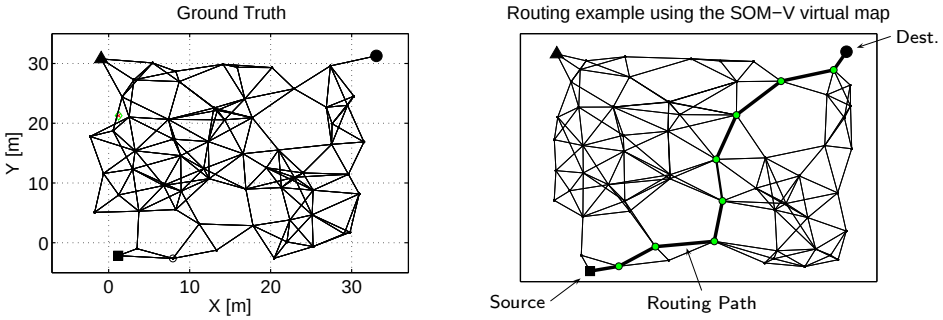


Fig. 7: Routing path discovered by a geo-routing using the coordinates produced by SOM-V (note: the two maps has been aligned using the three reference marks on the plot, but this step is not necessary for routing).

sense that nodes with similar coordinates are physically close. Virtual coordinates, which facilitate network tasks such as location-based queries and proximity-based service discovery, have found prominent application in the area of geo-routing [Karp and Kung 2000; Kuhn et al. 2003; Rao et al. 2003]. By knowing the relative node positions, these schemes achieve efficient packet delivery without the memory overhead of table-driven protocols or the latencies of on-demand approaches.

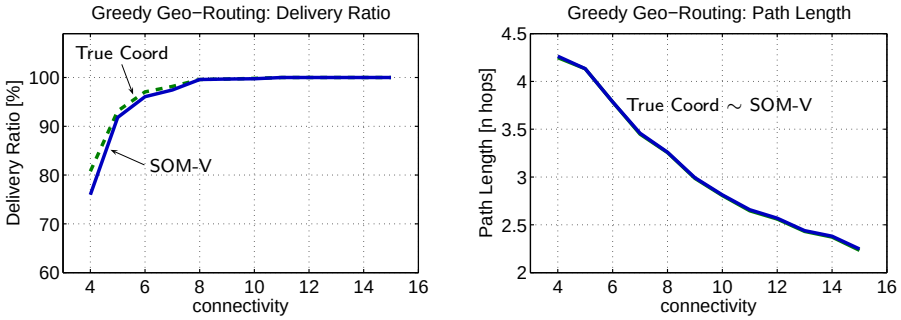
A direct comparison between virtual coordinates and the ground truth is not possible[3], therefore we evaluate the quality of the results by considering the performance of a *geo-routing* scheme that uses the virtual maps produced by SOM-V. A similar approach has been used to evaluate other range-free schemes, e.g. [Niculescu and Nath 2001] and [He et al. 2003]. We consider a *greedy* routing scheme: given a source and a destination pair, each intermediate node forwards the message to the neighboring node closest to the destination. The selection rule is:

$$next\_hop = \arg\min_{n} \|(x_n, y_n) - (x_{\text{dest}}, y_{\text{dest}})\| \,, \tag{7}$$

where (x_n, y_n) are the virtual coordinates of the neighboring nodes, $(x_{\text{dest}}, y_{\text{dest}})$ are those of the destination, and $\|\cdot\|$ denotes the usual Euclidean norm. This basic scheme simply gives up if it is unable to get closer to the destination; however, it defines an useful comparison baseline for more advanced strategies.

Figure 7a represents a 64 node network deployed in a square region with side 30 m and PF parameter equal to 25%. We show a routing example by first using SOM-V to compute a virtual map of the network, and then using (7) to route a message between two nodes in the corners of the map. Figure 7b shows the virtual map together with routing path. The length of the path is equal to the minimum hop count distances between the nodes.

[3]In the case of range-based localization algorithm that operates without anchor nodes, some quality metrics based on the inter-node distances are available (e.g. see the *Global Energy Ratio* (GER) in [Priyantha et al. 2003]). However, in the case of range-free localization, the results are not only possibly rotated or flipped, but also arbitrarily scaled; therefore error metrics bases on the inter-node distance are not applicable.



(a) (b)

Fig. 8: Average performance of a greedy geo-routing algorithm using the true coordinates and the virtual maps produced by SOM-V. The path length plots (right) are almost completely overlapping for the two cases.

In a more exhaustive simulation experiment, we generate 50 topologies similar to the one in Figure 7a with PF uniformly selected in the interval between 10% and 50%. For each topology, we use (7) to route messages between 50 pairs of randomly selected nodes. Figures 8a and 8b show the simulation results for different connectivity levels obtained by varying the communication range R. The delivery ratio of the scheme using the SOM-V coordinates is close to the value achieved when using the true node positions, and it rapidly approaches 100% as the connectivity increases. We measured no substantial differences between the lengths of the routing paths produced using SOM-V and the length of those computed using the true coordinates, in fact the two plots are almost completely overlapping in Figure 8b.

5. ABSOLUTE COORDINATES

Virtual coordinates are useful to implement efficient packet routing and other network tasks, but some WSNs require absolute positioning. For example, in a disaster relief application knowing the sensor positions is necessary to accurately pinpoint the location of an event and provide prompt assistance. To convert relative node positions into absolute coordinates, at least three non-collinear anchor points are needed for the two-dimensional case. When this information is available, the virtual maps are aligned by applying a linear transformation that resolves rotational, scaling and flipping ambiguities. This *a-posteriori* transformation can be used to align the results of any anchor-free localization technique, including SOM-V. Our basic algorithm, however, can be modified to include anchors' information in the training phase of the map. As we are about to show, this modification increases the scheme's accuracy for networks with low connectivity.

5.1 Exploiting Anchor Information: The SOM-A Scheme

The *anchored* version of the algorithm, SOM-A, is derived from the basic version by applying three modifications: **1)** Weights corresponding to anchors are initialized with the true node positions and never updated. **2)** Whenever an anchor node is elected as BMU, the training sample at current iteration is replaced with the anchor's position. **3)** The training points are sampled from a distribution whose values are

compatible with the deployment area's coordinates. The first two modifications ensure that weights corresponding to anchors remain in their position. The presence of these fixed points facilitate the map organization during the initial iterations and, assuming three or more anchors, allows SOM-A to generate maps that do not require alignment. The last modification is required to ensure that weights converge to meaningful values; differently from SOM-V, we are now working with absolute coordinates and the input samples cannot be generated from an arbitrary distribution.

In SOM-A, the sampling area is obtained by considering the rectangle enclosing the anchor locations; we assume anchors located near the perimeter of the deployment area, preferably closer to the corners. Assuming that the network contains M anchor nodes placed at locations $[x_a^{(k)}, y_a^{(k)}]$ for $k = 1, \ldots, M$, then, the training points $\mathbf{x_i} = [x_i, y_i]$ are generated by sampling an uniform distribution in the following intervals:

$$\begin{cases} x_i \in [x_a^{(\min)} - \Delta_{ax};\, x_a^{(\max)} + \Delta_{ax}] \\ y_i \in [y_a^{(\min)} - \Delta_{ay};\, y_a^{(\max)} + \Delta_{ay}], \end{cases} \tag{8}$$

where $x_a^{(\min)} = \min\{x_a^{(k)}\}, x_a^{(\max)} = \max\{x_a^{(k)}\}$ and the values $y_a^{(\min)}$ and $y_a^{(\max)}$ are computed similarly. The sampling area is expanded in each direction by factor Δ_{ax} and Δ_{ay} to compensate for *border effects* that are notorious in the use of SOM technique. Border effects arise because boundary neurons have fewer neighbors than inner neurons; as a result, the weights along the perimeter of the map are slightly contracted toward the center. In our case, we compensate for this effect by expanding the sampling area. We experimentally determined an effective value for Δ_{ax} and Δ_{ax} to be equal to

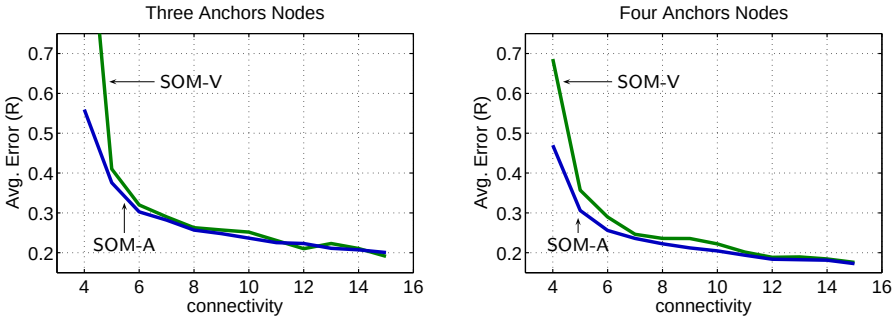
$$\Delta_{ax} = \frac{x_a^{(\max)} - x_a^{(\min)}}{\sqrt{N+M}-1} \text{ and } \Delta_{ay} = \frac{y_a^{(\max)} - y_a^{(\min)}}{\sqrt{N+M}-1} \text{ and} \tag{9}$$

where $N + M$ is the total number of network nodes (anchors and non-anchors).

5.2 Simulation Results: Comparison Between SOM-A and SOM-V

Figure 9 reports the average localization error for the same set of 50 networks used in the previous section, assuming the presence of three and four anchors on the corners of the map. The results of SOM-V, which are obtained using the *a-posteriori* transformation described earlier, are compared against those of SOM-A. The plots show that the anchored version is more effective in localizing network with low connectivity. For networks with connectivity equal to four, SOM-A reduces the error by 52% in the case of three anchor, and by 32% for the case of four anchors.

