



# A study of localization metrics: Evaluation of position errors in wireless sensor networks

Hidayet Aksu, Demet Aksoy, Ibrahim Korpeoglu \*

Bilkent University, Department of Computer Engineering, 06800 Ankara, Turkey

## ARTICLE INFO

### Article history:

Received 8 June 2010

Received in revised form 16 March 2011

Accepted 28 June 2011

Available online 26 July 2011

### Keywords:

Localization

Accuracy

Topology similarity

Relative accuracy

Wireless sensor networks

## ABSTRACT

For wireless sensor network applications that require location information for sensor nodes, locations of nodes can be estimated by a number of localization algorithms, which inevitably may introduce various types of errors in their estimations. How an application is affected from errors and a location error metric's response to errors may depend on the error characteristics. Therefore it is important to use the right error metric to evaluate the error performance of alternative localization techniques that is possible to use for an application. To date, unfortunately, only simplistic error metrics that depend on the Euclidean distance between an actual node position and its estimate in isolation to the rest of the network has been considered for evaluation of localization algorithms. In this paper, we first clarify the problem with this traditional approach and then propose some alternative and new metrics that consider an overall network topology and its estimate in computing a metric value. We compared the existing and new metrics via simulation experiments done using some typical application and error scenarios, and observed that some new metrics are more sensitive to some type of errors and therefore can distinguish better among alternative localization algorithms for applications that are more sensitive to those types of errors. We also go through a case study with some localization algorithms from literature to give an idea about the practical use of our approach. Finally, we provide a step-by-step guideline for selecting the best metric to use for a given sensor network application.

© 2011 Elsevier B.V. All rights reserved.

## 1. Introduction

Wireless sensor networks are used in a wide range of applications such as scientific research, military, health-care, and environmental monitoring [1–3]. In a wireless sensor network, sensor nodes collect information about the environment and communicate their observations to a data collection point, a.k.a. the sink node [1,3], from where users can access the collected data without the need to travel to the monitored area. Sensor nodes tag their observations with their location information and such information is critical for illustrating a representative picture of the monitored environment.

In wireless ad hoc sensor networks, node positions may not be known prior to or at the time of deployment. The process of estimating the unknown node positions within the network is referred to as *localization*. The limited power supply, size and cost considerations in sensor networks may prohibit the use of a GPS (Global Positioning System) module at each sensor node. Instead, it may be preferred to limit the number of nodes with GPS modules and then rely on location estimation algorithms for the rest of the nodes.

Obviously, errors are inevitable in estimations. In Fig. 1(a), we illustrate a simple example for location estimation of three sensor nodes. The actual positions of the nodes are  $P_1$ ,  $P_2$ , and  $P_3$ , and they are represented by solid circles in the figure. The actual positions of the nodes are not known by the application and estimated by use of a localization algorithm. Assume a localization algorithm estimates the node positions as in Fig. 1(a). The estimated positions are

\* Corresponding author. Tel.: +90 312 290 25 99; fax: +90 312 266 40 47.  
E-mail address: [korpe@cs.bilkent.edu.tr](mailto:korpe@cs.bilkent.edu.tr) (I. Korpeoglu).

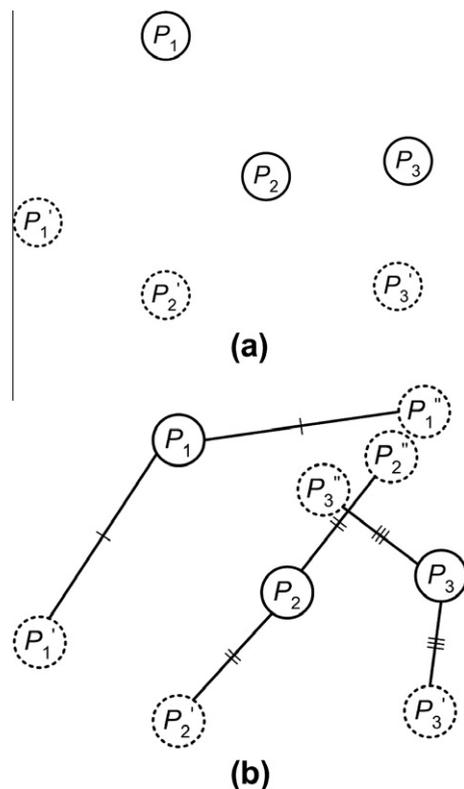
$P'_1, P'_2,$  and  $P'_3$  and shown with dotted circles. Now, assume another localization algorithm produces node position estimates for the same set of nodes as  $P''_1, P''_2,$  and  $P''_3$ , shown in Fig. 1(b).

When comparing the accuracy of these two sets of estimates (i.e. the results of two localization algorithms), the traditional approach uses the Euclidean distance between the actual and estimated positions of the individual nodes,  $P_1, P'_1,$  or  $P''_1,$  etc. In this example, if we consider the accuracy of each estimate in isolation to the estimates of other nodes' positions, this will suggest a similar error in both cases, since the average Euclidean distance for both cases is the same. However, these two sets of estimates may have quite different implications for data management in practical applications. In particular, in the estimates of the second algorithm, the relative standing of the estimated positions  $P''_1$  and  $P''_2$  (and also  $P''_1$  and  $P''_3$ ) are very different in comparison to the relative standing of the actual node positions, and this may result in misleading conclusions during data analysis for some applications. For instance, the advection of a particulate pollutant monitored by an environmental monitoring system may appear to be in the *reverse* direction than it really is.

In this scenario, even though the Euclidean errors are nearly the same for both estimation algorithms, the estimates of the first algorithm ( $P'_1, P'_2,$  and  $P'_3$ ) are much better than the estimates of the second algorithm ( $P''_1, P''_2,$  and  $P''_3$ ), considering relative standing. This simple example of Fig. 1 motivates the need for better metrics to distinguish the error performance of localization algorithms, since Euclidean distance metric is not distinguishing very well for some cases. As another simple example, consider a topology that is simply shifted towards left (or right) in its estimation version. Hence, the estimated position of each node shifted to the same direction with the same amount. In such a case, the Euclidean distance metric will possibly have a large value as the error introduced by the localization algorithm used. But, this kind of estimate may be perfectly fine for some applications that just need to use the relative positions of nodes against each other. Hence, the error of such an estimate by some localization algorithm can be considered as zero or very low for these kind of applications.

In general, the precise location of each sensor node is not necessarily needed in most sensor network applications [1]. Yet, accurate estimate of overall topologies are vital for accurate identification,<sup>1</sup> routing, in-network processing as well as overall analysis of observations. Our focus, therefore, is on the overall estimation of the sensor network topology, rather than on the individual estimates, as has been the major focus in previous studies, e.g., [5–10].

Towards this goal, in this study we first set forth to reply the question: “How do we measure the similarity of two network topologies?” Defining the similarity of two sets of data points, two sequences of coordinates, etc. has been a challenging question in various fields. In this study, we focus on location estimation metrics for



**Fig. 1.** (a) position estimates,  $P'_1, P'_2,$  and  $P'_3$ , for the sensor nodes  $P_1, P_2,$  and  $P_3$ , respectively; (b) comparison with an alternative set of estimates. Individual errors look similar. However, the estimates,  $P''_1, P''_2,$  and  $P''_3$ , result in a completely misleading overall topology.

localization algorithms designed to be used in wireless sensor networks, especially in environmental engineering applications of sensor networks. In this scope, we outline some existing approaches to evaluate the accuracy of position estimates and also propose some novel approaches to address the problems we discussed.

The contributions of this paper are: (1) pointing out the need for a new distance (similarity) measure for localization algorithms in wireless sensor networks, (2) proposing some new metrics; (3) emphasizing that a metric to be used for evaluating alternative localization algorithms depends on the context (i.e. application). Besides proposing and analyzing some new metrics that consider application requirements, we analyze some existing and commonly used metrics as well.

The rest of this paper is organized as follows. In the next section, we briefly discuss the related work. In Section 3, we discuss the meaning of similarity for two topologies. In Section 4 we describe traditional error metrics used in localization studies and also present some new and novel alternatives that can be used within this scope. In Section 5 we discuss some simple topology change scenarios and by using them we evaluate the performance of the new and existing metrics. In Section 6 we go through a case study with some localization algorithms from the literature, and in Section 7 we suggest a metric selection method that can be used to select an appropriate metric for a certain application. Finally, we present our conclusions in Section 8.

<sup>1</sup> For large scale deployments, producing arbitrary addresses for billions of nodes is not feasible; if estimated accurately, geographic locations can help identify nodes, routing, etc.

## 2. Related work

Localization is an important issue for some wireless sensor network applications [2–4,11–14], but not all of such applications require absolute positions of sensor nodes. There are a lot sensor network applications or services that may do quite well with the knowledge of relative positioning of the nodes. For example, a geographic routing or data dissemination service may work equally well if the relative positioning of the nodes against each other are known; in other words, if the topology of the network is known without the exact positions of the nodes. For such applications and services, even though some location errors are introduced by the estimation algorithms, this may not be harmful for the applications provided that the relative positioning information is correct.

Therefore, there are various localization algorithms proposed for sensor networks that do not require use of GPS in every node and that have different error performance. Those localization algorithms can be classified in various ways: range-based algorithms, range-free algorithms, region-based algorithms, connectivity-based algorithms, hop-counting techniques, and so on [15–21]. There are also some localization techniques developed for sensor networks where nodes can be mobile [22].