Although comparison with other schemes is deferred to Section 5.4, the SOM-A's results in Figure 9b show a localization error as low as 0.3 R for networks with average connectivity equal to five and using only 6.25% of anchors nodes (4 anchors out of 64 nodes). These figures suggest SOM-A as a suitable approach for localization in low-cost deployments with low connectivity and a small percentage of anchor nodes. These networks are likely to be exploited for applications such as environmental monitoring and precision agriculture, where slowly varying signals such as temperature and humidity are monitored over large areas. In these appli-



(a) (b)

Fig. 9: Average localization error for a set of 50 random topologies with 64 nodes deployed in a $30\,m \times 30\,m$ square region.

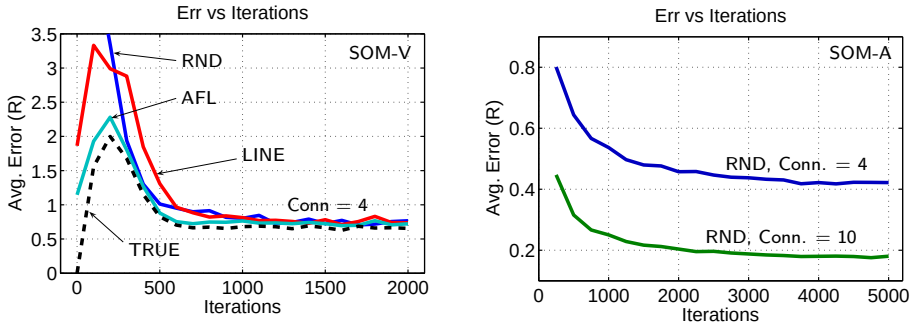
cations, nodes are often placed in sparse configurations to reduce the installation and maintenance costs.

5.3 Weight Initialization and Convergence

The results in Figure 9 were obtained using random initialization of the weights and 2000 training samples. Existing SOM literature (e.g. [Kohonen 2001]) shows, however, that different initialization strategies influence both the convergence speed and the topological accuracy of the solution. Understanding the effect of the initial weights and number of iterations on the final results is important because the scheme may execute on nodes with limited computational resources.

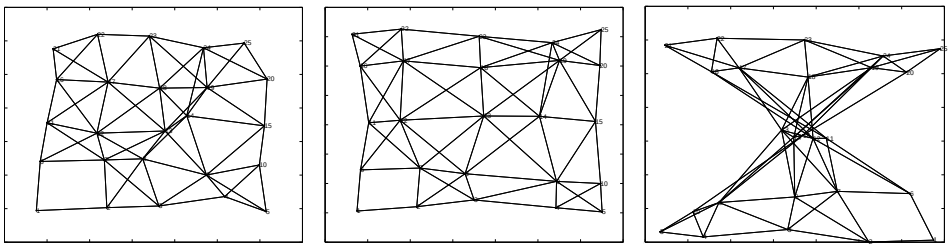
We have repeated the previous simulations by varying the number of iterations and using different initialization strategies. In addition to random initialization, we consider the AFL scheme proposed by Priyantha et al. [2003], who used this approach to generate *fold-free* initial configurations for a spring-mass based algorithm, and a simple scheme that initializes the weights by aligning them along a line. In our previous work [Giorgetti et al. 2007], we found this heuristic to be effective in reducing the occurrence of maps with large topological errors. We finally consider a best-case scenario by initializing the weights with the true node positions.

Figure 10a shows the SOM-V's error as a function of the number of iterations for the case of network with average connectivity equal to four. We consider this case because we found that the differences between alternative initialization schemes are most noticeable for topologies with low connectivity. Even in this case, however, the error plots converge to a similar value as the number of iterations increases over 1000. The weak correlation between the final error and the initial weights is a consequence of training the map with a large neighborhood function. In our solution, the initial standard deviation for the Gaussian kernel is equal to the radius of the network: $\sigma_{\max} = \max(D_h)/2$ (see Algorithm 1). Such large value causes strong interactions among the neurons; therefore, during the initial iterations, the weight vectors will assume a similar value close to the centroid of the input distribution, regardless of the initial positions. In the case where neurons were already partially ordered, the convergence speed of the map could be improved by using a smaller value for $\sigma_{\max}$ to preserve some of the initial information. However, we do not pur-



(a) SOM-V, various initialization schemes. (b) SOM-A, random initialization

Fig. 10: Average localization error as a function of the number of iterations used in training the map. The results were generated using the same set of networks discussed in the previous section.



(a) Ground Truth (b) Convergence OK (c) Converge Problem

Fig. 11: Localization example. If the neighborhood function's initial radius is not wide enough, SOM will occasionally produce incorrect results (case c). This problem is avoided by using the values in Algorithm 1.

sue this strategy because using a narrower neighborhood function will occasionally result in maps that are only partially ordered (see Figure 11).

Figure 10b shows the SOM-A's error for the same set of simulations used in the previous case. When the node positions are computed using SOM-A, the differences between alternative initialization strategies become negligible even for a low number of iterations. Given the minimal differences, we only report the error obtained using random initialization for networks with low and medium connectivity. The error decreases similarly in both cases and stabilizes around 3000 iteration for connectivity equal to four, and about 2000 iterations when the connectivity is equal to ten. Since the execution time of the algorithm depends on the number of training samples (i.e. the number of iterations), adjusting this parameter provides a mean to optimize the trade-off between accuracy and resources spent (see Section 9).

5.4 Comparison With Other Schemes

We compare SOM-A with two popular range-free schemes, DV-HOP and *Multi Dimensional Scaling* (MDS).

In the DV-HOP scheme proposed by Niculescu and Nath [2001; 2003], each node computes its hop-count distance from the anchors present in the network. The

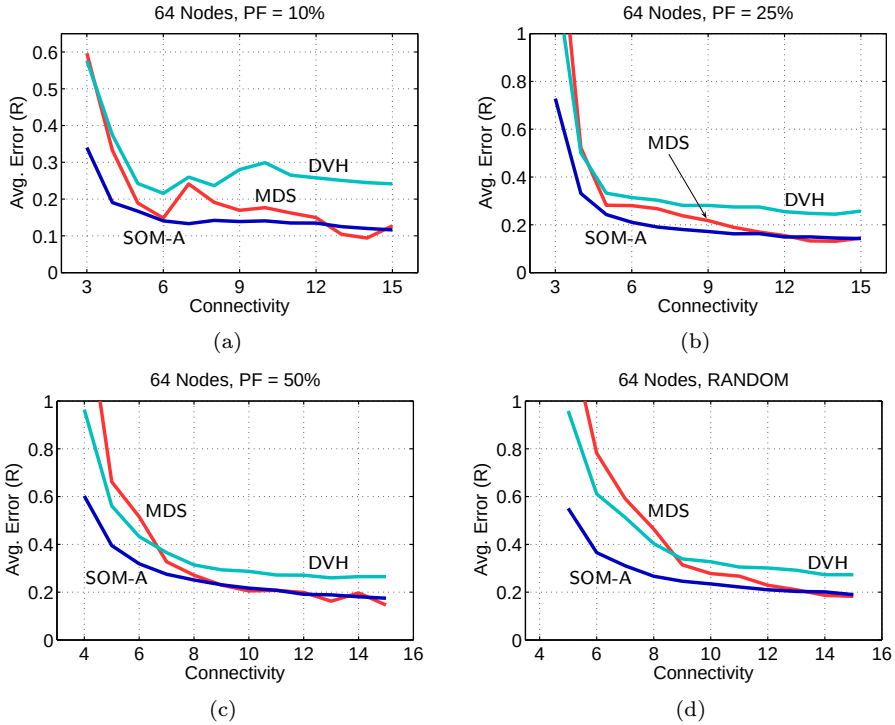


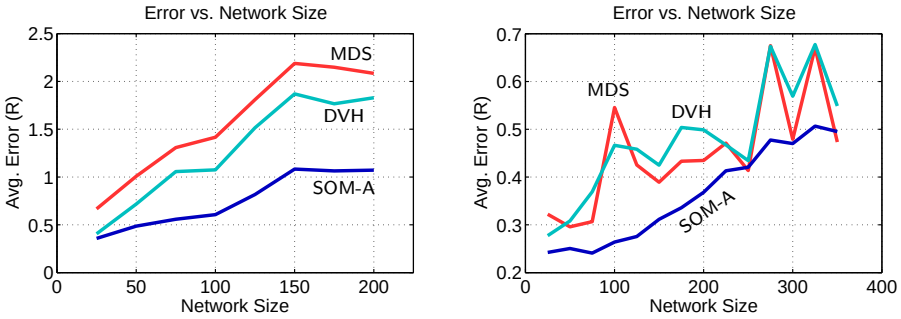
Fig. 12: Average error achieved by SOM-A, DV-HOP, and MDS in localizing sets of 50 networks with increasing perturbation factors. All networks have four anchors.

hop-count values are then converted into range estimates and used to determine the node positions via multi-lateration. An approach similar to DV-HOP has been proposed by Nagpal et al. [2003]; comparative studies between DV-HOP and other schemes have been reported by He et al. [2003].

The other scheme used for comparison is MDS, a projection technique that maps points from a high-dimensional set to a low-dimensional space. Traditionally used in behavioral, econometric and social sciences, MDS has also been applied to the localization problem in sensor networks. Initially proposed by Shang et al. [2003], the method has been successively extended to work in a distributed fashion [Shang and Ruml 2004; Ji and Zha 2004; Costa et al. 2005; Vivekanandan and Wong 2006]. In our tests, we evaluate a centralized version that uses the hop-count values as a distance measure between pair of nodes (Isomap [Tenenbaum et al. 2000]).

Figure 12 shows the localization results for four sets of 50 random topologies with 64 nodes deployed in a square region with side equal to $30\,m$. Three of the four sets were generated using the noisy grid model described in Section 3.2.1 with $\mathsf{PF} = \{10\%, 25\%, 50\%\}$, while the last set contains networks with node positions sampled from i.i.d. random variables. We note that networks with PF greater than 50% are qualitatively similar to random deployments.

The plots in Figure 12 show that the results generated by SOM-A and DV-HOP have a similar trend, but SOM-A consistently produces a lower error. The



(a) Connectivity = 5 (b) Connectivity = 10

Fig. 13: Average error achieved by SOM, DV-HOP, and MDS in localizing sets networks with increasing number of nodes and four anchors.

SOM-A's results, which were are obtained using random initialization and 5000 training samples, are on average 27% to 45% more accurate than DV-HOP. The differences are more marked for networks with perturbation factor equal to 10%.

In comparing the performance of SOM-A with MDS, the results depend on the network connectivity. The difference are negligible for networks with connectivity greater than ten, but SOM-A significantly outperforms MDS for sparse networks. When the connectivity is equal to four or five, the SOM-A's error is approximately between 60% and 40% lower than the MDS's error. In general, when the connectivity is low, the hop-count distances are poor approximations of the true node distances, especially for nodes that are several hops away. Since MDS equally weights all the available distances, including those with large hop values, the error in sparse networks are usually large. SOM-A does not exhibit this drawback because nodes that are several hops away have a weak effect on each other's positions.

In a second set of experiments, we localize sets of 25 random networks with increasing numbers of nodes and fixed connectivity values equal to five and ten (see Figure 13). In the first case (connectivity = 5), we simulated random networks with up to 200 nodes; given the low connectivity value, above this size it becomes progressively more difficult to generate connected networks. In the second case (connectivity = 10), we generated networks with up to 350 nodes. The SOM-A technique ensures a localization error around 1.0 R up to 200 nodes and connectivity equal to five. If the connectivity is increased to ten, the error of SOM-A maintains below 0.5 R for networks up to 350 nodes. The results obtained for the other two schemes are consistent with those published by Niculescu and Nath [2001; 2003] for DV-HOP, and those reported by Shang et al [2003] for MDS. The error generated by both schemes is significantly higher than the SOM-A's error, especially for networks with low connectivity.

6. LOCALIZATION IN NOISY ENVIRONMENTS

The *ideal radio model* used in previous sections provides an intuitive abstraction useful in simulation studies, but it does not adequately capture the random nature of wireless communication. Multi-path fading due to reflection, diffraction and scattering of the RF signal causes variations in the received power and ultimately

affects the capacity of the recipient to correctly decode a radio message. A more realistic connectivity model that takes into account such variability has been proposed by Patwari and Hero III [2003]. In this section, we adopt the same model to simulate localization in "less than ideal" radio environments.

6.1 Log-Normal Shadowing Model and Threshold-based Connectivity

We assume that nodes collect *Received Signal Strength* (RSS) values by exchanging several radio messages and that the measured values follow the *log-normal shadowing model*, a propagation model widely used in link budget analysis [Rappaport 1996]. Let P_{ij} denote the average of the RSS values between node i and node j. If the received power is measured in dB or dBm, then P_{ij} is modeled as a random variable with normal distribution:

$$P_{ij} \sim \mathcal{N}(\bar{P}_{ij}, \sigma_{\mathrm{dB}}) \tag{10}$$

$$\bar{P}_{ij} = P_0 + 10\, n_{\mathrm{p}} \log_{10} \left(\frac{d_0}{d_{ij}} \right). \tag{11}$$

In the expression above, P_0 is the received power at a distance d_0[4], n_{p} is the *path loss* exponent and d_{ij} is the distance between nodes i and j. The standard deviation σ_{dB} models the variability that is usually measured between different pairs of nodes. Obstructions in the path between the nodes and reflections of the signal on nearby obstacles can produce significant differences in the average received power for pairs of nodes with the same separation distance, but installed in different locations.

According to the approach proposed by Patwari and Hero III [2003], we determine if two nodes are neighbors by comparing the average signal strength against a fixed threshold P_{th}:

$$c_{ij} = \begin{cases} 0 & \text{if } P_{ij} < P_{th} \quad \text{(nodes disconnected)} \\ 1 & \text{if } P_{ij} \geq P_{th} \quad \text{(nodes connected).} \end{cases} \tag{12}$$

This model is more realistic than assuming ideal connectivity, and, in addition, it also provides the basis for a more formal analysis of the results. Using the *measurement model* defined by equations (10),(11) and (12), localization can be casted as a parameter estimation problem and the results can be evaluated using *Cramér-Rao Bound* (CRB) analysis. The CRB defines the minimum variance achievable by any *unbiased estimator*[5] that uses the measurements c_{ij} to compute the node positions. Notably, the CRB provides a bound that is independent of the particular algorithm used, but it only depends on the network topology and the parameters of the measurement model. Several authors have used CRB analysis to evaluate the performance of their schemes, mostly in the range-base case; see, for example, [Patwari et al. 2003; Savvides et al. 2003; Moses et al. 2003]. Details on how to derive the CRB for the case of connectivity measurements are reported by Patwari and Hero III [2003].

[4]The reference value P_0 at distance d_0 is either measured or computed using the Friis' free space equation. For low-power radios, P_0 is usually measured at $d_0 = 1\,m$.

[5]If $\hat{\theta}$ is an estimate of the unknown parameter θ obtained using T, i.e. $\hat{\theta} = T(C)$, then the estimator T is unbiased if $E\{\hat{\theta}\} = \theta$.

6.2 Simulation Results and Comparison with the CRB

We use the CRB as a comparison baseline for the error of SOM-A, MDS and DV-HOP in localizing the network in Figure 14a. The average value P_{ij} between each pair of nodes were sampled according to the distribution (10) in fifty different trials to simulate the deployment of the same topology in different environments. For each scheme, we computed the average *Root Mean Square* (RMS) error defined as:

$$\text{RMS Err} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(x_i-\hat{x}_i^{(k)}\right)^2+\left(y_i-\hat{y}_i^{(k)}\right)^2} \tag{13}$$

where K is the total number of repetitions (fifty in our case), N is the number of nodes (excluded the anchors), (x_i, y_i) are the true node coordinates, and $(\hat{x}_i^{(k)}, \hat{y}_i^{(k)})$ are the coordinate computed using one of the localization algorithm in the k^{th} trial. In our simulation, the RMS Error is compared against the average value of the node positions' standard deviation computed using the CRB.

Before discussing the localization results, we note that the presence of the shadowing variance σ_{dB}^2 in (10) can be regarded as a source of noise that disrupts the dependence of P_{ij} from the distance d_{ij}. In particular, the estimation error for the node positions using RSS or connectivity measurements increases with the ratio $\sigma_{\text{dB}}/n_{\text{p}}$ [Patwari et al. 2003; Patwari and Hero III 2003]. For high values of the ratio $\sigma_{\text{dB}}/n_{\text{p}}$, the RSS data are only loosely related to the distance, and the information content associated with these measurements becomes marginal. Since RSS data are used in (12) to define which nodes are neighbors, increasing the ratio $\sigma_{\text{dB}}/n_{\text{p}}$ also cause the localization error to increase when connectivity measurements are used.

Typical values for n_{p} are comprised between 2 and 4, while σ_{dB} usually varies between 3 and 12 dBm [Rappaport 1996]. We cover different scenarios by considering three values for the ratio $\sigma_{\text{dB}}/n_{\text{p}}$: low noise ($\sigma_{\text{dB}}/n_{\text{p}} = 3/4\,\text{dBm}$), medium noise ($\sigma_{\text{dB}}/n_{\text{p}} = 6/3\,\text{dBm}$) and high noise ($\sigma_{\text{dB}}/n_{\text{p}} = 9/2\,\text{dBm}$). As shown in Section 7.3, the values chosen are consistent with the values measured in real WSN testbeds.

The results are reported in Figure 14 for different connectivity values that are achieved by changing the threshold P_{th} in (12). When the noise due to shadowing effects is low (see Figure 14b), the RSS values and the connectivity information are strongly correlated with distance, and the results are qualitatively similar to those discussed for the ideal radio model. In this case, for low connectivity values, SOM-A outperforms the other schemes, achieving a RMS error close to the bound. For larger connectivity values, MDS produces more accurate results with an error close to the CRB for connectivity comprised between 25 and 45. Again, the error of DV-HOP increases similarly to that of SOM, but its performance is further removed from the theoretical bound.

When the ratio between $\sigma_{\text{dB}}/n_{\text{p}}$ increases (see Figures 14c,d), the noise due to shadowing effects corrupts the connectivity information and cause the localization error to increase (including the CRB). The effects are more severe for MDS that performs worse than the other two solutions for high noise values (see Figures 14d). The error generated by SOM, which is inherently a stochastic scheme, and therefore less sensitive to measurements errors, degrades more gracefully as the noise increases.

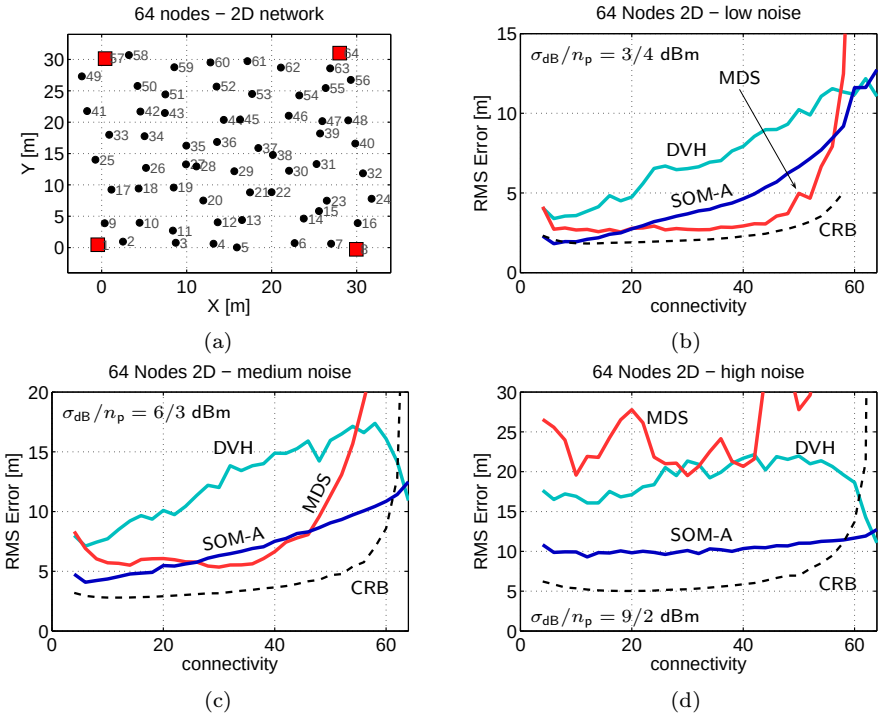


Fig. 14: a) 2D sample topology (red squares are anchor nodes); b,c,d) average RMS error achieved by SOM-A, MDS and DV-HOP for different values of the propagation model.