It is reported in the literature that the errors introduced by those different classes of algorithms may have different characteristics. Hence, there are various types of errors that can be introduced by location estimation algorithms [8]. In this paper, we consider three types of errors: shift, rotational, and random (distorted) errors. There are also studies that investigate the impact of various types of errors on wireless sensor network applications [23].

Nearly all studies consider the average Euclidian distance between actual and estimated node positions as the error metric in judging the accuracy of localization algorithms. There are also metrics that are direct functions of Euclidian distance, for example metrics that use the normalized value of the distance (normalized according to the communication range) [15,16,24]. There are, however, not much studies that consider other metrics that we argue as necessary in this paper.

To the best of our knowledge, this paper is the first attempt to consider various other metrics as well while judging the accuracy of localization algorithms. Some of these other metrics are already well-known in other domains [25,26], but not much in the localization domain. Besides these well-known metrics, this paper also proposes some new novel metrics that can be better to use for some applications (depending on how applications tolerate various types of errors) and therefore can be considered as alternative metrics in localization domain. Hence, we consider our study here is as an original contribution to the literature of performance of localization algorithms.

## 3. Similarity of topologies

In this study we are interested in evaluating metrics that compare and evaluate the difference between two sensor network topologies, one consisting of the actual positions

of the sensor nodes in the network, the other consisting of the estimated positions of the same sensor nodes by some algorithm. To be able to do that we should first define what a topology is, so that we can define the distance or similarity between two topologies. It is possible to come up with various definitions of a topology. One way is to consider a topology as an undirected graph  $G(V,E)$ , where  $V$  is the set of node positions and  $E$  is set of edges so that there is an edge between two nodes that are in the transmission range of each other (assuming symmetric range). According to this definition, topology depends on node positions and on the given transmission range. Performance of localization algorithms, however, does not have to depend on a given transmission range. More important issue is absolute or relative node positions and how we estimate them. Therefore we use another definition of a topology in our study. We define sensor network topology to be a set of  $x$ - $y$  coordinates (i.e., points or node positions) in a two dimensional space. It is possible to extend this definition to three dimensional space considering the altitude of deployed sensor nodes, but for simplicity, our discussion is confined to two-dimensional space.

More formally, when we refer to a topology  $T$  of  $N$  nodes, we refer to a sequence of node positions  $P_1, P_2, \dots, P_N$  where  $P_i = (x_i, y_i)$  is the coordinate of a sensor node  $i$ . We then refer to the estimated topology as  $T'$  consisting of a sequence of coordinates  $P'_1, P'_2, \dots, P'_N$  where  $P'_i = (x'_i, y'_i)$  is the estimated position of the sensor node  $i$ . We use the notation  $d_x(P, P')$  to denote the distance between node positions  $P$  and  $P'$  (nodal/point distance or positional distance), and  $\mu_x(T, T')$  to denote the distance (dissimilarity) between network topologies  $T$  and  $T'$  (topological distance) based on some distance metric  $x$ .

Additionally,  $\vec{V}_{ij} = P_i P_j$  denotes the vector from node position  $P_i$  to node position  $P_j$ ; and  $\vec{V}'_{ij} = P'_i P'_j$  denotes the vector from estimated node position  $P'_i$  to estimated node position  $P'_j$ . Vector  $\vec{V}_{ij}$  indicates the actual relative positioning (arp) of two nodes  $i$  and  $j$ ; and vector  $\vec{V}'_{ij}$  indicates the estimated relative positioning (erp) of two nodes  $i$  and  $j$ .

## 4. Existing and new metrics

In this section we describe different approaches and metrics that can be used to evaluate localization algorithms, and in the next section, Section 5, we evaluate all these metrics when applied to some common scenarios of topology estimations and changes.

We start this section by describing two common metrics, Euclidian distance metric and Manhattan distance metric [27], currently used by localization algorithms followed by the description of two other metrics, Cosine distance metric and Tanimoto coefficient distance metric [25–27], that, to the best of our knowledge, are not applied for localization algorithm evaluation, but can be considered as possible candidates. Then we propose and describe four novel metrics that we think can be alternative metrics for evaluating localization algorithms designed for wireless sensor networks and environmental engineering applications: Relative Euclidian distance (RED) metric, Cumulative vectorial distance (CVD) metric, Extremes distance metric, Spring distance metric. We provide four versions of the

Spring distance metric: Spring A, Spring B, Spring C, and Spring D distance metrics.

Besides the metrics that we consider in this paper, graph-theoretic distance metrics can be considered as well. Graphs are modeled as set of vertices ( $V$ ) and edges ( $E$ ) and this model is appropriate to describe the connectivity of a sensor network. However, a graph-theoretic model does not contain node positions which are particularly used in location estimation problems. Thus, graph-theoretic models do not sense shift, rotation and connectivity preserving distortion on sensor network topology. Therefore, graph-theoretic distance metrics are not considered in this paper and left as a future work.

While describing each metric below, we also illustrate an example in Fig. 2 about how the metric can be computed. Fig. 2(a) contains a sample network of two nodes and its estimation under a localization algorithm. Based on that, Fig. 2(b) through (i) provide examples of how different metrics can be computed to evaluate the error between the given topology and its estimate.

4.1. Euclidean distance

Euclidean distance (error) is the most widely used distance metric. It is defined to be the shortest distance (the length of the straight line) between two coordinates. The

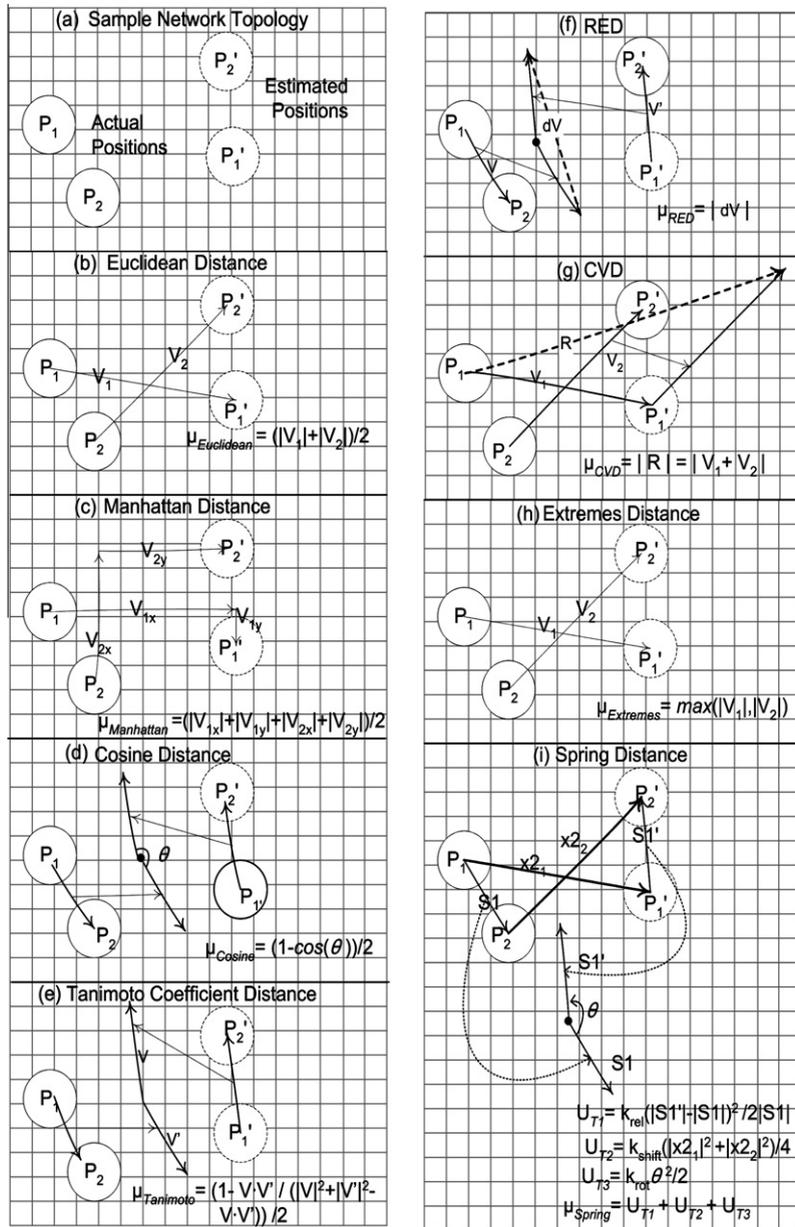


Fig. 2. For a network topology of just two nodes, the figure provides an example for each metric how it can be computed to reflect the distance between the original and estimated topology.

Euclidean distance between two topologies can be computed as follows (see also Fig. 2(b)):

$$d_{\text{Euclidian}}(P_i, P'_i) = \sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2}, \quad (1)$$

$$\mu_{\text{Euclidian}}(T, T') = \frac{1}{N} \sum_{i=1}^N d_{\text{Euclidian}}(P_i, P'_i), \quad (2)$$

where  $x_i$  and  $y_i$  are the actual coordinates of a node  $i$  and  $x'_i$  and  $y'_i$  are the estimated coordinates of the node. In this equation and in the subsequent equations for other metrics,  $N$  is the number of nodes in the network. Above, the topological distance  $\mu_{\text{Euclidian}}(T, T')$  indicates the error introduced by a localization algorithm considering all node positions and their estimates.

In this metric, each node position and its estimate are considered in isolation to other node positions and their estimates. As we discussed in the introduction, however, since this metric does not take the direction or the relative position of a node with respect to other nodes in the network into consideration, it may not be a good metric in applications for which estimating relative positions is more important than estimating absolute positions. Many scientific applications, for example, care more about relative positions of reporting nodes than a few perfect position estimates - without knowing which ones.

#### 4.2. Manhattan (Hamming) distance

Manhattan (Hamming) distance is another simple and popular metric that is computed considering a two dimensional coordinate system. It is the distance between two coordinates measured along the axes at right angles. In other words, assuming that you can move only along the  $x$  and  $y$ -axis in the plane (not in any other arbitrary direction as in the case of Euclidean distance), it measures the distance to get from one point to the other. Similar to Euclidean distance, however, it falls short of representing relative positioning of nodes. Below is the formula to compute the Manhattan distance between a topology  $T$  and its estimate  $T'$  (see also Fig. 2(c)).

$$d_{\text{Manhattan}}(P_i, P'_i) = (|x'_i - x_i| + |y'_i - y_i|), \quad (3)$$

$$\mu_{\text{Manhattan}}(T, T') = \frac{1}{N} \sum_{i=1}^N d_{\text{Manhattan}}(P_i, P'_i). \quad (4)$$

#### 4.3. Cosine distance

Cosine similarity is a well-known technique that considers not only a single value and its estimate, but multiple values at the same time [25]. It is a common metric used in information retrieval domain. In localization domain, we consider vectors  $\vec{V}_{ij}$  and  $\vec{V}'_{ij}$  connecting any two nodes' actual and estimated positions. The Cosine similarity is defined to be the cosine value of the angle ( $\theta$ ) between these two vectors (see Fig. 2(d)). In this respect it is the opposite of the Euclidian distance metric.

Note that Cosine similarity is a good metric for applications that only care about the relative direction of nodes regardless of the actual distance between the pairs of esti-

mates. The absolute distance between nodes, however, is not captured by this metric.

Cosine similarity, like other similarities, has a range of  $-1$  to  $+1$ . We define the Cosine distance between a two-node topology and its estimate as  $(1 - \cos\theta)/2$ . For a topology with more than two nodes, all pairs of nodes are considered, as shown below, to compute the topological distance (Eq. (7)).

$$\cos\theta = \frac{\vec{V}_{ij} \cdot \vec{V}'_{ij'}}{|\vec{V}_{ij}| |\vec{V}'_{ij'}|}, \quad (5)$$

$$d_{\text{Cosine}}(P_i, P_j) = \frac{1 - \cos\theta}{2}, \quad (6)$$

$$\mu_{\text{Cosine}}(T, T') = \frac{2}{N(N-1)} \sum_{i=1}^N \sum_{j=i+1}^N d_{\text{Cosine}}(P_i, P_j). \quad (7)$$

#### 4.4. Tanimoto coefficient distance

Tanimoto coefficient is a more complex metric that considers vectors rather than points [26]. It is a highly popular metric in text matching problems of information retrieval where it is defined as the size of the intersection divided by the size of the union of the sample sets. It can be adapted to our domain as follows. We consider the pairwise relative positions of nodes in both sets ( $T$  and  $T'$ ) as vectors (see Fig. 2(e)). We then compute Tanimoto coefficient ( $TC$ ) of these vectors (Eq. (8)). Using the Tanimoto coefficient, we compute the Tanimoto distance (Eq. (9)) for a node pair. Then the Tanimoto distance between a topology  $T$  and its estimate  $T'$  is computed as in Eq. (10).

$$TC(P_i, P_j) = \frac{\vec{V}_{ij} \cdot \vec{V}'_{ij'}}{|\vec{V}_{ij}|^2 + |\vec{V}'_{ij'}|^2 - \vec{V}_{ij} \cdot \vec{V}'_{ij'}}, \quad (8)$$

$$d_{\text{Tanimoto}}(P_i, P_j) = \frac{1 - TC(P_i, P_j)}{2}, \quad (9)$$

$$\mu_{\text{Tanimoto}}(T, T') = \frac{2}{N(N-1)} \sum_{i=1}^N \sum_{j=i+1}^N d_{\text{Tanimoto}}(P_i, P_j). \quad (10)$$

#### 4.5. Relative Euclidean Distance (RED)

Relative Euclidean Distance (RED) is a novel metric that we propose based on our observations on how Euclidean distance fails to capture the relative position of a pair of nodes. Euclidean distance considers a coordinate in reference to the origin which is a fixed point. With RED metric, instead, we try to capture the relative positional difference between two sets of positions: the actual positions set and the estimated positions set.

To compute RED metric, we consider nodes in pairs. We first compute the RED of one pair. i.e. of two nodes. It is done as follows. Considering any pair of nodes  $i$  and  $j$  in the network and their actual  $(P_i, P_j)$  and estimated  $(P'_i, P'_j)$  positions, we first obtain the vectors  $V = P_1P_2$  and  $V' = P'_1P'_2$ . We then compute the RED metric value as the magnitude of the vector connecting the end-points of these two vectors (see Fig. 2(f) and Eq. (11)).

The value of the metric depends on both the relative directions of the vectors and the difference in the magnitudes of the vectors. For example, if the estimated positions are aligned along the same direction with the actual positions, we expect the angle between the two vectors to be relatively small, indicating that the directional error is low. Similarly, if the vectors have nearly the same magnitude, then their magnitude difference will be low, indicating again a low error value.

The above process is repeated for all pairs of nodes to compute the distance between a topology  $T$  and its estimate. The topological RED distance is the average of the RED distances of all pairs of nodes in the topology (Eq. (12)).

$$d_{RED}(P_i, P_j) = [(|\vec{V}|^2 + |\vec{V}'|^2 - 2\vec{V} \cdot \vec{V}')^{1/2}], \quad (11)$$

$$\mu_{RED}(T, T') = \frac{2}{N(N-1)} \sum_{i=1}^N \sum_{j=i+1}^N d_{RED}(P_i, P_j). \quad (12)$$

#### 4.6. Cumulative Vectorial Distance (CVD)

This metric we propose is motivated by Cosine similarity metric. We aim at including distance as well as the angle into account. First, for each node we record the difference between its actual and estimated  $x$ -coordinate. We repeat the same process for the  $y$ -coordinate. We then sum up all these differences for both  $x$  and the  $y$ -coordinates and construct two perpendicular vectors (starting at the origin) whose magnitudes are equal to these sums respectively. The distance between the end coordinates of these vectors is defined as the CVD metric (Fig. 2(g)). The formula below computes the CVD distance between topologies  $T$  and  $T'$ :

$$\mu_{CVD}(T, T') = \frac{1}{N} \left( \left[ \sum_{i=1}^N (x'_i - x_i) \right]^2 + \left[ \sum_{i=1}^N (y'_i - y_i) \right]^2 \right)^{1/2}. \quad (13)$$

#### 4.7. Extremes distance

The maximum error of location estimation may have a significant effect on some applications and the quality of service they get from the network and location based services running on the network. For such applications, metrics assuring certain error bound (i.e. level of quality in location estimation) are necessary. Thus, we introduce Extremes Distance metric which measures the distance between a topology  $T$  and its estimate  $T'$  as the maximum Euclidian distance among individual node positions and their estimates (see also Fig. 2(h)).

$$\mu_{Extremes}(T, T') = \max_{i \in 1..N} \{d_{Euclidian}(P_i, P'_i)\}. \quad (14)$$

#### 4.8. Spring distance

For this final metric we use an analogy based on a physical model. We model the network topology as an elastic object, and we consider the distance between the actual and estimated topologies as the difference in the potential energy of the original elastic object (corresponding to the

actual topology) and its deformed version (corresponding to the estimated topology).

Hence, we model a sensor network as an elastic object consisting of a set of nodes connected with springs. Each sensor node in the model is connected to all other nodes and the ground with springs.

For a network of  $N$  nodes, there are  $N - 1$  springs per node connecting the node to other nodes. These springs are called Type-1 springs. Each such spring connects a pair of nodes and is responsive (stores potential energy) to a change in the Euclidean distance between those nodes. Moreover, there is one spring per node connecting the node to the ground. This spring is called Type-2 spring and is responsive to the node's individual relocation. In addition to Type-1 and Type-2 springs, which are of tension/extension springs, we have an additional spring per pair of nodes, Type-3 springs, which are torsion springs. A torsion spring is a type of string that can not be extended or compressed, but can be rotated/distorted when a force is applied. A Type-3 string is responsive to a change in the direction of the Type-1 spring connecting these two nodes. Since we have one Type-3 string per pair of nodes, there are  $N - 1$  Types-3 strings associated with a node (one string per other node the node is connected to with a Type-1 string). The potential energy stored on a Type-3 string is related with the angle of distortion ( $\theta$ ) of the string (vector) connecting the corresponding two nodes.

We assume that all strings in the model of the actual network have their relaxed length (equilibrium condition, storing zero potential energy) and then we deform this model into the model corresponding to the estimated network. We then compute the potential energy that is stored in the deformed network model, and this gives us the Spring distance. We know that the more an elastic object is deformed, the more potential energy is stored in it. Therefore, we can consider the stored potential energy as the measure of topological distance.