6.3 Localization in 1D and 3D spaces

In addition to the canonical application of localization in two dimensions, we also evaluated the three schemes when nodes are placed in 1D and 3D spaces. WSNs with linear configurations of nodes find application, for example, in traffic monitoring along highways and perimeter control, while 3D deployments are found in asset tracking application for large warehouse or when instrumenting multistory buildings for ubiquitous computing.

All the three localization algorithms can be easily modified to work in dimension different from two. In particular, the only modifications required by the SOM schemes are the use of weights with a different dimensionality and a corresponding changes in the sampling space (points are sampled from a line for localization in 1D and from a cube for 3D localization).

We report the test topologies used and the results in Figure 15 for the case of medium noise ($\sigma_{\mathrm{dB}}/n_{\mathrm{p}} = 6/3\,\mathrm{dBm}$). The result for other noise levels are qualitatively similar to the 2D case. The only exception are the result of the MDS scheme, which produced large localization error in all the 1D configuration tested. In contrast, both SOM and DV-HOP were capable of achieving an error close to the bound for low connectivity values.

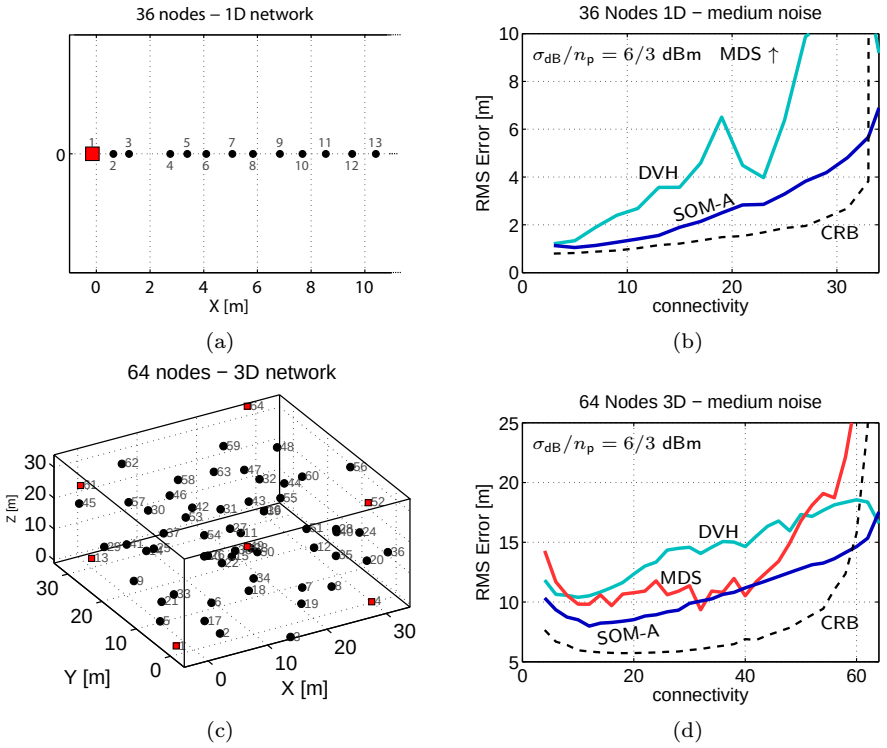


Fig. 15: a,c) 1D and 3D sample topologies (red squares are anchor nodes); b,c,d) average RMS error achieved by SOM-A, MDS, DV-HOP for different noise values and comparison with the CRB value.

7. LOCALIZATION IN DENSE NETWORKS

Both simulation results and CRB analysis show a large localization error when the connectivity reaches values close to the network size (see Figures 14 and 15). When most of the nodes are in the radio range of each other, connectivity data are of scarce utility in determining the node positions. In particular, in the extreme case of a fully connected network, since all the nodes have the same neighbors, they will be associated to the same, or a similar position. This drawback precludes application of range-free approaches to WSNs with densely deployed nodes, which are often found in indoor applications or smart spaces. In this section we introduce a new variant (SOM-R) that achieves low localization error for highly connected networks without sacrificing the original accuracy in localizing sparse networks.

7.1 The SOM-R algorithm

The negative effects of large connectivity values on the SOM schemes is easily understood by recalling the update rule (2) discussed in Section 2.1. When the network is highly connected, a large number of neurons will be within the same distance from the BMU, therefore the weight updates will be similar for many nodes and the map will not be able to accurately represent the input distribution.

In SOM-R we avoid the shortcomings of range-free localization in dense networks

by using the RSS values to redefine the hop count distance d_{hop}. The idea is to augment the proximity information by sorting the one-hop neighbors on the basis of their received power. In particular, we consider the attenuation of the RF signal between a pair of nodes i and j expressed by the *path loss* P_{L}:

$$P_{\text{L}}(i,j) = P_0 - P_{ij}, \tag{14}$$

where P_0 and P_{ij} have the same meaning defined in (10) and (11). Similarly to P_{ij}, P_{L} is also a random variable with normal distribution; the expected value for P_{L}, is approximately zero for nodes whose separation distance is d_0, and it increases for nodes that are far apart. Here we assume that d_0 is small compared to the separation distances between the network nodes, therefore $P_{\text{L}}(i,j) \geq 0, \forall i,j$. If the network is implemented using transceivers with a typical *sensitivity* P_s[6], we expect the path loss to increase up to a maximum value $P_{\text{L-MAX}} = P_0 - P_s$. When such value is reached, the RF power at the receiver will equal P_s and a further increment in P_{L} will cause the communication to fail with a high probability. Based on these considerations, we define a new neighborhood function:

$$h_{cj}^{(\text{PL})} = \exp\left(-\frac{d_{\text{hop}}^{(\text{PL})}(c,j)^2}{2\,\sigma^2}\right). \tag{15}$$

In the expression above, $d_{\text{hop}}^{(\text{PL})}(c,j)$ uses the path-loss values to measure the distance between the BMU at index c and the a generic node/neurons at index j:

$$d_{\text{hop}}^{(\text{PL})}(c,j) = \begin{cases} P_{\text{L}}(c,j)/P_{\text{L-MAX}} & \text{if} \quad d_{\text{hop}}(c,j) = 1 \\ d_{\text{hop}}(c,j) & \text{if} \quad d_{\text{hop}}(c,j) \neq 1. \end{cases} \tag{16}$$

The modified hop-count distance is unchanged for nodes that are not in the radio range of the BMU ($d_{\text{hop}}(c,j) \neq 1$), but it has increased resolution for one-hop neighbors, which are treated differently depending on their path loss value. A $P_{\text{L}}(c,j)$ value close to zero will result in a small hop-count value ($d_{\text{hop}}^{(\text{PL})} \approx 0$), which in turn will cause a strong interaction ($h_{cj}^{(\text{PL})} \approx 1$) between the BMU and node j. On the other hand, when $P_{\text{L}}(i,j)$ is close to $P_{\text{L-MAX}}$, the two nodes will be treated as regular one-hop neighbors.

As mentioned in Section 2.1.1, the shape of the neighborhood function, and consequently the choice of d_{hop}, is not a critical factor in the SOM learning algorithm. In using the P_{L} values as shown above, our intent is not to accurately model the distance between nodes, but simply to provide a mechanism to differentiate between neurons that otherwise would be at the same distance from the BMU. Figure 16a shows an example of a neighborhood function based on $d_{\text{hop}}^{(\text{PL})}$.

The use of the $d_{\text{hop}}^{(\text{PL})}$ in place of the regular hop distance is the major difference between SOM-R and the other versions of the scheme. The modified neighborhood function (15) can be used both with SOM-V and SOM-A, but in the rest of our work we only evaluate the anchored version. We finally note that when the modified

[6]The radio sensitivity is the minimum signal power that the transceiver is able to demodulate with high probability. For example, transceivers compliant with the IEEE 802.15.4 should be able to ensure a Packet Error Rate (PER) less than 1% for signal with power equal to -86dBm or above.

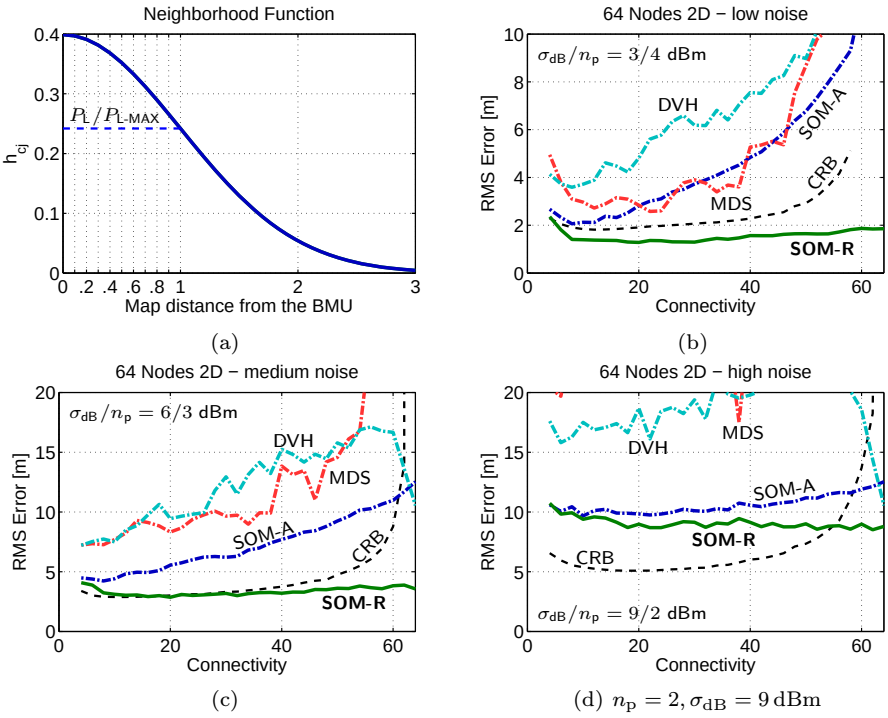


Fig. 16: Average RMS error achieved by SOM-A, SOM-R, MDS and DVH.

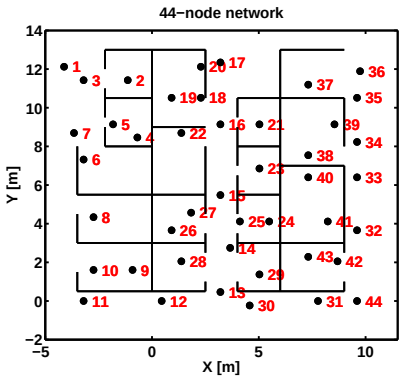
map distance is used, it is beneficial to train the map with a larger learning factor, therefore we use $\eta_{\max} = 0.5$ instead than $\eta_{\max} = 0.1$.

7.2 Simulation Results

Similarly to the previous section, we compute the RMS error over 50 localization experiments on a 64-node network with four anchor nodes. Once again we account for different radio environments by considering three noise levels previously used, and we vary the connectivity level by changing the threshold used in (12). In this case, the value P_{th} is used in place of the sensitivity P_s in the term $P_{\text{L-MAX}}$ in (15). In general, if a threshold is not used and the value of P_s is unknown, the maximum path loss value measured within the network can be used in place of $P_{\text{L-MAX}}$.

Figure 16b,c,d show the simulation results. SOM-R produces remarkable improvements in accuracy, especially for higher values of network connectivity. In the cases of low and medium noise, and for connectivity equal to 60, the RMS error is respectively 85% and 65% lower than the value produced by SOM-A. In the case of high noise ($\sigma_{\text{dB}}/n_{\text{p}} = 9/2$ dBm) the RMS's error is about 25% lower than that of SOM-A. Notably, SOM-R maintains the accuracy of SOM-A for sparse network, and it produces meaningful localization results even in fully connected networks, achieving results that are practically independent from the network connectivity.

Note that in Figure 16 the SOM-R's error is sometimes lower than the CRB. This is not in contradiction with the definition of the CRB, because SOM-R uses



Scheme	RMSE	Reference
MLE	2.18 m	[Patwari et al. 2003]
MDS	4.30 m	[Costa et al. 2005]
dwMDS	2.48 m	[Costa et al. 2005]
SOM-R	**2.17 m**	this paper

Notes

—Nodes 3,10,35 and 44 are used as anchors in the localization process.

—The MDS schemes evaluated by Costa et al. uses range estimates from RSS values.

Fig. 17: The 44-node network derived from in-field RSS measurements [Patwari et al. 2003] and published localization result for the same network.

not only connectivity constraints, but also the RSS information. Using raw RSS data, however, is different from other approaches that use signal strength to estimate the inter-node distances. To produce such range estimates, for example using a *Maximum Likelihood Estimator* (MLE) [Patwari et al. 2003], knowledge of the propagation model parameters n_{p}, σ_{dB} is required[7], which in turns involves collection of a large set of controlled measurements and adds to the costs of the localization service implemented. The SOM-R scheme can localize nodes deployment in environments for which the parameters of the propagation model are unknown.

7.3 Localization Using RSS Data From Real Deployments

We complete the evaluation of SOM-R and the other SOM variants by presenting localization results obtained from data collected in real WSN deployments.

7.3.1 *Test Case 1: 44 Node Network, Medium Noise.* We use the SOM-R scheme to localize the nodes in Figure 17. The data for this network has been collected by measuring the average RSS between pair of nodes in the cubicles of an office space. The measurements are described by Patwari et al. [2003], and the data is freely available on the web[8]. The nodes in this network are all in the radio range of each other and the estimated parameters for propagation model are $\sigma_{\mathrm{dB}} = 3.91$dBm, and $n_{\mathrm{p}} = 2.3$. The ratio $\sigma_{\mathrm{dB}}/n_{\mathrm{p}}$ is equal to 1.7 dBm, close to the value we used to simulate networks with medium noise ($\sigma_{\mathrm{dB}}/n_{\mathrm{p}} = 6/3$ dBm).

The SOM-R scheme is executed with 5000 samples from a uniform distribution computed using (8). We repeat the localization experiment fifty times, changing the seed of the random number generator each time. The average RMS error achieved by the SOM-R is equal to $2.167\,m$, with the best and worst localization attempts that produced an error of $1.875\,m$ and $2.518\,m$ respectively. By comparing the SOM-R's error with previously published results (see Figure 17), we note that

[7]The basic MLE estimator only requires knowledge of the the parameter n_{p} (in addition to P_0 and d_0). To compute an unbiased version of the same estimator, knowledge of the parameter σ_{dB} is also necessary [Patwari et al. 2003].

[8]WSN Localization Measurement Repository: `http://www.eecs.umich.edu/~hero/localize`.

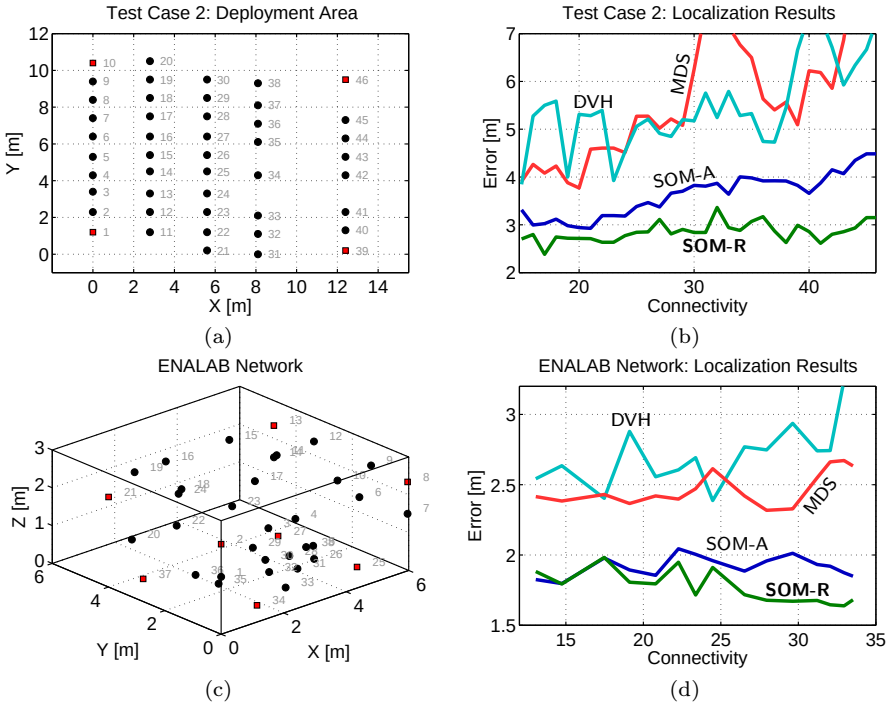


Fig. 18: Node deployments and localization results for test cases 2 and 3. Red squares represent the anchor nodes used.

SOM-R achieves performance similar to those reported by Patwari et al. [2003] for a centralize MLE estimation scheme. We remark that SOM-R's results are obtained without knowledge of the the parameters n_{p} and σ_{dB}.

7.3.2 *Test Case 2: 46 Node Network - High Noise.* In the second test case we use data from a 46 node network deployed in an indoor space measuring approximately 14×10 m. The nodes use a 2.4 GHz transceiver and are arranged in a grid as shown in Figure 18a. Some grid positions are missing due to node malfunctioning at the moment of the test. The RSS data were collected by exchanging 100 messages between each pair of nodes and computing the average of the data collected. The deployment area, an empty office space with some metallic fixtures, was relatively uncluttered, but we measured a significant level of variability in the RSS data, presumably due to multi-path reflection from the metallic walls on the perimeter of the area and due to different antenna orientation. In fact, the nodes for the test use an integrated antenna and were randomly oriented. Using the measured data, we estimated the propagation model parameters to be equal to $\sigma_{\mathrm{dB}} = 8.13$ dBm and n_{p}=2.74, resulting in a ratio close to the one we used to simulate noisy networks. In absence of previously published results, we evaluate the performance of SOM-A and SOM-R by comparing the localization error against those of MDS and DVH. Figure 18b shows the localization results for different connectivity levels that were obtained by varying the threshold P_{th}. The results are qualitatively similar to those obtained in simulating networks with high ratio $\sigma_{\mathrm{dB}}/n_{\mathrm{p}}$.

7.3.3 *Test Case 3: 38 Node Network in a 3D Space - High Noise.* In the last test case we use RSS measurements from a 38 node network deployed in an indoor 3D space (see Figure 18c). The data is freely available on the ENALAB web site[9] and the measurements are discussed in detail in [Lymberopoulos et al. 2006]. Similarly to the previous case, Lymberopoulos et al. found different antenna orientations and multi-path to be source of significant variability in the RSS data, which have low correlation with the distance. Figure 18d shows the localization error for the four scheme considered. Again, we note that error of SOM-A and SOM-R is significantly lower than the error of DV-HOP and MDS.