The potential energy of a tension/extension spring of length  $l$  and elastic modulus or constant  $\lambda$  under compression or extension of  $x$  is  $U_e = \frac{\lambda x^2}{2l}$ . Similarly, the potential energy of a torsion spring of elastic modulus or constant  $k$  with the angle of twist ( $\theta$ ) from its relaxed position is  $U_e = \frac{1}{2}k\theta^2$ .

Spring distance between topology  $T$  and its estimate  $T'$  is the overall potential energy stored in Type-1, Type-2 and Type-3 springs (see Fig. 2(i) and Eq. (15)):

$$\mu_{spring}(T, T') = U_{T1}(T, T') + U_{T2}(T, T') + U_{T3}(T, T'), \quad (15)$$

where

$$U_{T1}(T, T') = \frac{2}{N(N-1)} \sum_{i=1}^N \sum_{j=i+1}^N \frac{k_{rel} \left| |\vec{V}_{ij}| - |\vec{V}'_{ij}| \right|^2}{2|\vec{V}_{ij}|}, \quad (16)$$

$$U_{T2}(T, T') = \frac{1}{N} \sum_{i=1}^N \frac{k_{shift} d_{Euclidian}(P_i, P'_i)^2}{2}, \quad (17)$$

$$U_{T3}(T, T') = \frac{2}{N(N-1)} \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{2} k_{rot} \theta_{ij}^2, \quad (18)$$

$$\theta_{ij} = \arccos \left( \frac{|\vec{V}_{ij} \bullet \vec{V}'_{ij}|}{|\vec{V}_{ij}| |\vec{V}'_{ij}|} \right). \quad (19)$$

$U_{T1}$  computes the potential energy stored in Type-1 springs under tension resulted by changes in relative location of each pair of nodes (Eq. (16)). The type-1 spring constant  $k_{relative}$  is called *displacement\_sensitivity* parameter and assumed to be 1 in the model.  $U_{T2}$  computes the potential energy stored in Type-2 springs under tension resulted by changes in the absolute location of each individual node (Eq. (17)). The relaxed length of a Type-2 string is assumed to be 1 and the type-2 string constant,  $k_{shift}$ , is called *shift\_sensitivity* parameter in the model.  $U_{T3}$  computes the potential energy stored in Type-3 springs under tension resulted by changes in relative direction of each pair of nodes (Eq. (18)). The type-2 spring constant  $k_{rotate}$  is the *rotational\_sensitivity* parameter of the model.

Computational complexity of  $U_{T2}$  is  $O(N)$  while it is  $O(N^2)$  for  $U_{T1}$  and  $U_{T3}$ . Hence computational complexity for Spring distance is  $O(N^2)$  with  $N$  nodes in the network.

Force constants of springs affect the behavior of spring distance metric. By increasing/decreasing the *shift\_sensitivity* and *rotation\_sensitivity* parameters, metric's response to changes can be adjusted. In the simulations, we use four versions of spring distance: Spring A distance is the one with *shift\_sensitivity* = *rotational\_sensitivity* = 0.5; Spring B distance is the one with *shift\_sensitivity* = 1 and *rotational\_sensitivity* = 0; Spring C distance is the one with *shift\_sensitivity* = 0 and *rotational\_sensitivity* = 1; Spring D distance is the one with *shift\_sensitivity* = 0 and *rotational\_sensitivity* = 0.

## 5. Evaluation of metrics under sample scenarios

In this section we present some basic topology change (error) scenarios and use them to compare and evaluate the metrics we presented in the previous section. For each topology change scenario studied, we discuss the impact of those types of errors on applications.

While some distance metrics we study give bounded values, e.g., Cosine distance metric, some others give unbounded values, e.g., Euclidean distance metric. Therefore, comparing the values of the metrics directly, without any normalization, can be misleading. Because of this we normalize each metric's result with its maximum value reported in simulations. In this way, we can monitor the behavior of metrics in response to the changes in the network topology.

We used Matlab for simulating error scenarios and evaluating the metrics. We have written custom Matlab code to simulate various network topologies, topology changes, and to compute the distances between the actual and changed (estimated) topologies according to various metrics we study in this paper. For our simulation experiments, we generate sample network topologies synthetically that are deployed over a square area of 20 by 20 unit length. We keep the area size constant. We consider 10 different network sizes, changing from 40 nodes up to 400 nodes, with a step size of 40. For each simulation experiment, the nodes are deployed on the area with a uniform distribution. For each network size, the simulation experiments are repeated 20 times and average results are reported. There are three basic error scenarios that we consider: shifted topology, ro-

tated topology, distorted (random) topology. For shifted and rotated topology simulations, actual network topology is shifted or rotated depending on the scenario parameters, i.e., rotation angle, and then the resulting topology is used as the estimated topology. In case of distorted topologies, three different distortion approaches are used. First, nodes are distorted by uniform distribution with various ranges. Second, nodes are distorted according to Gaussian distribution with fixed mean and various sigma values. Finally, nodes are distorted by Gaussian distribution with fixed sigma and various mean values.

### 5.1. Rotated topologies

Rotated topologies are common error scenarios for environmental monitoring applications. We focus on topologies that are rotated with respect to a coordinate system. To simulate such a topology change scenario, we place all nodes on a plane and then rotate the plane so that the distance between any two nodes stay exactly the same while the overall alignment differs.

We run simulations for various rotations by increasing the angle of rotation. In Fig. 3, the distance between the original and estimated topology is plotted for various metrics as rotation angle increases. As can be seen from the figure, all metrics, except Spring D metric, report an increasing error as the angle increases up to 180 degrees. The behavior is fully symmetric for all metrics studied, reaching a peak error at 180 degrees and returning back to zero error at 360 degrees, which reflects the original topology.

In traditional pattern matching problems, we would expect similarity degrees to be high in rotated topologies since the shape on the plane does not change when we rotate the plane. Yet, in environmental engineering applications the reference to the coordinate system does play a significant role in the interpretation of the observations from the network.

As seen in Fig. 3, Spring C metric increases exponentially while Tanimoto, Cosine, Spring A, Spring B metrics increase linearly and Euclidean, Manhattan, Extremes and CVD metrics increase logarithmically while the angle of rotation is increased from 0 to 180 degrees. In this regard, among all metrics, Spring C seems to be the most sensitive metric for rotated topologies. On the other hand, Spring D distance metric is not sensitive to rotation operation at all.

### 5.2. Shifted topologies

The second error scenario we consider is a topology with a perfect shift. That is all nodes in the network are subject to the exact distortion in a particular direction. For instance, all nodes deployed on a lake surface may have moved northeast by forces of wind after location estimation. We simulate this scenario by taking the estimated location for a node as  $(x + n, y + n)$ , where  $(x, y)$  is the original coordinate of the node and  $n$  is a number representing the shift amount, between 1 and 10, in both  $x$  and  $y$  dimensions. Even though this is a rather simplified assumption, i.e., in practice some nodes can move more than the others, the scenario will help us to observe the behavior of metrics for the general case of shifted topologies.

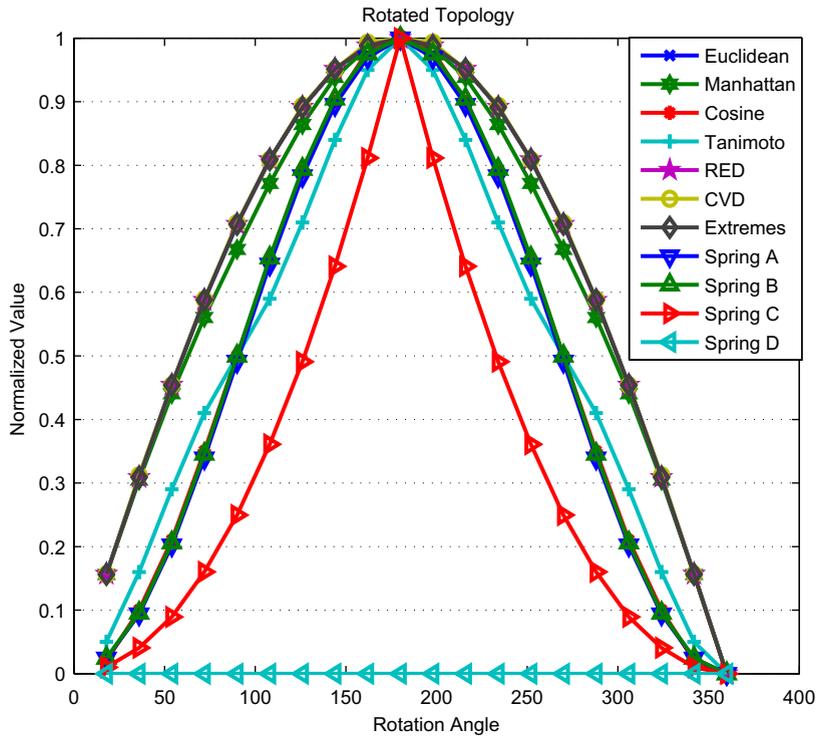


Fig. 3. Behavior of metrics in case of rotated replicas of the original topology: On the x-axis the rotation angle is increased from 0 to 360 degrees and the normalized metric value comparing the original and the rotated topology is reported on the y-axis.