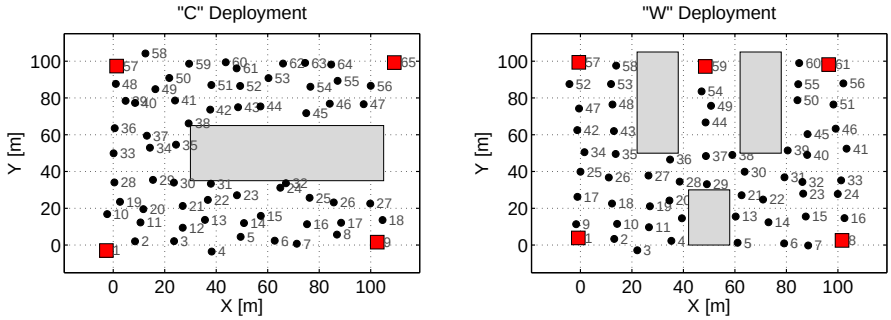
8. LOCALIZATION IN ANISOTROPIC DEPLOYMENTS

Anisotropic layouts result from deploying sensors in regions with obstacles (e.g. tall buildings), or when localized node failures lead to "holes" in otherwise isotropic topologies. It is known that localization in anisotropic networks is challenging for schemes that use the hop count values as an approximation of the true node distance (e.g. MDS and DV-HOPS). In fact, while this approach works well when the path connecting any two nodes lies approximately on a straight line, it generates large errors in presence of obstacles because any two nodes can be physically close even if their hop distance is large. The large error in the case of anisotropic networks has motivated alternative approaches. For example, some schemes use MDS to compute small local maps that are then stitched together into a global map [Shang and Ruml 2004; Ji and Zha 2004]. Although this approach yields to an interesting distributed scheme, the process of map stitching increases the complexity of the solution and is susceptible to large errors when the connectivity is low. If some components of the network are not rigidly connected, the sub-maps may get stitched together with the wrong relative orientation [Moore et al. 2004].

Here we argue that it would be useful to have a scheme capable of localizing irregular networks without having to partition the map and encumber the complexity of map stitching. The SOM schemes is potentially well-suited for this task because its learning algorithm is designed to mainly exploit the interaction of nodes within a short hop distance, while nodes that are several hops away have a weak interaction and do not directly influence each other.

We evaluate the performance of the proposed schemes by generating simulation scenarios with few large obstacles blocking the communication between nodes. Two sample topologies are shown in Figures 19a and 19b. We will refer to these two simulation scenarios as "C" and "W" deployments. For each topology, we generate 50 networks using the same noisy grid model described in Section 3.2.1, with the only difference that nodes were not allowed in correspondence of the obstacles. The connectivity information were obtained by first sampling pairs of RSS values using the shadowing model of Section 6 with parameters $n_{\mathrm{p}} = 4, \sigma_{\mathrm{dB}} = 3\mathrm{dBm}$, and then selecting a threshold as in (12). We use two different threshold values, resulting in half the networks having connectivity around 6.5 and the other half above 12. For each network, localization was repeated 25 times using different realization of the RSS values; the results of these repetitions were used to compute the RMS error and compare it with the CRB.

[9] `http://www.eng.yale.edu/enalab/XYZ/data_set_1.htm`



(a) (b)

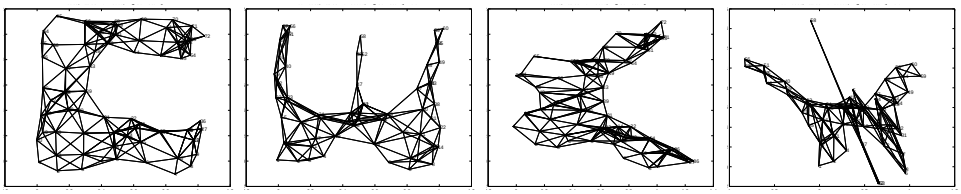
Fig. 19: Sample anisotropic topologies. Red squares are the anchor nodes.

	"C" conn = 6.68		"C" conn = 12.49		"W" conn = 6.69		"W" conn = 12.92	
Scheme	**Err(R)**	**RMS**	**Err(R)**	**RMS**	**Err(R)**	**RMS**	**Err(R)**	**RMS**
MDS	1.44	37.0 m	1.12	45.7 m	1.15	28.8 m	0.69	27.2 m
DVH	0.86	20.3 m	0.63	22.1 m	0.74	18.7 m	0.53	21.0 m
SOM-A	0.31	7.7 m	0.27	9.5 m	0.31	7.9 m	0.24	9.9 m
SOM-R	0.29	7.3 m	0.21	7.8 m	0.28	7.3 m	0.18	7.4 m
CRB		**6.8 m**		**7.0 m**		**7.8 m**		**6.5 m**

Table III: Localization Results in anisotropic networks.

Table III show the simulation results; in addition to the average RMS error, we also report the average localization error relative to the communication range R[10]. SOM-A's and SOM-R's results were computed using the same training distribution described in Section 5.1 (i.e. without knowledge of the obstacles' presence.)

The results shows that SOM-A achieves an average error comparable to the case of uniform networks. On average, the SOM-A's error is 75% and 60% lower than the error of MDS and DVH respectively. The errors for MDS and DVH on the "C" topology are comparable with previously published results reported by Vivekanandan and Wong [2006] and Niculescu and Nath [2001] for the same type of network. SOM-R generates an additional 15% error reduction with respect to SOM-A.



(a) SOM-A: "C" net (b) SOM-A: "W" net (c) MDS: "C" net (d) MDS: "W" net

Fig. 20: Sample results for anisotropic layouts: the SOM-A algorithm reduces the average localization error of 75% with respect to MDS.

[10]The maximum range was computed using the parameter of the shadowing model and the threshold, so it should be intended in the sense of *expected* maximum communication range.

9. COMPUTATIONAL COMPLEXITY

Although all the three variants proposed are centralized localization schemes, we show that the low communication and computation requirements make them suitable for sensor networks were nodes have limited resources.

The SOM algorithms operate on the basis of connectivity information, therefore each sensor needs to communicate the set of its radio neighbors to the unit in charge of the computation. Assuming that node IDs are coded using two bytes (up to 65536 nodes), the information can be transmitted using a fairly small size radio messages. For example, in a network where the average connectivity is 7, only 14 bytes need to be transmitted by each node, and the total traffic can be reduced by means of in-network data aggregation techniques.

Having received the neighbor sets, the data is used to generate the adjacency matrix of the undirected network graph requiring $[N(N-1)/2]/8$ bytes, and then to compute the D_h matrix with the hop count distances between nodes. The solution is obtained by repeating N executions of the popular Dijkstra's algorithm or using the Floyd's scheme. The complexity is $O(N^3)$ in both cases, while the table needs enough storage space for $N(N-1)/2$ elements. The memory requirements for this table can be reduced by taking into account the maximum hop count distance between any two nodes (i.e. the network diameter). Our simulations show, for example, that in 100 node networks with connectivity equal to six the diameter is usually lower than 16. Using 4 bits to code the hop-count distances, the size of the table is reduced to $N(N-1)/4$ bytes of memory. Note that, even if some hop distances were larger than 16, replacing the actual value with the upper limit does not have a noticeable impact on the algorithm because the interactions between units far from each other are weak. Finally, we need to reserve the memory space to store the node coordinates (i.e. the SOM weights). Assuming quantized values represented with 2 bytes, the total occupation is $4N$ bytes.

As for the computational complexity of our approach, the iterative solution allows a trade-off between accuracy and execution time (see Section 5.3). Each iteration requires N comparisons to compute the BMU, and the application of the update rule (2) to the map weights. Considering that the radius of the neighborhood function shrinks from a value initially equal to the network radius and then goes to zero, the average number of weight updates is approximately $N/2$. We note that, as the width σ of the neighborhood function shrinks, the running time of our solution could be further improved by only applying the update rule to those weights that are close to the BMU.

It takes about 0.3 seconds to localize a 100 node network by executing 5000 iterations of MATLAB code on a PC with a 2.66 GHz, however, we also implemented a TinyOS[11] version to test the scheme on sensor nodes. We execute the code on TelosB [Polastre et al. 2005], a popular COTS sensor node with a 16-bit RISC microcontroller featuring 10KB of RAM, 48KB and working at the frequency of 8MHz. The algorithm was implemented as reported in Section 3.1.3, with the only exception that the Gaussian neighborhood function was replaced with a triangular function, which produces similar results using much less computation. Table IV

[11] `http://www.tinyos.net`

N. Nodes	Memory	Exec. time Dijkstra	Exec. time for 1000 iter.
36	0.42 KB	1 sec	62 sec
64	1.48 KB	6 sec	102 sec
100	3.42 KB	22 sec	156 sec

Table IV: Memory requirements and execution time of the SOM-A algorithm on a TelosB node.

reports the memory occupation of the data structures described above and the execution time to compute the D_h matrix and then to perform 1000 iterations of the localization algorithm. Even using these highly constrained nodes, it only takes about 3 minutes to localize a network with 100 nodes. During the computation, the radio can be turned off and the microcontroller draws only few milliamp of current, with negligible impact on the energy budget of the sensor node. We conclude by noting that since the algorithm runs with limited overhead on the same hardware used to implement the sensing task, the system reliability will be improved by simply running the computation on a few back-up units.

10. RELATED WORK

Relevant related work has been discussed through the previous sections. General survey papers on localization schemes and positioning techniques have been proposed by several authors (e.g [Hightower and Borriello 2001; Langendoen and Reijers 2003]) and several books on WSNs contain chapters on localization (e.g. [Zhao and Guibas 2004; Raghavendra et al. 2004; Beutel 2005; Stoleru et al. 2007]). In this section we focus on previous localization research using SOM and we discuss the relation between the theoretical aspects of localization and convergence results available for self-organizing maps.

10.1 Localization using SOM

Ertin and Priddy [2005] have used SOM to solve the localization problem in WSNs. Their model is based on the assumption that the network contains N nodes capable of sensing a common phenomena, such as acoustic or seismic, at synchronized time steps. A further assumption is that the correlation between sensor readings s_i and s_j from nodes i and j is a function only of the distance between nodes: $E[s_i s_j] = f(\|p_i - p_j\|)$, where $p_i = (x_i, y_i)$ and $p_j = (x_j, y_j)$ are the physical location of the two sensors. The input samples used to train the SOM are obtained by concatenating the sensor reading collected at each time step: $\mathbf{x_n} = [s_1^{(n)}, \ldots, s_N^{(n)}]$, where $s_i^{(n)}$ is the output of sensor i at time step n. Once the map has been trained with the samples $\mathbf{x_n}$, each neuron contains a weight vector $\mathbf{w_j} = [w_{j1}, \ldots, w_{jN}]$ whose dimensionality is equal to the number of nodes in the network. At this point there is not a direct correspondence between the neurons and the position of the sensor nodes yet, however, the authors suggest that each sensor i can be associated with the neuron j having the largest component w_{ji}. The relative position of such neuron j in the lattice of neurons defines the virtual coordinates of sensor i. No numerical results are provided to characterize the accuracy of the solution, but the author qualitatively describe a possible application to the target tracking problem.

A similar approach has been used by Sakurai et al. [2005] to implement a tracking

application for people moving inside a building. Similarly to the previous case, the input samples used to train the map contain the value sensed by the N sensors installed in the monitored area: $\mathbf{x_n} = [s_1^{(n)}, \ldots, s_N^{(n)}]$. In this application the SOM is not used to compute the physical location, but to create a *logical map* where sensor readings with similar values are grouped together. Again, only a qualitative analysis of the result is presented. Numerical results are instead provided by Xu et al. [2007], who use the SOM to track the movement of people in large outdoor areas using signal strength values measured from nearby cellular stations. In their case the training samples are given by the RSS value collected by the mobile users as they move among the cells covered by several base-stations. This approach and the others described above are substantially different from the scheme presented in this work and have more resemblance with other *fingerprinting* localization techniques (see, for example [Bahl and Padmanabhan 2000; Lorincz and Welsh 2007]).

Takizawa et al. [2006] propose a distributed range-based scheme that uses some of the concepts found in the SOM technique. In this approach, the nodes use a modified version of the update rule (2) discussed in Section 2.1 to iteratively update their position. Interaction between nodes is limited to 1-hop and 2-hop neighbors. This method is similar to the refinement phase used in several range-based schemes (see, for example [Savvides et al. 2002; Savarese et al. 2002]), and it is susceptible to convergence to local minima. An heuristic solution is proposed to avoid this situation. Paladina et al. [2007] also propose a distributed localization scheme based on the use of SOM. Their model assumes nodes deployed in a regular grid, therefore each node can be thought as positioned in the center of a small 3×3 SOM where the remaining eight neurons contain the position of the surrounding one-hop neighbors. Each node uses this small SOM to process the the positions transmitted by its neighbors and compute its own position, which is then propagated to the remaining nodes.

Our approach is analogous to previous applications of the SOM technique to graph drawing [Meyer 1998; Bonabeau and Henaux 1998], a branch of graph theory that deals with the visualization of complex graphs. The graph layout problem is similar to localization in the sense that it also seeks to find a coordinate assignment such that vertices connected by edges are positioned close to each other. But, while the evaluation of a graph layout is mostly based on aesthetic factors (e.g. uniform distribution of nodes and edge lengths, separation between graph elements, number of edge crossing), the results of the localization assignment are directly comparable with the true sensor locations. In this work we explicitly focused on using SOM to produce maps with low localization error.

Finally, we note that the SOM-R described in Section 7 shares some similitude with other range-free schemes where RSS values have been used to complement connectivity information. For example, in the PRI scheme proposed by Li et al. [2004] the received signal strength is used to compute "sub-hops" by sorting the one-hop neighbors, and Liu et al. [2004] compute the node positions as intersection of concentric rings derived from the RSS. The SOM-R scheme also share some resemblance with the work of Nguyen et al. [2005] who used RSS value collected between the nodes to train a kernel-based classifier. However, while the classifier only detects if a node is contained in a given region or not, the SOM technique

implements a more straightforward approach to localization that produces explicit position estimates as a consequence of the training phase of the map.

10.2 Theoretical results

Network localization is a challenging problem that has been actively investigated over the last few years. In addition, since a wireless network can be represented as a graph, relevant results from established fields such as graph theory and computational geometry have helped in putting the problem on firm theoretical basis. As discussed in the introduction, the problem of computing the sensor positions is formally equivalent to the one of embedding a graph in a Euclidean space, which consists in computing the positions of a set vertex using constraints imposed by the length of the edges. This is a NP-hard problem even when accurate internode distances are available [Saxe 1979]. More recently, a number of authors have built on this result to characterize the complexity of the problem specifically for sensor networks, which might contains anchor nodes and where the measurments are likely to be affected by noise. Theoretical results are available for the cases where node have to be localized using ranges [Aspnes et al. 2004; Basu et al. 2006] and angle estimates [Bruck et al. 2005]. In all the case presented the problem is still NP-hard, unless the distance estimates are noise-free and are available for a large-number of nodes [Biswas and Ye 2004; Aspnes et al. 2006]. In addition to the computational complexity of the problem, the solution might be ambiguous unless enough range measurements are available. The characterization of uniquely localizable networks is a problem of particular importance in sensor networks because nodes have limited sensing range and measurements are only possible with a few nearby nodes. The problem of a unique localization solution and its relation to theory of *rigid graphs* has been specifically addressed in a number of works [Eren et al. 2004; Moore et al. 2004; Goldenberg et al. 2005; Aspnes et al. 2006].

Several theoretical results are also specifically available for range-free localization, which is the application considered in our work. An overview of the most relevant theoretical results has been presented by O'Dell et al. [2005]. In particular, a network with connectivity constraints can be modeled as a Unit Disk Graph (UDG)[12] and, similar to the previous case, the localization problem can be posed as one of embedding an UDG in an Euclidean space. This problem is NP-Complete in one dimension and NP-hard in two dimensions [Breu and Kirkpatrick 1998]. Recently, the problem has been proved to be APX-hard [Lotker et al. 2004], meaning that the solution cannot even be efficiently approximated. In fact, there exists node configurations for which even an optimal algorithm cannot produce an embedding with quality better than $\sqrt{3/2}$ [Kuhn et al. 2004]. The only known algorithm with bounded error has been proposed by Moscibroda et al. [2004], who addressed the problem of localization using connectivity constraints (ideal disk connectivity).

The results reported above help in understanding the intrinsic difficulty in computing the node positions and why, in the general case, only approximate solutions are available. The merit of applying the SOM technique to the localization problem is that it provides a low-complexity solution that has been shown to produce accurate localization results in different localization scenarios. Regarding the SOM

[12] A Unit Disk Graph is a graph where two node are connected *iff* their distance is less than 1.

technique itself, despite the attention received, self-organizing maps algorithm have proven to be very resistant to mathematical characterization and theoretical results are only available for the case of one-dimensional configuration of neurons. The first formal proof on ordering and convergence properties of SOM has been presented by Cottrell and Fort [1987] for the case of uniform distribution and step-neighborhood function. The proof has been extended to more general neighborhood function by Fort and Pages [1995], but theoretical results for the two-dimensional case are still incomplete [Cottrell et al. 1998]. However, we note that a lack of formal proof in the general case does not necessarily penalize this approach with respect to other techniques. Given the possible ambiguity in the localization results and additional uncertainty caused by the noise in the measurements, even a solution with proven convergence properties would not be guaranteed to converge to the ground truth. At present, simulations and test-field experimentations are the only tool available to compare the performance of different localization schemes working under realistic system configurations. This is also the approach we have followed in developing our research.

11. CONCLUSIONS

We proposed a centralized algorithm to solve the localization problem for WSNs. The algorithm is proposed in different variants that are able to produce accurate results in situations where other approaches have a poor performance. In addition, the light-weight implementation of the scheme is suitable for resource-poor nodes commonly found in WSN applications.

12. ACKNOWLEDGMENTS

This research work was partly funded by MIUR-Interlink, INT01ACA89 program. We would like to thank Guofeng Deng, Kari Torkkola, Jie Gao, Kenneth Bannister, Lisa Sparbelli and anonymous reviewers for their helpful comments. We also thanks Mary Murphy-Hoye from Intel for the RSS data used in Section 7.3.2.

REFERENCES

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

Aspnes, J., Goldenberg, D., and Yang, Y. 2004. On the computational complexity of sensor network localization. *ALGOSENSORS*.

Bahl, P. and Padmanabhan, V. 2000. RADAR: an in-building RF-based user location and tracking system. *INFOCOM.*

Basu, A., Gao, J., Mitchell, J., and Sabhnani, G. 2006. Distributed localization using noisy distance and angle information. *MobiHoc*, 262–273.

Beutel, J. 2005. *Location managemnent in wireless sensor networks. Handbook of sensor networks: compact wireless and wired sensing systems.* CRC Press, Boca Raton, Fla.

Biswas, P. and Ye, Y. 2004. Semidefinite programming for ad hoc wireless sensor network localization. *IPSN*, 46–54.

Bonabeau, E. and Henaux, F. 1998. Self-Organizing Maps for Drawing Large Graphs. *Information Processing Letters 67,* 4, 177–184.

Breu, H. and Kirkpatrick, D. 1998. Unit disk graph recognition is NP-hard. *Computational Geometry: Theory and Applications 9,* 1-2, 3–24.

Bruck, J., Gao, J., and Jiang, A. 2005. Localization and routing in sensor networks by local angle information. *MobiHoc*, 181–192.

Bulusu, N., Heidemann, J., and Estrin, D. 2000. GPS-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE 7,* 5, 28–34.

Costa, J., Patwari, N., and Hero III, A. 2005. Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks. *ACM Trans. on Sensor Networks*.