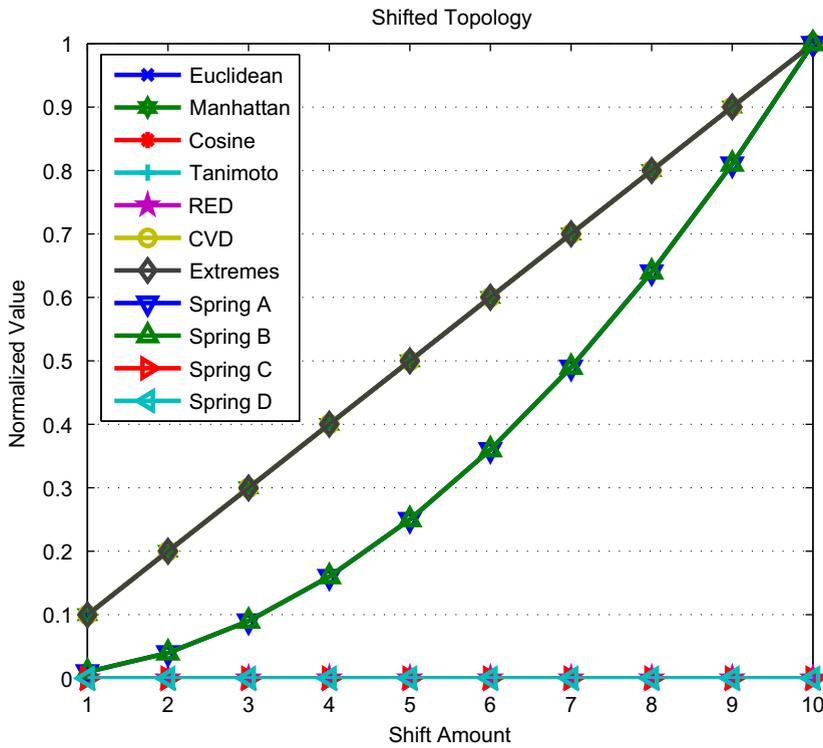
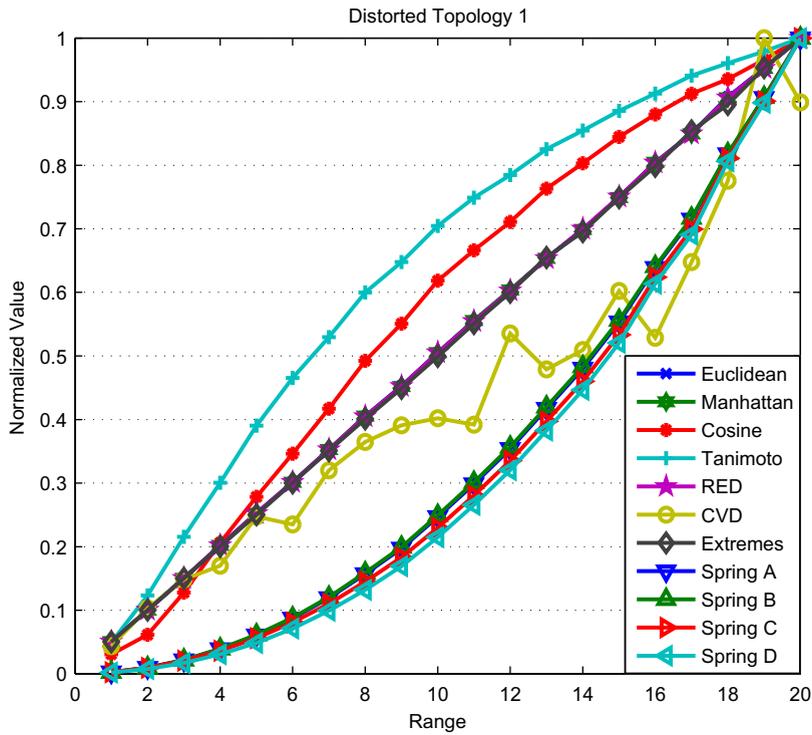
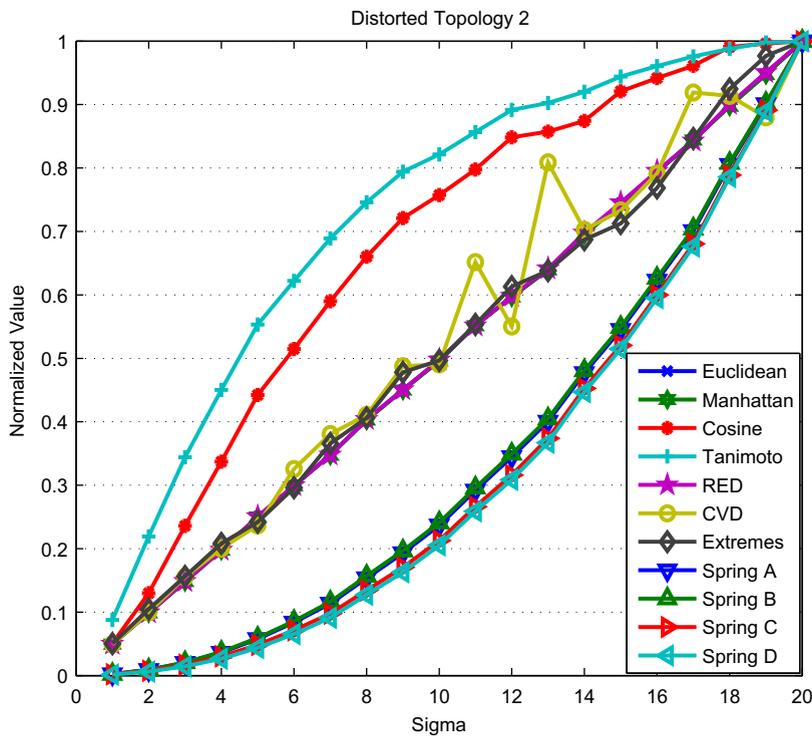


Fig. 4. Behavior of metrics in case of topology shifts. On the x-axis the shift amount is increased from 1 to 10 and normalized metric value is reported on the y-axis. Cosine, Tanimoto, RED, Spring C and Spring D distance report no change for perfect shifts. Spring A and Spring B distance show exponential response; and Euclidean, Manhattan, CVD and Extremes distance show linear response against perfect shift in network topology.



**Fig. 5.** Behavior of metrics in case of topology distortion with uniformly distributed error. On the x-axis, the range is increased from 1 to 20 and normalized metric value is reported on the y-axis. For increasing distortion range, metric values grow logarithmically for Tanimoto and Cosine metric; and linearly for Extremes, Red, Euclidean, CVD and Manhattan metric; and exponentially for Spring group metrics.



**Fig. 6.** Behavior of metrics in case of topology distortion with Gaussian error. Gaussian error is computed with  $\mu = 0$  and increasing the  $\sigma$  from 1 to 20, shown on the x-axis. Normalized metric value is reported on the y axis.

Fig. 4 shows the values of various distance metrics as the amount of the shift ( $n$ ) is increased on the  $x$ -axis. As seen, even though the complete topology (graph representation of the network) is preserved perfectly, Euclidean, Manhattan, CVD, Extremes, Spring A and Spring B distance metrics respond the change. Among these matrices, Spring A and Spring B distance report a distance that is exponentially related to the shift amount, while other metrics report an error that is linearly related to the shift amount. On the other hand, Cosine, Tanimoto, RED, Spring C and Spring D distance metrics report no change for perfect shifts. In shifted topologies, even though the absolute location of nodes is changing, it may not be a major concern for many expert applications. In most cases, shifts that maintain the relative positioning of nodes are acceptable for environmental monitoring applications. For instance, a pollutant flow in northeast direction will still appear in the same direction if all nodes maintain their relative positioning.

### 5.3. Distorted topologies

Distorted topologies represent arbitrary errors made in position estimates. In this category we study a scenario where node positions are shifted along statistically. We apply an independent distortion to each node such that the resulting topology will have some relative accuracy errors. First, we introduce uniformly distributed distortion on each node, where size of the range of distortion changes

from 1 up to 20. As shown in Fig. 5, Tanimoto and Cosine metrics report logarithmic error with respect to distortion range, however, Extremes, RED, Euclidean and Manhattan metrics report error perfectly linear with respect to distortion range. On the other hand, Spring metrics report exponential error with respect to distortion range. Only CVD metric's response is not stable. However, it can also be approximated to a linear relationship with distortion range. Second, we apply a distortion that is Gaussian distributed with a fixed  $\mu = 5$  and an increasing  $\sigma$  from 1 up to 20. In response this distortion, metrics give similar results with the uniform case, as shown in Fig. 6. Finally, we launch distortion that is Gaussian distributed with a fixed  $\sigma = 5$  and an increasing  $\mu$  from 1 up to 20. As illustrated in Fig. 7, CVD, Euclidean, Extremes and Manhattan metrics respond with a linear increase in error, and Spring A and Spring B respond with an exponential increase in error when  $\mu$  is increased. This behavior resembles the previous behavior, however, in this case, Spring D, Spring C, Cosine, Tanimoto and RED metrics show the same high response to Gaussian distortion with  $\sigma = 5$  and  $\mu$  varied.

### 6. A practical case study

In order to illustrate the applicability of our approach to evaluate localization algorithms with various metrics, we perform a case study using a sample wireless sensor network topology. We keep the topology as simple as possible so that visual representation and interpretation of the re-

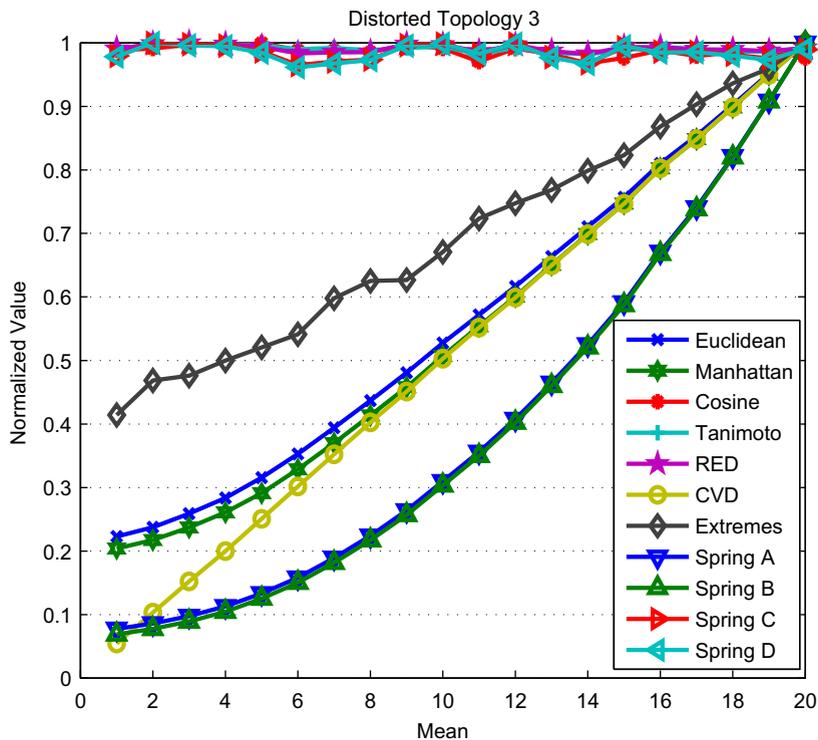
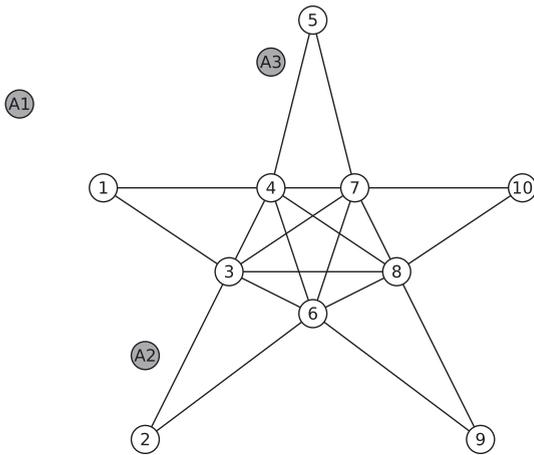


Fig. 7. Distorted Topologies metric behavior in case of topology distortion with Gaussian error. Gaussian error with  $\sigma = 5$  and increasing  $\mu$  is used.  $x$ -axis shows  $\mu$  increasing from 1 to 20, and normalized metric value is reported on the  $y$ -axis. CVD, Euclidean, Extremes and Manhattan metrics response linear error; Spring A and Spring B metrics response exponential error; Spring D, Spring C, Cosine, Tanimoto and RED metrics response constant high error.



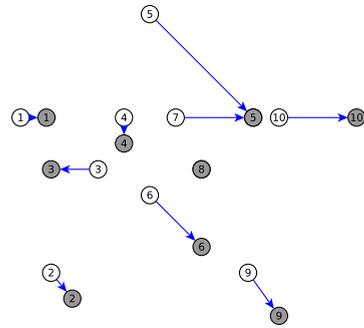
**Fig. 8.** A sample sensor network of 13 nodes. The nodes labeled 1 to 10 are nodes whose locations are unknown and have to be estimated. The nodes labeled A1, A2, A3 are anchor nodes whose locations are known exactly and can be used by localization algorithms to aid in the position estimation of other nodes. There is a line between two nodes that are in the range of each other, indicating a direct wireless link. We do not show links to/from anchor points.

sults become feasible. Fig. 8 shows our sample wireless sensor network which has a total of 13 nodes. The figure shows the actual positions of the nodes. Three of these nodes, A1, A2, A3, are anchor points whose exact locations are assumed to be already known. They may be utilized by a localization algorithm to predict the locations of other nodes. The locations of the other 10 nodes are not known and have to be estimated via a localization algorithm. These nodes are labeled with integers 1 through 10. The figure also gives the reachability information, i.e. network topology. Reachability information shows which node can communicate with which other node directly. Any pair of nodes in the range of each other are assumed to have a direct wireless link between them, hence we have a line connecting them in the figure. The links to/from anchor points are not shown to simplify the figure, since the positions of anchor points will not be estimated.

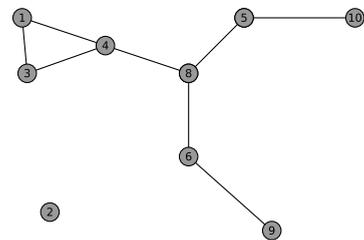
We apply our approach on four localization algorithms selected from the literature. We run simulations so that each algorithm estimates the unknown node positions for the sample network of Fig. 8. Then, for each algorithm, in order to emphasize the nature of localization error of the algorithm, we illustrate the actual and estimated node positions in a figure where there is an arrow pointing from each actual node position to its estimated position. Additionally, we show the estimated reachability graph (estimated network) by using the estimated node positions. We also compute the localization error according to each metric we discuss in this paper. Hence, at the end we have various metric values computed for all localization algorithms. Using this data, for each metric, we can rank the algorithms from best to worst. In this way, we can see how different metrics evaluate the performance of different localization algorithms. We can also see a metric's sensitivity to topology changes.

We use the following sample localization algorithms from the literature: DV-Hop [29], DV-Distance [29], QUAD

[30] and Smooth [31]. They all can utilize one or more anchor nodes which may broadcast their locations periodically. The direct neighbors of anchor nodes estimate their



(a) Actual and DV-Hop-estimated node positions and estimation errors.



(b) Estimated topology by DV-Hop.

Metric	Error
Euclidean	2.21
Manhattan	2.7
Cosine	0.55
Tanimoto	1.32
RED	2.76
CVD	1.64
Extremes	5.66
Spring A	4.38
Spring B	4.18
Spring C	4.59
Spring D	3.99

(c) DV-Hop errors computed by each metric.

**Fig. 9.** DV-Hop Algorithm's performance is visually represented and numerically evaluated by each metric.

positions based on received signal strength value and they propagate their estimates to non-neighbor nodes that are multiple-hops away from the anchor nodes. The algorithms differ in how they propagate the estimations and in the approach used by the rest of network to estimate the distances and positions of the remaining nodes.

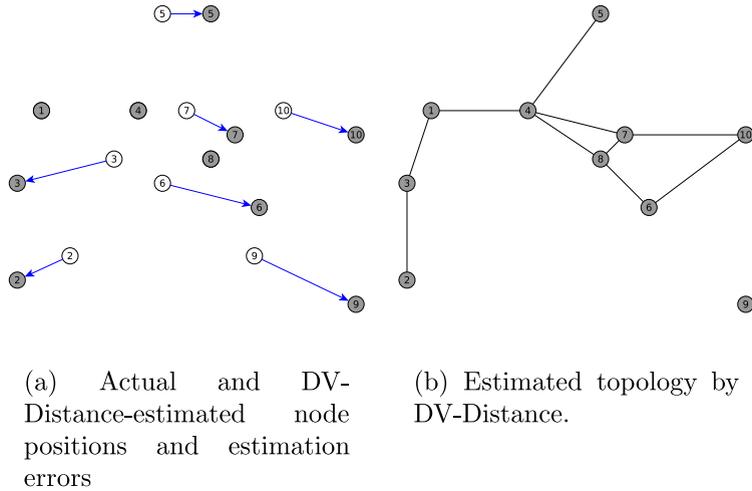
We simulated these localization algorithms over our sample topology and estimated node positions. The results are presented in Figs. 9–12. Each figure is about the results of a different localization algorithm and each figure has 3 sub-parts. In part (a) we show the actual and estimated node positions, in (b) we show the estimated topology (network), and in (c) we show localization error values according to various metrics.

Results for DV-Hop [29] algorithm are demonstrated in Fig. 9. Fig. 9(a) shows that DV-Hop localization algorithm's estimation contains errors in east–west–south directions

with different magnitudes. From Fig. 9(b) we notice that the estimated topology contains 6 actual links, lost 14 links (the links 2–3, 2–6, 3–6, 4–5, 4–6,...) and has two new links (5–8, 5–10).

Fig. 10 demonstrates the performance of the DV-Distance [29] algorithm and how the various metrics are expressing that performance numerically. The Fig. 10(a) shows that nodes 1, 4, 8 are located very well, but other nodes' estimated locations have errors in east–west–south directions. As Fig. 10(b) shows, the estimated topology contains 9 actual links and one new link (6–10). It has, however, 11 links missing.

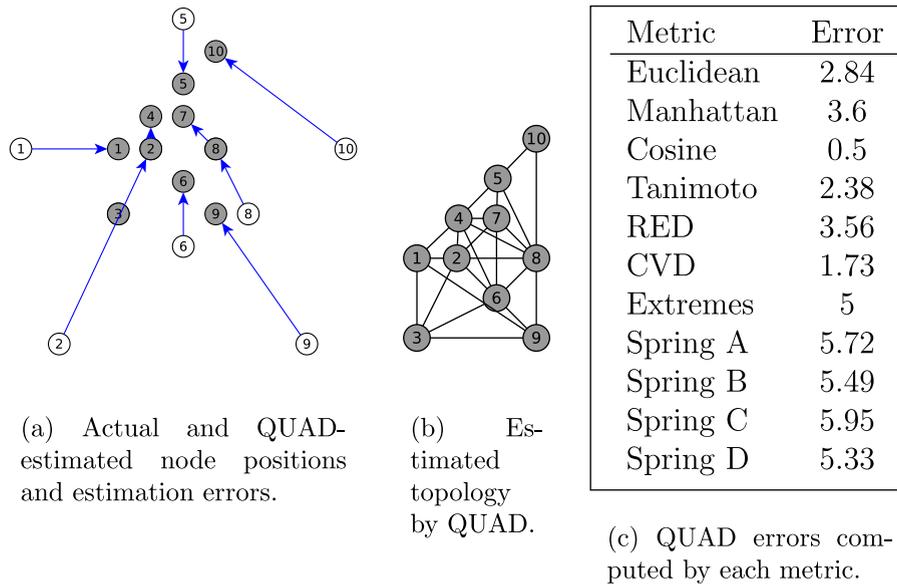
Results for the QUAD localization algorithm [30] are presented in Fig. 11. Fig. 11(a) shows that all estimations are biased towards the center of the network, where estimated locations have errors in all directions. As Fig. 11(b) shows, the estimated topology contains all actual links,



Metric	Error
Euclidean	2.24
Manhattan	2.8
Cosine	0.48
Tanimoto	1.26
RED	3.23
CVD	1.14
Extremes	4.47
Spring A	5.73
Spring B	5.5
Spring C	5.96
Spring D	5.27

(c) DV-Distance errors computed by each metric.

Fig. 10. DV-Distance Algorithm's performance is visually represented and numerically evaluated by each metric.



**Fig. 11.** QUAD Algorithm's performance is visually represented and numerically evaluated by each metric.

however, many new links, i.e., 1–2, 1–5, 1–9, are also introduced. The estimation results in a nearly fully-connected topology.

In Fig. 12, the results of the Smooth algorithm [31], the last localization algorithm we evaluate, are demonstrated. Fig. 12(a) shows that Smooth algorithm's estimation contains errors in east-south directions with similar magnitudes. Therefore, we can say that the network estimation has a shift in about south-east direction. As Fig. 12(b) shows, the estimated topology contains 17 actual links, has three links missing, and has no new links introduced. This estimation has resulted with a topology that is nearly the same with the actual topology except the three lost links.

After computing the localization errors of these algorithms according to various metrics, we create Table 1. In this table, for each metric discussed in the paper, we list the localization algorithms in ascending order of error performance by that metric (the first algorithm listed is the best performing one). As the table shows, there are significant differences in the order of algorithms for various metrics. For instance, Euclidean metric evaluates DV-Hop as the algorithm with minimum error and Smooth as the algorithm with maximum error. Contrary to this, Cosine metric suggests the DV-Hop as the algorithm with maximum error and Smooth as the one with minimum error. Similarly, Tanimoto, RED and Spring metrics suggest Smooth as the algorithm with minimum error. Looking at Fig. 9(b) and Fig. 12(b), it is straightforward for this case to conclude that Smooth algorithm provides a better estimation of the actual network topology than, for example, DV-Hop algorithm.

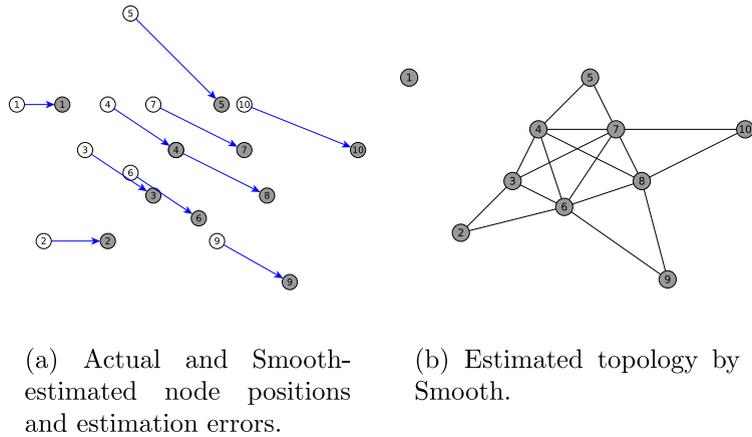
On the other hand, looking at Fig. 9(a) and Fig. 12(a), it is clear that the DV-Hop algorithm provides better estimates for individual node positions when overall topology is not much a concern. Hence, if we are looking for an algorithm to estimate node locations to be used as part of, for example, a geographic routing algorithm, we should

choose Smooth as the localization algorithm. However, if the location estimates are required for a sniper-localization-like application [28], then we should better use DV-Hop algorithm, since for such applications estimating the individual node locations as good as possible rather than estimating the overall topology is more important.

To sum up, according to the location error characteristics of an application, we need to choose the metric which is more appropriate to evaluate possible alternative localization algorithms. Here, we use a small and visually interpretable WSN topology and consider only four localization algorithms. Since we consider only a few localization algorithms, we encounter only a limited number of possible cases. For instance, none of algorithms we used results in a rotated topology, therefore, Cosine, Tanimoto and Spring distances show similar behavior as shown in Table 1. However, with some other algorithms that would result in rotated topologies, we would expect more significant differences in the values of these metrics, considering the simulation results shown in Fig. 3.

## 7. Lessons learned

There are various applications of wireless sensor networks some of which require precise location information, e.g., sniper localization [28], while some other applications may only need relatively accurate location information, e.g., ZebraNet [4] where behaviors of animals are observed. Environmental monitoring applications have a wide range of accuracy requirements. For instance, remote sensing satellite data is a commonly used tool for environmental engineers. One of the problems with the use of remotely sensed data is the calibration requirement for interpreting the measurements. For example, a precisely matching coordinate at the time of the observation is required to make sense of collected data. Ground based sensors are usually deployed for this purpose. Given manageable levels



Metric	Error
Euclidean	3.94
Manhattan	5.2
Cosine	0.13
Tanimoto	0.5
RED	1.7
CVD	3.85
Extremes	5.66
Spring A	2.22
Spring B	2.13
Spring C	2.31
Spring D	2.02

(c) Smooth errors computed by each metric.

Fig. 12. Smooth Algorithm's performance is visually represented and numerically evaluated by each metric.

Table 1

Four sample localization algorithms are ordered according to their performances with respect to various metrics studied in the paper. The first algorithm in an order is the best performing one according to that metric.

Metric	Order of Algorithms
Euclidean	DV-Hop DV-Distance QUAD Smooth
Manhattan	DV-Hop DV-Distance QUAD Smooth
Cosine	Smooth DV-Distance QUAD DV-Hop
Tanimoto	Smooth DV-Distance DV-Hop QUAD
RED	Smooth DV-Hop DV-Distance QUAD
CVD	DV-Distance DV-Hop QUAD Smooth
Extremes	DV-Distance QUAD DV-Hop Smooth
Spring A	Smooth DV-Hop QUAD DV-Distance
Spring B	Smooth DV-Hop QUAD DV-Distance
Spring C	Smooth DV-Hop QUAD DV-Distance
Spring D	Smooth DV-Hop DV-Distance QUAD

Table 2

Metric suggestions for localization algorithms based on the location error response characteristics of the application (i.e. application's sensitivity to location errors) for which the localization algorithms are considered. For example, some applications may not be affected negatively from shift errors, but may be affected negatively from rotational errors. Localization algorithms that will be considered for such an application can be better compared with a metric that is shift-insensitive but rotation-sensitive. A localization algorithm that provides low value on the selected metric may be a good candidate.

Application's sensitivity for location errors	Suggested metrics
Shift-sensitive AND rotation-sensitive	Spring A, Spring B
Shift-sensitive AND rotation-insensitive	Euclidean, CVD
Shift-insensitive AND rotation-sensitive	Spring C, Cosine
Shift-insensitive AND rotation-insensitive	Spring D

of location errors, it may be possible to align an estimated topology reported by ground-based sensor nodes with the region pictured with the remote sensing satellite image to

allow such a calibration. Relative distances play a major role for such calibration such that an alignment is possible.

The impact of errors in location information mostly depends how the applications use the location data. Therefore, while selecting a metric to evaluate some alternative

localization algorithms and their errors, characteristics of the location information required by an application should be considered as well. Note that, the maximization or minimization of a metric value is not significant by itself. The behavior of the metric response against the types of changes in the estimated network topology with respect to the original topology is also important. Therefore, we aggregate the simulation results and analyze them to classify the metric's behavior in a sensitivity table. For the topology change scenarios we simulated, we create groups of metrics responses we have learned from simulations, such as shift sensitive ones. We favor exponential over linear and linear over logarithmic response behavior by giving assigning weights from high to low. Then we obtain the Table 2 by performing appropriate set operations. For instance, we decide about shift and rotate sensitive metrics by intersection of shift sensitive group and rotation sensitive group with highest weight. As a result of this, for example, we decide to use Spring A and Spring B distance which have exponential sensitivity for shift type changes and linear sensitivity for rotation type changes.

For a planned wireless sensor network application, we suggest first identifying and listing the characteristics of the required location data based on its sensitivity to shift and rotation errors. For this purpose, we categorize errors according to a reference coordinate in the deployment plane. Then, appropriate metric can be chosen by looking up the Table 2, in which we suggest metrics according to application requirements on location data errors of algorithms. Subsequently, candidate algorithms may be simulated, and their performance is evaluated by the chosen distance metric. Finally, the localization algorithm which is the most appropriate for the planned application is ready to be picked up.

## 8. Conclusions

A number of algorithms have been proposed for the localization problem in wireless sensor networks. Yet, the evaluation of these algorithms traditionally depends on fairly simplistic metrics based on the original and the estimated coordinates of each node in isolation to the rest of the network. In this paper, we first discussed the implications of errors considering the expectations of end users. We then discussed that there is a need for new metrics that will consider the relative positioning of each node with respect to other nodes for accurate data analysis. We then studied and proposed alternative distance metrics to evaluate localization algorithms. We studied various metrics using some basic topology change (error) scenarios to provide an understanding of how the metrics respond to various type of errors and what can be the implications of these responses to end user applications. We also discussed the advantages of one metric with respect to other ones for some specific applications. We provide a case study, in which we evaluate some localization algorithms from literature using various metrics, to show the applicability of our approach. At the end, we suggest a metric selection methodology that is summarized into a table and that can consider the localization requirements of applications.

## Acknowledgements

This work is supported in part by European Union FP7 Framework Program FIRESENSE Project 244088.

## References

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, *IEEE Communications Magazine* 40 (8) (2002) 102–114.
- [2] H. Alemdar, C. Ersoy, Wireless sensor networks for healthcare: a survey, *Computer Networks* 54 (2010) 2688–2710.
- [3] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, *Computer Networks* 52 (12) (2008) 2292–2330.
- [4] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, D. Rubenstein, Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebnet, in: *ASPLOS: Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, New York, USA, 2002, pp. 96–107.
- [5] L. Lazos, R. Poovendra, SeRLoc: secure range-independent localization for wireless sensor networks, in: *Proceedings of the ACM Workshop on Wireless Security*, Editors: Markus Jakobsson and Adrian Perrig, New York, USA, 2004, pp. 21–30.
- [6] D.C. Moore, J.J. Leonard, D. Rus, S.J. Teller, Robust distributed network localization with noisy range measurements, in: *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, USA, 2004, pp. 50–61.
- [7] R. Nagpal, H. Shrobe, J. Bachrach, Organizing a global coordinate system from local information on an ad hoc sensor network, in: *Proceedings of 2nd International Workshop on Information Processing in Sensor Networks*, Lecture Notes in Computer Science, vol. 634, Springer, 2003, pp. 333–348.
- [8] D. Niculescu, B. Nath, Error characteristics of ad hoc positioning systems, in: *Proceedings of the 5th ACM international symposium on Mobile ad hoc Networking and Computing (MobiHoc)*, New York, NY, USA, 2004, pp. 20–30.
- [9] C. Savarese, J. Rabaey, J. Beutel, Locationing in distributed ad-hoc wireless sensor networks, in: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Salt Lake City, UT, USA, 2001, pp. 2037–2040.
- [10] M. Wong, D. Aksoy, Relative accuracy based location estimation in wireless ad hoc sensor networks, in: *IEEE ICC*, Glasgow, 2007, pp. 3244–3250.
- [11] X. Wanga, O. Bischoffa, R. Laura, S. Paula, Localization in wireless ad-hoc sensor networks using multilateration with RSSI for logistic applications, in: *Proceedings of the Eurosensors XXIII Conference*, Procedia Chemistry, vol. 1, 2009, pp. 461–464.
- [12] S.-R. Hua, S. Peeta, C.-H. Chu, Identification of vehicle sensor locations for link-based network traffic applications, *Transportation Research Part B* 43 (2009) 873–894.
- [13] K.-P. Shih, S.-S. Wang, H.-C. Chen, P.-H. Yang, COLLECT: collaborative event detection and tracking in wireless heterogeneous sensor networks, *Computer Communications* 31 (2008) 3124–3136.
- [14] S.-M. Lee, H. Cha, R. Ha, Energy-aware location error handling for object tracking applications in wireless sensor networks, *Computer Communications* 30 (2007) 1443–1450.
- [15] S. Zhang, J. Cao, Y. Zeng, Z. Li, L. Chen, D. Chen, On accuracy of region based localization algorithms for wireless sensor networks, *Computer Communications* 33 (12) (2010) 1391–1403.
- [16] M. Boushaba, A. Hafid, A. Benslimane, High accuracy localization method using AoA in sensor networks, *Computer Networks* 53 (18) (2009) 3076–3088.
- [17] G. Mao, B. Fidan, B.D.O. Anderson, Wireless sensor network localization techniques, *Computer Networks* 51 (10) (2007) 2529–2553.
- [18] S.-S. Wang, K.-P. Shih, C.-Y. Chang, Distributed direction-based localization in wireless sensor networks, *Computer Communications* 30 (6) (2007) 1424–1439.
- [19] D. Niculescu, B. Nath, DV based positioning in ad hoc networks, *Kluwer Journal of Telecommunication Systems* 22 (1) (2003) 267–280.
- [20] G.J. Jordt, R.O. Baldwin, J.F. Raquet, B.E. Mullins, Energy cost and error performance of range-aware, anchor-free localization algorithms, *Ad Hoc Networks* 6 (4) (2008) 539–559.
- [21] Y. Ding, C. Wang, L. Xiao, Using mobile beacons to locate sensors in obstructed environments, *Journal of Parallel Distributed Computing* 70 (6) (2010) 644–656.

- [22] L. Hu, D. Evans, Localization for mobile sensor networks, in: Proceedings of the Tenth Annual International Conference on Mobile Computing and Networking (MobiCom 2004), Philadelphia, USA, 2004.
- [23] D. Al-Abri, J. McNair, On the interaction between localization and location verification for wireless sensor networks, *Computer Networks* 52 (14) (2008) 2713–2727.
- [24] M. Erol-Kantarci, S. Oktug, L. Vieira, M. Gerla, Performance evaluation of distributed localization techniques for mobile underwater acoustic sensor networks, *Ad Hoc Networks* 9 (1) (2011) 61–72.
- [25] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.
- [26] D.R. Flower, On the properties of bit string-based measures of chemical similarity, *Journal of Chemical Information and Computer Sciences* 38 (3) (1998) 379–386.
- [27] R. Duda, P. Hart, D. Stork, Pattern Classification, second ed., John Wiley and Sons, 2001.
- [28] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J. Sallai, K. Frampton, Sensor network-based countersniper system, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys04), New York, USA, 2004, pp. 1–12.
- [29] D. Niculescu, B. Nath, Ad hoc positioning system (APS), in: Global Telecommunications Conference, 2001, GLOBECOM '01 IEEE, vol. 5, 2001, pp. 2926–2931.
- [30] M. Wong, D. Aksoy, QUAD: Quadrant-based relative location estimates for representative topologies in wireless sensor networks, *Computer Networks* 53 (12) (2009) 1967–1979.
- [31] R. Nagpal, H. Shrobe, J. Bachrach, Organizing a global coordinate system from local information on an ad hoc sensor network, in: Proceedings of Second International Workshop on Information Processing and Sensor Networks, 2003.



**Hidayet Aksu** is a Ph.D. student in Department of Computer Engineering, Bilkent University, Turkey. He received his M.S. and B.S. degrees again from Department of Computer Engineering of Bilkent University. His research interests include wireless networks, wireless ad hoc and sensor networks, localization, and p2p networks.



**Demet Aksoy** is a computer science consultant at Silicon Valley who specializes in problems of wireless communications meeting information management. She received her Ph.D. (doctoral degree) in Computer Science from University of Maryland, College Park. She has more than 15 years of expertise in IT and has been managing diverse projects that mainly focus around wireless information management. Her recent project experience includes information distribution in resource-constrained wireless devices, location estimation in ad hoc networks, self-organization in sensor networks, and multimedia over wireless. She is certified in project management, PMP (Project Management Professional), by PMI.



**Ibrahim Korpeoglu** received his Ph.D. and M.S. degrees from University of Maryland at College Park, both in Computer Science, in 2000 and 1996, respectively. He received his B.S. degree in Computer Engineering from Bilkent University in 1994. He joined Bilkent University in 2002, and he is an Associate Professor in the Department of Computer Engineering. Before that, he worked in several research and development companies in USA including Ericsson, IBM T.J. Watson Research Center, Bell Laboratories, and Bell Communications Research (Bellcore). He received Bilkent University Distinguished Teaching Award in 2006 and IBM Faculty Award in 2009. He is a member of ACM and a senior member of IEEE.