Cottrell, M. and Fort, J. 1987. Etude d'un algorithme d'auto-organisation. *Annales de l'Institut Poincaré 23*, 1.

Cottrell, M., Fort, J., and Pagès, G. 1998. Theoretical aspects of the SOM algorithm. *Neurocomputing 21,* 1-3, 119–138.

Doherty, L., Pister, K., and El Ghaoui, L. 2001. Convex position estimation in wireless sensor networks. *INFOCOM*.

Eren, T., Goldenberg, D., Whiteley, W., Yang, Y., Morse, A., Anderson, B., and Belhumeur, P. 2004. Rigidity, computation, and randomization in network localization. *INFOCOM*.

Ertin, E. and Priddy, K. 2005. Self-localization of wireless sensor networks using self-organizing maps. *SPIE 5803*, 138.

Fort, J. and Pages, G. 1995. On the AS Convergence of the Kohonen Algorithm with a General Neighborhood Function. *The Annals of Applied Probability 5,* 4, 1177–1216.

Gerecke, U. and Sharkey, N. 1999. Quick and dirty localization for a lost robot. *Computational Intelligence in Robotics and Automation, CIRA*, 262–267.

Giorgetti, G., Gupta, S. K. S., and Manes, G. 2007. Wireless localization using self-organizing maps. *IPSN*, 293–302.

Goldenberg, D., Krishnamurthy, A., Maness, W., Yang, Y., Young, A., Morse, A., Savvides, A., and Anderson, B. 2005. Network localization in partially localizable networks. *INFOCOM*.

He, T., Huang, C., Blum, B., Stankovic, J., and Abdelzaher, T. 2003. Range-free localization schemes for large scale sensor networks. *MobiCom*, 81–95.

Hightower, J. and Borriello, G. 2001. Location systems for ubiquitous computing. *Computer 34,* 8, 57–66.

Janet, J., Gutierrez, R., Chase, T., White, M., and Sutton, J. 1997. Autonomous mobile robot global self-localization using Kohonen and region-feature neural networks. *Journal of Robotic Systems 14,* 4, 263–282.

Ji, X. and Zha, H. 2004. Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. *INFOCOM*.

Karp, B. and Kung, H. 2000. GPSR: greedy perimeter stateless routing for wireless networks. *MobiCom*, 243–254.

Kaski, S., Kangas, J., and Kohonen, T. 1998. Bibliography of self-organizing map (SOM) papers: 1981–1997. *Neural Computing Surveys 1,* 3&4, 1–176.

Kohonen, T. 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics 43,* 1, 59–69.

Kohonen, T. 1993. Things you haven't heard about the self-organizing map. *Neural Networks, 1993., IEEE International Conference on*, 1147–1156.

Kohonen, T. 2001. *Self-Organizing Maps*. Springer.

Krishnamachari, B., Wicker, S., Bejar, R., and Pearlman, M. 2002. Communications, information and network security, ch. Critical Density Thresholds in Distributed Wireless Networks.

Kuhn, F., Moscibroda, T., and Wattenhofer, R. 2004. Unit disk graph approximation. *Proceedings of the 2004 joint workshop on Foundations of mobile computing*, 17–23.

Kuhn, F., Wattenhofer, R., Zhang, Y., and Zollinger, A. 2003. Geometric ad-hoc routing: of theory and practice. *PODC*, 63–72.

Langendoen, K. and Reijers, N. 2003. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks 43,* 4, 499–518.

Li, X., Shi, H., and Shang, Y. 2004. A partial-range-aware localization algorithm for ad-hoc wireless sensor networks. *LCN*, 77–83.

Lim, H. and Hou, J. 2005. Localization for anisotropic sensor networks. *INFOCOM*.

Liu, C., Wu, K., and He, T. 2004. Sensor localization with Ring Overlapping based on Comparison of Received Signal Strength Indicator. *MASS*, 516–518.

Lorincz, K. and Welsh, M. 2007. MoteTrack: a robust, decentralized approach to RF-based location tracking. *Personal and Ubiquitous Computing 11,* 6, 489–503.

Lotker, Z., de Albeniz, M., and Perennes, S. 2004. Range-Free Ranking in Sensors Networks and Its Applications to Localization. *ADHOC-NOW*, 158–171.

Lymberopoulos, D., Lindsey, Q., and Savvides, A. 2006. An Empirical Characterization of Radio Signal Strength Variability in 3-D IEEE 802.15. 4 Networks Using Monopole Antennas. *EWSN*, 326–341.

Meyer, B. 1998. Self-organizing graphs-a neural network perspective of graph layout. *Proceedings of the 6th International Symposium on Graph Drawing*, 246–262.

Moore, D., Leonard, J., Rus, D., and Teller, S. 2004. Robust distributed network localization with noisy range measurements. *SenSys*, 50–61.

Moscibroda, T., O'Dell, R., Wattenhofer, M., and Wattenhofer, R. 2004. Virtual coordinates for ad hoc and sensor networks. *Proc. of Foundations of mobile computing*, 8–16.

Moses, R., Krishnamurthy, D., and Patterson, R. 2003. A Self-Localization Method for Wireless Sensor Networks. *EURASIP Journal on Applied Signal Processing 2003,* 4, 348–358.

Mulier, F. and Cherkassky, V. 1994. Learning rate schedules for self-organizing maps. *Pattern Recognition. Vol. 2-Conference B: Computer Vision & Image Processing.*

Nagpal, R., Shrobe, H., and Bachrach, J. 2003. Organizing a global coordinate system from local information on an ad hoc sensor network. *IPSN*.

Nguyen, X., Jordan, M., and Sinopoli, B. 2005. A kernel-based learning approach to ad hoc sensor network localization. *ACM Transactions on Sensor Networks (TOSN) 1,* 1, 134–152.

Niculescu, D. and Nath, B. 2001. Ad hoc positioning system (APS). *GLOBECOM*.

Niculescu, D. and Nath, B. 2003. DV Based Positioning in Ad Hoc Networks. *Telecommunication Systems 22,* 1, 267–280.

Niculescu, D. and Nath, B. 2004. Error characteristics of ad hoc positioning systems (aps). *MobiCom*, 20–30.

O'Dell, R. and Wattenhofer, R. 2005. Theoretical aspects of connectivity-based multi-hop positioning. *Theoretical Computer Science 344,* 1, 47–68.

Oja, M., Kaski, S., and Kohonen, T. 2003. Bibliography of self-organizing map (SOM) papers: 1998-2001 addendum. *Neural Computing Surveys 3,* 1, 1–156.

Paladina, L., Paone, M., Iellamo, G., and Puliafito, A. 2007. Self Organizing Maps for Distributed Localization in Wireless Sensor Networks. *ISCC 2007*, 1113–1118.

Patwari, N., Hero, A., Perkins, M., Correal, N., and O'Dea, R. 2003. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal Processing 51,* 8, 2137–2148.

Patwari, N. and Hero III, A. 2003. Using proximity and quantized RSS for sensor localization in wireless networks. *WSNA*, 20–29.

Polastre, J., Szewczyk, R., and Culler, D. 2005. Telos: enabling ultra-low power wireless research. *IPSN*.

Priyantha, N., Balakrishnan, H., Demaine, E., and Teller, S. 2003. Anchor-free distributed localization in sensor networks. *SenSys*, 340–341.

Raghavendra, C., Sivalingam, K., and Znati, T. 2004. Wireless sensor networks.

Rao, A., Papadimitriou, C., Shenker, S., and Stoica, I. 2003. Geographic routing without location information. *MobiCom*, 96–108.

Rappaport, T. 1996. *Wireless Communications: Principles and Practice*. IEEE Press Piscataway, NJ, USA.

Sakurai, A., Nakamura, M., Furubo, S., and Ban, H. 2005. Sensor Localization Using Self-Organizing Map for Human Tracking. *Proc. of Intl. Conf. on Sensing Technology*, 311–315.

Savarese, C., Rabaey, J., and Langendoen, K. 2002. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. *USENIX Technical Annual Conference*, 317–328.

Savvides, A., Garber, W., Adlakha, S., Moses, R., and Srivastava, M. 2003. On the error characteristics of multihop node localization in ad-hoc sensor networks. *IPSN*, 317–332.

Savvides, A., Park, H., and Srivastava, M. 2002. The bits and flops of the n-hop multilateration primitive for node localization problems. *WSNA*, 112–121.

Saxe, J. 1979. Embeddability of weighted graphs in k-space is strongly NP-hard. *Proc. 17th Allerton Conf. Commun. Control Comput*, 480–489.

Shang, Y. and Ruml, W. 2004. Improved MDS-based localization. *INFOCOM*.

Shang, Y., Ruml, W., Zhang, Y., and Fromherz, M. 2003. Localization from mere connectivity. *MobiHoc*, 201–212.

Stoleru, R., He, T., and Stankovic, J. A. 2007. *"Range-free Localization", chapter in "Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks" editors: Radha Poovendran, Cliff Wang, and Sumit Roy.*.

Takizawa, Y., Davis, P., Kawai, M., Iwai, H., Yamaguchi, A., and Obana, S. 2006. Self-Organizing Location Estimation Method using Ad-hoc Networks. *MDM*.

Tenenbaum, J., Silva, V., and Langford, J. 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction.

Vivekanandan, V. and Wong, V. 2006. Ordinal MDS-based localisation for wireless sensor networks. *International Journal of Sensor Networks 1,* 3, 169–178.

Wang, C. and Xiao, L. 2008. Sensor localization in concave environments. *ACM Trans. Sen. Netw. 4,* 1, 1–31.

Whitehouse, K., Karlof, C., Woo, A., Jiang, F., and Culler, D. 2005. The effects of ranging noise on multihop localization: an empirical study. *IPSN*, 73–80.

Xu, J., Shen, X., Mark, J., and Cai, J. 2007. Mobile location estimation for DS-CDMA systems using self-organizing maps. *Wireless communications and mobile computing*, 285–298.

Zhao, F. and Guibas, L. 2004. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann.