

**AN INQUIRY INTO THE METRICS FOR
EVALUATION OF LOCALIZATION
ALGORITHMS IN WIRELESS AD HOC AND
SENSOR NETWORKS**

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Hidayet Aksu

January, 2008

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. İbrahim K rpeođlu (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr.  zg r Ulusoy

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Sinan Gezici

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

AN INQUIRY INTO THE METRICS FOR EVALUATION OF LOCALIZATION ALGORITHMS IN WIRELESS AD HOC AND SENSOR NETWORKS

Hidayet Aksu

M.S. in Computer Engineering

Supervisor: Assist. Prof. Dr. İbrahim Körpeoğlu

January, 2008

In ad-hoc and sensor networks, the location of a sensor node making an observation is a vital piece of information to allow accurate data analysis. GPS is an established technology to enable precise position information. Yet, resource constraints and size issues prohibit its use in small sensor nodes that are designed to be cost efficient. Instead, most positions are estimated by a number of algorithms. Such estimates, inevitably introduce errors in the information collected from the field, and it is very important to determine the error in cases where they lead to inaccurate data analysis. After all, many components of the application rely on the reported locations including decision making processes. It is, therefore, vital to understand the impact of errors from the applications' point of view. To date, the focus on location estimation was on individual accuracy of each sensor's position in isolation to the complete network. In this thesis, we point out the problems with such an approach that does not consider the complete network topology and the relative positions of nodes in comparison to each other. We then describe the existing metrics, which are used in the literature, and also propose some novel metrics that can be used in this area of research. Furthermore, we run simulations to understand the behavior of the existing and proposed metrics. After having discussed the simulation results, we suggest a metric selection methodology that can be used for wireless sensor network applications.

Keywords: Wireless Sensor Networks, Localization, Topology Similarity, Relative Accuracy, Error Metrics.

ÖZET

TASARSIZ VE ALGILAYICI AĞLARDA YER BELİRLEME ALGORİTMALARININ DEĞERLENDİRİLMESİNDE KULLANILAN METRİKLER ÜZERİNE BİR ARAŞTIRMA

Hidayet Aksu

Bilgisayar Mühendisliği, Yüksek Lisans

Tez Yöneticisi: Yrd. Doç. Dr. İbrahim Körpeoğlu

Ocak, 2008

Algılayıcı ağlarında gözlem yapan bir algılayıcının yeri, doğru veri analizi yapılması için hayati önem arz eder. Küresel Konumlama Sistemi (GPS) hassas yer tayinine imkan sağlayan oturmuş bir sistemdir. Buna karşın, kaynak sınırlamaları ve boyut gereksinimleri söz konusu teknolojinin maliyet etkin olarak tasarlanan küçük algılayıcılarda kullanılmasına olanak sağlamamaktadır. Bunun yerine, çoğu konumlar birtakım algoritmalar kullanılarak tahmin edilmektedir. Bu gibi tahminler sahadan toplanan veride kaçınılmaz olarak hata bulunmasına neden olmaktadır. Hatalı veri analizine yol açabilecekleri durumlarda bu tür hataların saplanması büyük önem arz etmektedir. Bununla birlikte, uygulamaların çoğu bileşenleri karar verme süreçlerinde bildirilen konum verisine itimat etmektedirler. Bu yüzden konumlama hatalarının etkilerini uygulama bakış açısıyla anlamak önemlidir. Bu güne kadar, her algılayıcının ağdan izole edilmiş bireysel konumu, konum belirlemede odak noktası olmuştur. Biz bu tezde, karşılaştırmalarda bütün ağ topolojisini ve görelî algılayıcı konumlarını dikkate almayan yaklaşımlarda bulunan problemlere dikkat çektik. Daha sonra literatürde kullanılan metrikleri tarif edip, bu araştırma alanında kullanılabilir birkaç yeni metrik önerdik. Ayrıca, mevcut ve önerdiğimiz metriklerin davranışlarını anlamak amacıyla simülasyonlar koşturduk. Simülasyon sonuçlarını tartıştıktan sonra kablosuz algılayıcı ağ uygulamaları için kullanılabilir bir metrik seçim metodolojisi önerdik.

Anahtar sözcükler: Kablosuz Ağlar, Yer Belirleme, Topolojik Benzerlik, Göreceli Doğruluk, Hata Metrikleri.

To my parents,

Hacer & M.Şefik

AKSU

Acknowledgement

I would like to express my deepest gratitude to my supervisor Assist. Prof. Dr. İbrahim Körpeoğlu for his invaluable support, continuous encouragement and incredible effort in the supervision of the thesis. It was a wonderful opportunity to work with him.

I acknowledge with thanks and appreciation the jury members, Prof. Dr. Özgür Ulusoy and Assist. Prof. Dr. Sinan Gezici for reviewing and evaluating my thesis.

I would like to thank Assist. Prof. Dr. Demet Aksoy for her invaluable help and suggestions that have been crucial for my work.

I thank to the members of Networking and Systems Group at Bilkent University during the years 2005 to 2007 for their invaluable contributions to my background on the field.

I also thank to the Scientific and Technical Research Council of Turkey (TÜBİTAK) for its support to the project with grant number EEEAG-104E028.

I thank to Uğur Dirim, Ahmet Çetintaş and Mustafa Turan for their review and suggestions on my thesis.

At last but not at least, I would like to express my thanks to my family, making me who I am now with their love, trust, freedom understanding and every kind of support throughout my life.

Contents

- 1 Introduction** **1**

- 2 Preliminaries and Related Work** **8**
 - 2.1 Mathematical Models 8
 - 2.1.1 Network Model 8
 - 2.1.2 Definition of Distance Metric 9
 - 2.1.3 Localization Algorithm Model 10
 - 2.1.4 Set Distance Metric Model 10
 - 2.2 Use Cases of Localization Algorithms 10
 - 2.2.1 Bird Observation 11
 - 2.2.2 ZebraNet 12
 - 2.2.3 Self-Healing Mine Field 13
 - 2.2.4 Sniper Localization 13
 - 2.2.5 Geographic Routing 14
 - 2.3 Metrics Used in the Literature 15

2.3.1	Metric Details	15
3	Proposed Work	25
3.1	The Proposed Metrics	25
3.2	Metric Details	25
3.2.1	Cumulative Vectorial Distance (CVD)	26
3.2.2	Relative Euclidean Distance	30
3.2.3	Normalized Relative Euclidean Distance	32
3.2.4	Spring Distance	34
4	Experiments and Results	39
4.1	Topology Scenarios	39
4.2	Simulation Parameters	40
4.3	Simulation Environment	41
4.4	Evaluation Based on the Metrics	41
4.4.1	Euclidean Distance Behavior	41
4.4.2	Hamming (Manhattan) Distance Behavior	42
4.4.3	Tanimoto Distance Behavior	45
4.4.4	Cosine Distance Behavior	45
4.4.5	CVD Behavior	45
4.4.6	RED Behavior	48
4.4.7	NRED Behavior	48

4.4.8	Spring Distance Behavior	52
4.5	Evaluation Based on the Topologies	55
4.5.1	Rotated Topologies	55
4.5.2	Shifted Topologies	58
4.5.3	Distorted Topologies	60
5	Conclusion and Future Work	62
5.1	Conclusion	62
5.2	Future Work	64

List of Figures

1.1	Graph illustrating nodes as vertices, communication constraints as edges.	3
1.2	Representative topology is more important than reducing the individual errors reported in isolation to the network: P_1 , P_2 , and P_3 are the actual positions of two sensor nodes. P'_1 , P'_2 , and P'_3 are the estimates of one localization algorithm. P''_1 , P''_2 , and P''_3 are the estimates of another algorithm that results in a similar pairwise error. However, the estimates, P''_1 , P''_2 , and P''_3 , result in a completely misleading overall topology from the end users point of view.	4
2.1	Test network setup with $P_1(1, 3)$, $P_2(3, 3)$; $P'_1(1, 2)$, $P'_2(4, 3)$	16
2.2	The vectors representing relative positions of the nodes. All examples are based on this network setup.	16
2.3	Euclidean distance is the straight line distance between two points.	17
2.4	Hamming distance is the distance between two points measured along axes at right angles.	19

2.5 Two sensor node positions P_1 and P_2 are shown with solid circles, with the edge between them describing their actual relative positioning. P_1' and P_2' are the position estimates produced by these nodes and the dashed edge between them is used to define their relative positioning based on the estimated positions. 22

3.1 Test network setup with $P_1(1, 3), P_2(3, 3); P_1'(1, 2), P_2'(4, 3)$ 27

3.2 The vectors representing relative positions of the nodes. All examples are based on this network setup. 27

3.3 The distance between pair (P_1, P_1') and (P_2, P_2') are recorded as vectors V_1 and V_2 . Then, by adding these vectors we get distance dV representing the distance between two topologies. 29

3.4 The distance between pair (P_1, P_2) and (P_1', P_2') are vectors V and V' . Relative Euclidean Distance (dV) is the distance between two vectors' end points when combined at a common starting point. 31

3.5 The vectors $\vec{V} : (P_1, P_2)$, $\vec{V}' : (P_1', P_2')$, $d\vec{V}_{s1} : (P_1, P_1')$ and $d\vec{V}_{s2} : (P_2, P_2')$ are made up of strings. $d\vec{V}_a$ which equals to $|\vec{V}'| - |\vec{V}|$ is the distance observed as absolute change. $d\vec{V}_r$ which equals to $|\vec{V}' - \vec{V}|$ is the distance observed as rotation change. $d\vec{V}_{s1}$ and $d\vec{V}_{s2}$ are the distance observed as shift changes. 34

4.1 Euclidean distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle. 43

4.2 Hamming distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle. 44

4.3 Tanimoto distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle. 46

4.4 Cosine distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle. 47

4.5 Cumulative Vectorial distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle. 49

4.6 RED metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle. 50

4.7 NRED metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle. 51

4.8 Spring A distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle. 53

4.9 Spring B distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle. 54

4.10 Spring C distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle. 56

4.11 Spring D distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle. 57

4.12 Estimation errors in case of estimates that are simply rotated replicas of the original topology: On the x -axis the rotation angle is increased from 0 to 360 and the metric value comparing the original and the rotated topology is reported on the y -axis. 59

4.13 Estimation errors in case of topology shifts. On the x -axis the shift amount is increased from 1 to 10. 60

4.14 Estimation errors in case of topology distortions. On the x -axis the distortion amount is increased from 1 to 10. 61

5.1 Suggested steps to choose appropriate algorithm for a planned wireless sensor network application. 63

List of Tables

2.1	List of the metrics used in the literature. Details of the metrics are given in Section 2.3.1.	15
3.1	List of the metrics developed in study of this thesis. Details of the metrics are given in Section 3.2.	26
3.2	List of Spring Distance variations.	36
5.1	Metric suggestions based on location characteristics.	63

Chapter 1

Introduction

As a result of recent improvements in integrated circuits and device manufacturing, deployment of sensor nodes, which are small, inexpensive, low-power, distributed devices with the capability of processing and wireless communication, are becoming feasible [26, 32]. Sensor nodes are the simplest intelligent devices used currently, main purpose of which is monitoring the environment near them and giving alerts about the main events that are taking place. Applications and systems built above them can make decisions according to the observations received from these devices [7]. Although each sensor node has only a limited processing capability, a group of sensor nodes working in coordination can achieve the ability to monitor the environment in detail. Therefore, a sensor network can be described as a group of sensor nodes which can perform some specific task in coordination. Dense deployment and close coordination is usually essential for sensor networks to carry out the task expected from them [2].

Wireless sensor networks (WSN) are emerging as an important platform that is built on specialized hardware and network structure on which many applications related to distinct areas can run. Those applications include but are not limited to environmental monitoring, industrial and manufacturing automation, health-care, and military. Generally, wireless sensor networks are limited with regard to power resources and computational capacity [23].

WSNs are targeted for various classes of applications, thus different objectives are considered in their design. Their design may aim to take required action on time, so, they usually monitor a certain environment to detect the occurrences of possible events. Their design may also aim to understand the behavior of the monitored entity, therefore, they gather and process data from a certain environment [23].

A fundamental issue in WSNs is determination of where sensor nodes reside, i.e. determination of their positions. In essence, sensor nodes collect information about the environment and transfer their observations to a data collection point, a.k.a. the sink node [1], from where users can access the collected data without the need to travel to the monitored area. In this regard, every user has to depend on the location information provided by the sensor node that reports an observation. As a result, the users view of the monitored area highly depends on reported locations, therefore, it is critical to illustrate a representative picture of observations to users. In ad hoc sensor networks, node positions are not known prior to the deployment. In extreme cases, sensor nodes are dropped from the air and scattered. 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 prohibit the deployment of GPS (Global Positioning System) at each sensor node. Instead, it is preferred to limit the number of nodes with GPS antennas and then rely on location estimation algorithms for the rest of the nodes.

In the network shown in Figure 1.1, the solid nodes represent nodes with known positions and the open nodes represent nodes with unknown positions. Location estimation algorithms estimate the positions of open nodes given the positions of solid nodes. In other words, the problem is:

- Given: the positions of some nodes (solid nodes)
- Find: the positions of all other nodes (open nodes)

Formally, the network is a graph where nodes are represented by vertices and bi-directional communication constraints are shown by edges. Positions of some

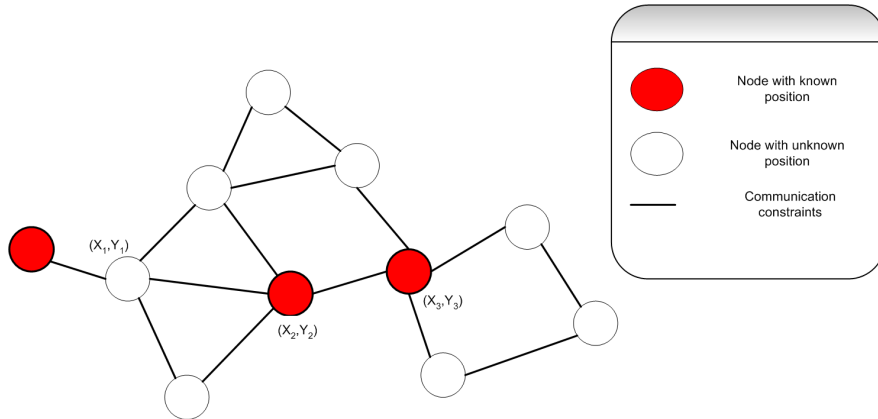


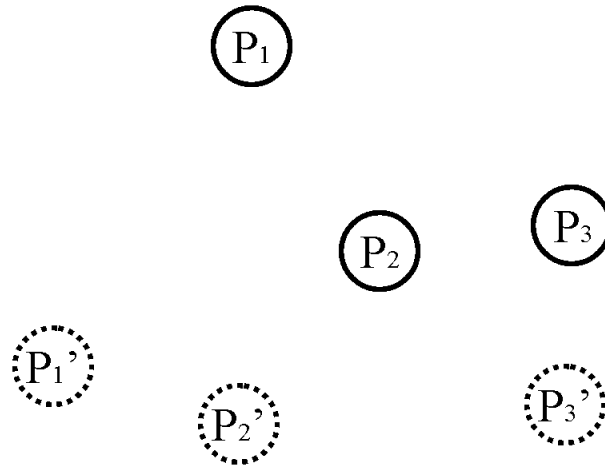
Figure 1.1: Graph illustrating nodes as vertices, communication constraints as edges.

nodes are known while the remaining positions are not known. The localization problem is then to find best approximation for unknown positions.

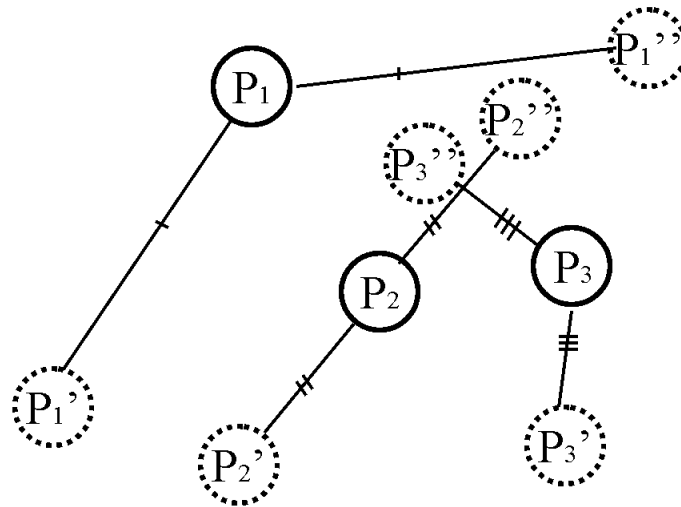
Obviously, errors are inevitable in estimations, and it is important to understand the impact of errors for a particular application. In Figure 1.2, we illustrate a simple example with three sensor nodes. The actual positions of P_1 , P_2 , and P_3 are represented by solid circles in the figure. Recall that, in practical applications, the actual positions of these nodes would not be known, and application user will have to depend on the position estimations reported by the nodes. In Figure 1.2(a), we plot location estimates of these nodes by a localization algorithm as P'_1 , P'_2 , and P'_3 using dotted circles.

Now, consider another localization algorithm that produces location estimates of P''_1 , P''_2 , and P''_3 for the same nodes as demonstrated in Figure 1.2(b). Following the traditional approach in localization studies, we would evaluate these two sets of estimations based on the Euclidean distance between the real and the estimated positions of individual nodes.

When considered in isolation as in previous work, this would suggest a similar error in both cases. However, these two sets of estimates have quite different impacts for data management in practical applications. In particular, the relative positions of P''_1 and P''_2 , and also P''_1 and P''_3 , etc. are incorrect in comparison to



(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 with similar pair-wise errors.

Figure 1.2: Representative topology is more important than reducing the individual errors reported in isolation to the network: P_1 , P_2 , and P_3 are the actual positions of two sensor nodes. P_1' , P_2' , and P_3' are the estimates of one localization algorithm. P_1'' , P_2'' , and P_3'' are the estimates of another algorithm that results in a similar pair-wise error. However, the estimates, P_1''' , P_2''' , and P_3''' , result in a completely misleading overall topology from the end users point of view.

the original deployment. In consequence, this may result in misleading conclusions during data analysis. For instance, the advection of a particulate pollutant may appear to be in the reverse direction than it really is. In this scenario, even though the Euclidean errors are the same, estimates $(P'_1, P'_2, \text{ and } P'_3)$ are much better than $(P''_1, P''_2, \text{ and } P''_3)$.

In general, the precise location of each sensor node is not necessary in most sensor network applications [1]. Yet, accurate overall topologies are vital for accurate identification¹, routing, in-network processing as well as overall analysis of observations. Our focus, therefore, is on the overall sensor network topology constructed, rather than on individual estimates as has been the major focus in previous studies, e.g. [24], [28], [15], [19], [22].

A network topology can be considered as a figure consisting of points that represent the positions of each sensor node in the field. We, then, can consider the similarity of the two figures constructed to represent each topology. Yet, even when we reduce the problem to a well-studied field, there is no universal definition of what figure is.

Indeed, definition of a figure and figure matching have been a major research problem for hundreds of years, finally taking its form in computer science fields such as computational geometry, etc. The traditional problem, however, deals with transforming a figure (shape), and measuring the resemblance with another one, using some similarity measure. For general figure matching, we have a wide range of similarity measures which depend on the particular application at hand, boundary matching, texture matching, etc. Yet, in sensor networks we are interested in a superset of attributes than those already studied. For instance, even when two figures are exactly the same, point-to point, it is still possible that some points have switched their positions and actually report erroneous positions with the observations they make. This makes comparison of two network topologies, in particular, one consisting of actual node positions, and the other consisting of estimates made by those nodes, a unique problem.

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

The main issue, after all, lays in the definition of a topology. What is a topology, so that we can define the similarity of two topologies. It is obviously possible to come up with various definitions of a topology. In our study, we define topology to be a set of (x, y) coordinates in a two dimensional space. It is possible to extend this definition to three dimensional space considering the altitude of deployed sensors. Yet, for the sake of simplicity, and to enable a common comparison basis with existing metrics, we will build our discussion on the two dimensional space.

In general, defining the similarity of two sets of data points, two sequences of coordinates, etc. has been a challenging question studied in a wide range of computer science fields, i.e., information retrieval, graphics, genome studies, etc. In our study, we focus on certain topological changes for sensor networks. Making use of common-sense expectations, we outline some existing approaches to evaluate the accuracy of position estimates and also propose some novel approaches to address the problems we discussed.

Many algorithms have been proposed in the literature [4, 11, 20, 30, 21, 29, 12] on localization estimation problem of wireless sensor networks, however it is still not clear how to choose which algorithm for a claimed application. Having so many localization algorithms for wireless sensor networks, a method is required to evaluate the success of available algorithms. Currently, Euclidean distance is used for the evaluation of the localization algorithms. However, Euclidean distance is not aware of the topology to which it is applied and thus, using this method can be deceptive. Therefore, an inquiry into the metrics, from the perspective of being used for measurement of topological distance between a given network and its estimate, is required to assess the metrics and to find out the circumstances they are reliable.

The main contributions of this thesis can be listed as follows:

1. We point out the need for a new distance measure for evaluation of localization algorithms.
2. We analyze different metrics that can be employed for location estimation

algorithms evaluation.

3. We implemented a basic metric evaluation framework which makes the evaluation of the metrics and algorithms easier.
4. We propose a set of alternative metrics depending on the application requirements.
5. We also suggest a methodology for metric selection based on the localization needs of a wireless sensor network application.

The rest of this thesis is organized as follows. In Chapter 2 we provide some background information and then describe traditional error metrics used in localization studies. Then in Chapter 3, we present some novel alternatives that can be used within this topic. We then discuss some simple scenarios to discuss the performance of these metrics for various applications in Chapter 4. Finally, we present our conclusions in Chapter 5.

Chapter 2

Preliminaries and Related Work

In this chapter we will first give some preliminary information and then briefly describe some related work on evaluation of localization algorithms.

2.1 Mathematical Models

In the rest of thesis, following mathematical models will be used.

2.1.1 Network Model

Let a network be represented by a set $S = (P_1, \dots, P_N)$, with the nodes defined by P_i in the Euclidean space where P_{ix} and P_{iy} indicates (x, y) coordinates of i 'th node in the set.

The above definition of the network topology, i.e. a set of node positions, becomes identical with the visibility graph representation of the network, provided that we fix a value for the maximum transmission range of the sensor nodes. This is because we can easily derive the visibility graph representing a network when we know the node positions and the maximum transmission range (assuming all

nodes have the same maximum transmission range and assuming the ranges are symmetric). This definition will be used in the rest of this thesis.

2.1.2 Definition of Distance Metric

In the literature, a distance metric on a set X is a function, called the distance function, $d : X \times X \rightarrow R$, where R is the set of real numbers. For all x, y, z in X , this function is required to satisfy the following conditions:

1. $d(x, y) \geq 0$ (non-negativity)
2. $d(x, y) = 0$ if and only if $x = y$ (self-distance)
3. $d(x, y) = d(y, x)$ (symmetry)
4. $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

In this thesis, the following metrics are discussed:

- Euclidean Distance,
- Hamming (Manhattan) Distance,
- Tanimoto Distance,
- Cosine Distance,
- Cumulative Vectorial Distance,
- Relative Euclidean Distance,
- Normalized Relative Euclidean Distance,
- Spring Distance.

2.1.3 Localization Algorithm Model

Let a localization algorithm be represented by a function $\delta_f : S \rightarrow S'$, with S representing a wireless sensor network (i.e. a set of node positions) and S' representing its estimation computed by the localization algorithm f .

2.1.4 Set Distance Metric Model

Let a set distance metric be represented by a function $\mu : (S, S') \rightarrow R$, where S is a wireless sensor network, S' is its estimation computed by a localization algorithm, and R is the set of real numbers.

For all networks S, S', H , this function is required to satisfy the following conditions:

1. $\mu(S, S') \geq 0$ (non-negativity)
2. $\mu(S, S') = 0$ if and only if $S = S'$ (self-distance)
3. $\mu(S, S') = \mu(S', S)$ (symmetry)
4. $\mu(S, S') \leq \mu(S, H) + \mu(H, S')$ (triangle inequality)

This is an extension of the metric given in Section 2.1.2 which just gives the distance between two points. This new metric model gives also the distance between two sets of points.

2.2 Use Cases of Localization Algorithms

In this section we explore the cases in which a wireless sensor network makes use of location information. Since localization algorithms' aim is to approximate the location of nodes, a research on use cases of location information in wireless sensor

networks is useful and may be used as a guide and benchmark for assessment of localization algorithms. By having a study of use of location information in a number of applications, we can justify our proposed metrics by comparing their result and expected ones.

In order to understand the dependency of applications on location information and the impact of errors to application design objectives, we define two sensitivity classes regarding location estimations and errors for applications running on sensor networks. These classes are:

1. *shift* sensitive or insensitive class, which indicates that whether the application is responsive to shift errors in location information of the whole topology.
2. *rotation* sensitive or insensitive class, which means that whether the application is responsive to rotation effect in location information of the whole topology.

In other words, when the estimated positions of all nodes in a network shift together with their actual positions, if the application running on this network does not come up with a misleading conclusion, then the application is said to be shift insensitive. The same argument works for rotation sensitivity and insensitivity cases.

Among various wireless sensor network applications, some well documented, representative applications are briefly described below.

2.2.1 Bird Observation

In order to observe the breeding behavior of birds on Great Duck Island, Maine, USA [17], a wireless sensor network is deployed. Since sensors can easily be deployed on a small island where studying individually might be unsafe or unwise, and since sensors do not have disturbance effects on birds, a wireless sensor network is used to understand the behavior of birds.

The biologists are primarily interested in the usage pattern of birds' nesting burrows, changes in environmental conditions outside and inside the burrows during the breeding season, variations among breeding sites, and the parameters of preferred breeding sites.

Sensor nodes used in this application can measure humidity, pressure, temperature, ambient light level, and sensors are installed inside the burrows and on the surface.

Sensed data with location information is used to study the

- usage patterns of birds' nesting burrows,
- changes in environmental conditions outside and inside the burrows during the breeding season,
- variations among breeding sites,
- parameters of preferred breeding sites.

Study goals make it necessary to know the sensor positions in real environment, so location information is sensitive to both shift and rotation errors.

2.2.2 ZebraNet

To observe the behavior of animals within a large habitat [13], a wireless sensor network is deployed at the Mpala Research Center in Kenya. Main goal is to study the behavior of individual animals, interactions inside a species, interactions among different species, and the impact of human activities on the species. The study is planned for a year or more.

Sensor nodes are equipped with light sensors, and nodes are deployed on the studied animals. Further sensors (head up or down, body temperature, ambient temperature) are planned for the future.

Sensed data with location information is used to study the

- individual animals activity patterns of grazing, graze-walking, and fast moving,
- interactions inside a species,
- grouping behavior and group structure of species.

Location information used for all of these study goals is required to be relatively accurate, so node positions are shift insensitive and rotation insensitive.

2.2.3 Self-Healing Mine Field

Main idea is to have anti-tank land mines equipped with sensing and communication capabilities to make sure that a certain area remains covered after a mine is destroyed to create a breach lane [18]. When the mine network detects a tamper in the network, one of the undamaged mines is selected and the mine jumps to the breach using its specialized hardware.

In this application the location information is used to

- detect nodes failures,
- move toward the failed node.

For all use cases in which location information is used, positions are required to be accurate relative to the network, so node positions are shift insensitive and rotation insensitive.

2.2.4 Sniper Localization

In order to locate trajectory of bullets and snipers [31], a wireless sensor network is used. Data gathered by this network provides valuable information for law enforcement. Nodes measure the muzzle blast and shock wave using acoustic

sensors, then the sensor nodes form a multi-hop ad hoc network, and by comparing the time of arrival at distributed sensor nodes, the sniper can be localized with an accuracy of about one meter, and with a latency of under two seconds. The sensor nodes use a special hardware to carry out the complicated signal processing functions.

In sniper localization application, location information is used for following objectives

- localize sniper,
- locate trajectory of bullets.

The objectives of this application make it necessary to use the absolute position of nodes, not positions relative to the network, therefore node positions are shift sensitive and rotation sensitive.

2.2.5 Geographic Routing

Geographic routing is a routing approach that is based on geographic position information. It is based on the idea that the source node sends messages to the geographic location of the destination node instead of using the network address.

Geographic routing uses location information in order to

- determine the route to destination

In geographic routing, decisions are made based on node locations and these decisions are used for routing in the same network. Use of location information in this application requires accuracy of positions relative to the network, therefore, node positions are shift insensitive and rotation insensitive.

2.3 Metrics Used in the Literature

In this section we present some existing metrics that can be applied to measure the distance between the actual and the estimated topologies, S and S' . The metrics studies in the literature are summarized in Table 2.1.

Name:	Formulation (For N Points):
Euclidean Distance	$\mu(S, S') = \frac{1}{N} \sum_{i=1}^N \left[(x'_i - x_i)^2 + (y'_i - y_i)^2 \right]^{1/2}$
Hamming Distance	$\mu(S, S') = \frac{1}{N} \sum_{i=1}^N x'_i - x_i + y'_i - y_i $
Tanimoto Distance	For each pair of nodes P_1 and P_2 in S , $d(P_1, P_2) = \left(1 - \vec{V} \cdot \vec{V}' / \left(\vec{V} ^2 + \vec{V}' ^2 - \vec{V} \cdot \vec{V}' \right) \right) * 10$ where $\vec{V} = P_1 \vec{P}_2$ and $\vec{V}' = P'_1 \vec{P}'_2$ then $\mu(S, S') = \frac{2}{N*(N-1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right)$
Cosine Dis- tance	For each pair of nodes P_1 and P_2 in S , $d(P_1, P_2) = \left(1 - \vec{V} \cdot \vec{V}' / \left(\vec{V} \times \vec{V}' \right) \right) * 10$ where $\vec{V} = P_1 \vec{P}_2$ and $\vec{V}' = P'_1 \vec{P}'_2$ then $\mu(S, S') = \frac{2}{N*(N-1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right)$

Table 2.1: List of the metrics used in the literature. Details of the metrics are given in Section 2.3.1.

2.3.1 Metric Details

Here we describe the metrics in detail. For each metric we first give its description followed by its formulation, then provide the motivation behind it if applicable, and finally run it on an example network and its estimation. The example network is the sample network setup shown in Figure 2.1, where the actual network S and its estimation S' are given as follows:

$$S = \{P_1(1, 3), P_2(3, 3)\} \text{ and } S' = \{P'_1(1, 2), P'_2(4, 3)\}$$

For this network, the vectors $\vec{V} = P_1\vec{P}_2$ and $\vec{V}' = P_1'\vec{P}_2'$ represent the relative positions of the nodes. Moving these vectors to origin, we get the setup in Figure 2.2.

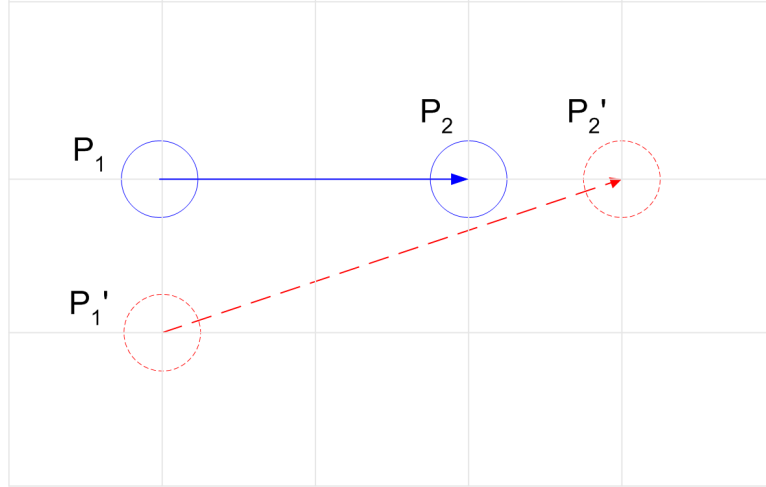


Figure 2.1: Test network setup with $P_1(1, 3), P_2(3, 3); P_1'(1, 2), P_2'(4, 3)$.

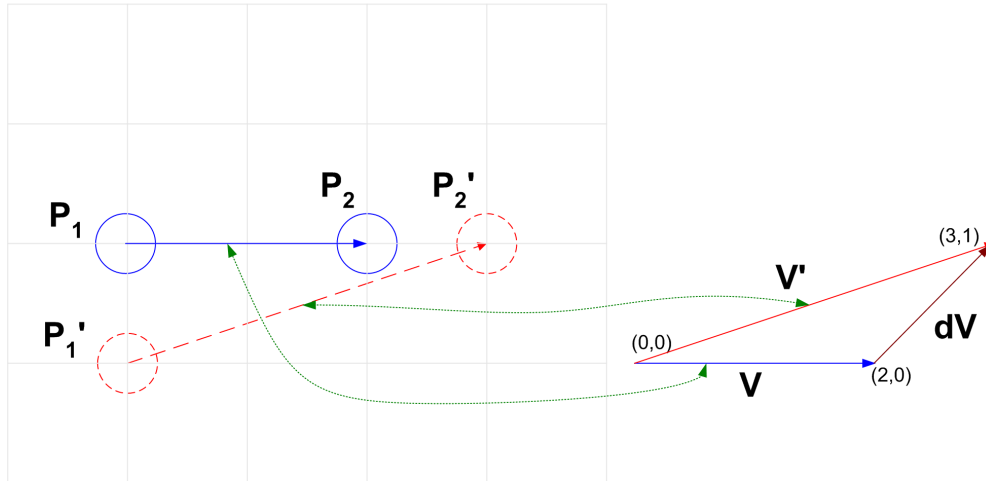


Figure 2.2: The vectors representing relative positions of the nodes. All examples are based on this network setup.

2.3.1.1 Euclidean Distance

Euclidean distance (error) is the most widely used distance metric. Vest majority of the studies on wireless sensor network localization issues make use of this

metric [5, 33, 38, 14, 6, 34, 16, 27, 25, 35, 8, 10, 3, 37, 36]. It is defined to be the shortest distance (the length of the straight line) between two points.

In literature, Euclidean distance between two sets of points is computed by

$$\mu_{euclidean}(S, S') = \frac{1}{N} \sum_{i=1}^N \left[(x'_i - x_i)^2 + (y'_i - y_i)^2 \right]^{1/2} \quad (2.1)$$

where S and S' contain N points, and x_i and y_i are the actual coordinates of the node i while x'_i and y'_i are the estimated coordinates of the node i . For sensor network topologies, this metric has been applied using the set of actual node positions and estimated node positions and the average error has been reported as the overall error of the localization algorithm. As we discussed in the introduction, however, since this metric does not take the relative position of a node with respect to other nodes in the network into consideration, it is prone to be erroneous for applications for which relative positions of nodes are more important than absolute positions.

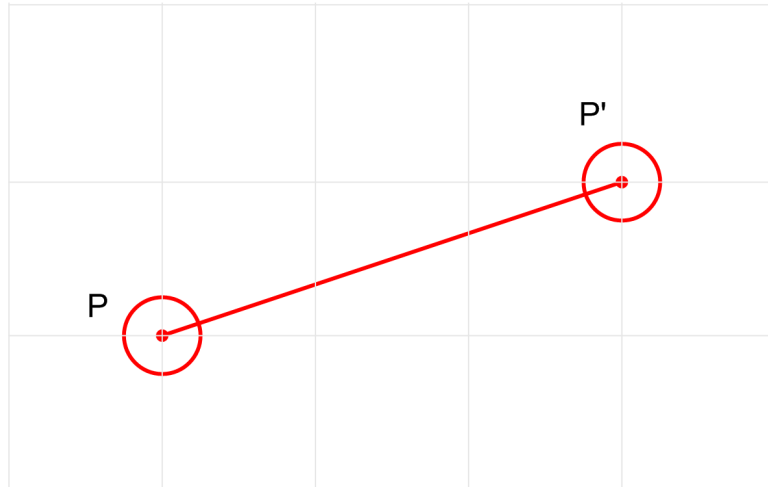


Figure 2.3: Euclidean distance is the straight line distance between two points.

Example: For our test network setup, the Euclidean distance between S and S' is:

$$\begin{aligned}
\mu_{euclidean}(S, S') &= \frac{1}{N} \sum_{i=1}^N \left[(x'_i - x_i)^2 + (y'_i - y_i)^2 \right]^{1/2} \\
&= \frac{1}{2} \left(\left[(x'_1 - x_1)^2 + (y'_1 - y_1)^2 \right]^{1/2} + \left[(x'_2 - x_2)^2 + (y'_2 - y_2)^2 \right]^{1/2} \right) \\
&= \frac{1}{2} \left(\left[(1 - 1)^2 + (2 - 3)^2 \right]^{1/2} + \left[(4 - 3)^2 + (3 - 3)^2 \right]^{1/2} \right) \\
&= \frac{1}{2} \left(\left[(0)^2 + (-1)^2 \right]^{1/2} + \left[(1)^2 + (0)^2 \right]^{1/2} \right) \\
&= \frac{1}{2} \left([0 + 1]^{1/2} + [1 + 0]^{1/2} \right) \\
&= \frac{1}{2} (1 + 1) \\
&= 1
\end{aligned}$$

2.3.1.2 Hamming (Manhattan) Distance

Hamming (Manhattan) distance is a popular metric due to its simplicity and its dependence on a two dimensional coordinate system. It is the distance between two points 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 arbitrary direction as in the case of Euclidean distance), it measures the distance to get to one point from the other. For sensor network topologies, similar to Euclidean distance, this metric has been applied to each individual node position and the average error has been reported as the overall error of the localization algorithm.

Formulation:

$$\mu_{hamming}(S, S') = \frac{1}{N} \sum_{i=1}^N \left(\left| x'_i - x_i \right| + \left| y'_i - y_i \right| \right) \quad (2.2)$$

where S and S' contain N points and x_i and y_i are the actual coordinates of node i while x'_i and y'_i are the estimated coordinates of the node i .

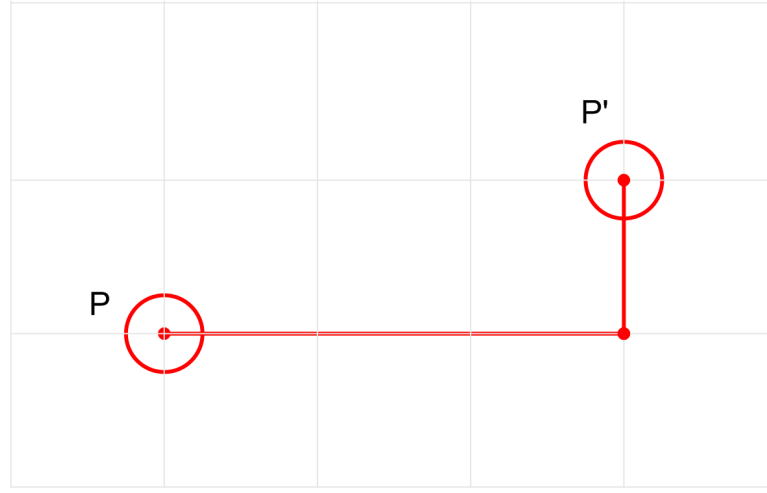


Figure 2.4: Hamming distance is the distance between two points measured along axes at right angles.

Example: For our test network setup, the Hamming distance between S and S' is:

$$\begin{aligned}
 \mu_{\text{hamming}}(S, S') &= \frac{1}{N} \sum_{i=1}^N (|x'_i - x_i| + |y'_i - y_i|) \\
 &= \frac{1}{2} \sum_{i=1}^2 (|x'_i - x_i| + |y'_i - y_i|) \\
 &= \frac{1}{2} [|x'_1 - x_1| + |y'_1 - y_1| + |x'_2 - x_2| + |y'_2 - y_2|] \\
 &= \frac{1}{2} [|1 - 1| + |2 - 3| + |4 - 3| + |3 - 3|] \\
 &= \frac{1}{2} [|0| + |-1| + |1| + |0|] \\
 &= \frac{1}{2} [0 + 1 + 1 + 0] \\
 &= 1
 \end{aligned}$$

2.3.1.3 Tanimoto Coefficients and Tanimoto Distance

The Tanimoto coefficient (TC) is a more complex metric that considers vectors rather than points. It is a highly popular metric in text matching problems of

information retrieval. It is defined as the size of the intersection divided by the size of the union of the sample sets. The interpretation in our domain is then as follows. To find the coefficient, we first get the relative position of points in both sets as vectors and then move these vectors to have their first points at the origin. We then compute Tanimoto coefficient for these vectors. For each pair of nodes P_1 and P_2 in S ,

$$TC(P_1, P_2) = \vec{V} \cdot \vec{V}' / \left(|\vec{V}|^2 + |\vec{V}'|^2 - \vec{V} \cdot \vec{V}' \right) \quad (2.3)$$

is the Tanimoto coefficient for the node P_1 and P_2 , where \vec{V} is the vector that combines the actual positions of the nodes (i.e. $\vec{V} = P_1 P_2$), and \vec{V}' is the vector that combines the estimated positions of the nodes (i.e. $\vec{V}' = P_1' P_2'$).

Tanimoto coefficient, in fact, measures the similarity of topologies while metrics are expected to measure the distance. In Section 4, behaviors of the metrics are discussed. In order to make comparison of results sounder, we introduce Tanimoto distance so that it gives a distance value from a Tanimoto coefficient. The measure of distance is derived by subtracting the computed similarity from the measure of perfect similarity. Then we scaled the distance by 10 to make its value comprehensible for the experiments we conduct, in which errors up to the magnitude 10 are introduced. As a result,

$$d(P_1, P_2) = \left(1 - \vec{V} \cdot \vec{V}' / \left(|\vec{V}|^2 + |\vec{V}'|^2 - \vec{V} \cdot \vec{V}' \right) \right) * 10 \quad (2.4)$$

gives Tanimoto distance for the pair of nodes P_1 and P_2 .

Then

$$\mu_{tanimoto}(S, S') = \frac{2}{N * (N - 1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right) \quad (2.5)$$

is the Tanimoto distance between the set S and its estimate S' .

Example: For our test network setup, we can compute the Tanimoto distance of S and S' as:

$$d(P_1, P_2) = (1 - \vec{V} \cdot \vec{V}' / (|\vec{V}|^2 + |\vec{V}'|^2 - |\vec{V} \cdot \vec{V}'|)) * 10$$

where $\vec{V} = P_1\vec{P}_2$ and $\vec{V}' = P_1'\vec{P}_2'$

$$\begin{aligned} |\vec{V}| &= (2^2 + 0^2)^{1/2} = 2 \\ |\vec{V}'| &= (3^2 + 1^2)^{1/2} \cong 3.16 \\ \vec{V} \cdot \vec{V}' &= |\vec{V}| * |\vec{V}'| * \cos(\alpha) = 3.16 * 2 * (3/3.16) = 6 \\ d(P_1, P_2) &= (1 - \vec{V} \cdot \vec{V}' / (|\vec{V}|^2 + |\vec{V}'|^2 - |\vec{V} \cdot \vec{V}'|)) * 10 \\ d(P_1, P_2) &= (1 - 6 / ((3.16)^2 + 2^2 - 6)) * 10 \\ d(P_1, P_2) &= (1 - 6/8) * 10 \\ d(P_1, P_2) &= 2.5 \\ \mu_{tanimoto}(S, S') &= \frac{2}{N * (N - 1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right) \end{aligned}$$

then

$$\begin{aligned} &= \frac{2}{2 * (2 - 1)} \left(\sum_{i=1}^2 \sum_{j=i+1}^2 d(P_i, P_j) \right) \\ &= \frac{2}{2 * 1} (d(P_1, P_2)) \\ &= \frac{2}{2 * 1} (2.5) \\ &= 2.5 \end{aligned}$$

2.3.1.4 Cosine Similarity and Cosine Distance

When we consider approaches that consider not only a single node but more nodes at the same time, Cosine similarity (CS) is another well-known technique. In this case we can consider two nodes with their actual and estimated positions as two vectors, as shown in Figure 2.5. In the figure, the actual positions of two nodes P_1 and P_2 are represented by the solid circles and the vector that combines these two positions $\vec{V} = P_1\vec{P}_2$ is shown by the solid edge that can be used to define their actual relative positioning difference in the deployment area. The dashed circles represent the estimated positions of these nodes and the dashed edge in between represents the vector $\vec{V}' = P_1'\vec{P}_2'$ which can be used to define estimated relative positioning. Cosine similarity then can be used to define the angle between these vectors. For instance, if the vectors \vec{V} and \vec{V}' are parallel, then Cosine similarity would suggest that the two sets of topologies were perfectly similar.

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 estimates. The distance between the nodes is, however, not captured by this metric.

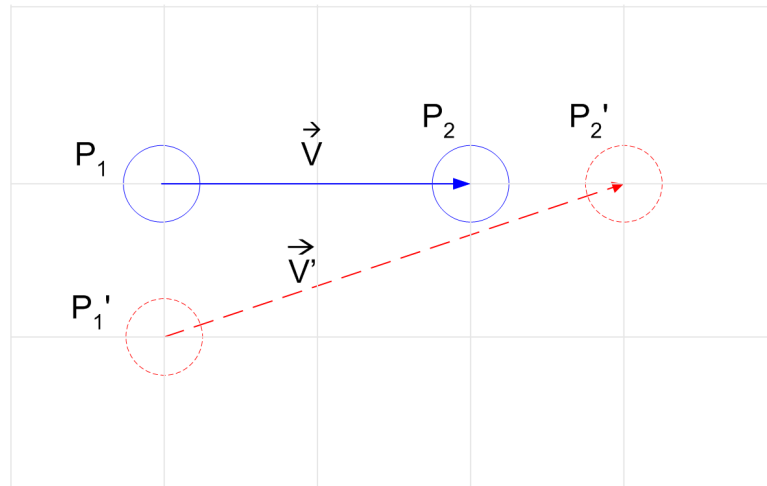


Figure 2.5: Two sensor node positions P_1 and P_2 are shown with solid circles, with the edge between them describing their actual relative positioning. P_1' and P_2' are the position estimates produced by these nodes and the dashed edge between them is used to define their relative positioning based on the estimated positions.

Formulation: For each pair of nodes P_1 and P_2 in S ,

$$CS(P_1, P_2) = \vec{V} \cdot \vec{V}' / \left(|\vec{V}| \times |\vec{V}'| \right) \quad (2.6)$$

is Cosine similarity for the node P_1 and P_2 , where \vec{V} is the vector that combines the actual positions of the nodes (i.e. $\vec{V} = P_1 \vec{P}_2$), and \vec{V}' is the vector that combines the estimated positions of the nodes (i.e. $\vec{V}' = P_1' \vec{P}_2'$).

Similar to Tanimoto coefficient, Cosine similarity, too, measures the similarity of topologies. As discussed in Tanimoto coefficient topic, metrics need to be distance, hence we introduced Cosine distance such that it produces distance values from Cosine similarity. As done previously, the measure of distance is derived by subtracting the computed similarity from the measure of perfect similarity and scaling it by 10. As a result,

$$d(P_1, P_2) = \left(1 - \vec{V} \cdot \vec{V}' / \left(|\vec{V}| \times |\vec{V}'| \right) \right) * 10 \quad (2.7)$$

gives Cosine distance for the pair of nodes P_1 and P_2 .

Then

$$\mu_{cosine}(S, S') = \frac{2}{N * (N - 1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right) \quad (2.8)$$

gives Cosine distance between the set S and its estimate S' .

Example: For our test network setup, we can compute the Cosine distance of S and S' as:

$$d(P_1, P_2) = \left(1 - \vec{V} \cdot \vec{V}' / \left(|\vec{V}| \times |\vec{V}'| \right) \right) * 10$$

where $\vec{V} = P_1\vec{P}_2$ and $\vec{V}' = P_1'\vec{P}_2'$

$$\begin{aligned}
|\vec{V}| &= (2^2 + 0^2)^{1/2} = 2 \\
|\vec{V}'| &= (3^2 + 1^2)^{1/2} \cong 3.16 \\
\vec{V} \cdot \vec{V}' &= |\vec{V}| * |\vec{V}'| * \cos(\alpha) = 3.16 * 2 * (3/3.16) = 6 \\
d(P_1, P_2) &= (1 - \vec{V} \cdot \vec{V}' / (|\vec{V}| * |\vec{V}'|)) * 10 \\
d(P_1, P_2) &= (1 - 6 / ((3.16) * 2)) * 10 \\
d(P_1, P_2) &= (1 - 6/6.32) * 10 \\
d(P_1, P_2) &= 0.51
\end{aligned}$$

then

$$\begin{aligned}
\mu_{\cosine}(S, S') &= \frac{2}{N * (N - 1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right) \\
&= \frac{2}{2 * (2 - 1)} \left(\sum_{i=1}^2 \sum_{j=i+1}^2 d(P_i, P_j) \right) \\
&= \frac{2}{2 * 1} (d(P_1, P_2)) \\
&= \frac{2}{2 * 1} (0.51) \\
&= 0.51
\end{aligned}$$

So far, we have presented some existing metrics which deal with the measure of distance between two topologies. We have described the metrics in detail: we gave their descriptions, wrote down the formulations, and then estimated the distance between two basic topologies by the presented metrics. We will talk about these metrics again after we have put forward some other metrics, those proposed in this work. Then all metrics will be evaluated through simulations, and then the results which indicate the metric characteristics will be revealed.

Chapter 3

Proposed Work

In this chapter we propose new metrics that can be used to measure the distance between the actual and estimated topologies, S and S' . We focus on different approaches that can be used to evaluate localization errors. We provide some new metrics developed using these different approaches and give the details of each metric.

3.1 The Proposed Metrics

Here, we present some novel metrics we came up with during the course of our study to address the issues we have raised. The proposed metrics are summarized in Table 3.1.

3.2 Metric Details

As we did in Section 2.3.1, for each metric we first give its description followed by its formulation and then provide the motivation behind it (if exists) and finally run it on an example network. The example network is the network setup shown in Figure 3.1, where the actual network S and its estimation S' are:

Name:	Formulation (For N Points):
Cumulative Vectorial Distance	$\mu(S, S') = \frac{1}{N} \left(\left[\sum_{i=1}^N (x'_i - x_i) \right]^2 + \left[\sum_{i=1}^N (y'_i - y_i) \right]^2 \right)^{1/2}$
Relative Euclidean Distance	<p>For each pair of nodes P_1 and P_2 in S,</p> $d(P_1, P_2) = \left[(\vec{V} ^2 + \vec{V}' ^2 - 2\vec{V} \cdot \vec{V}') \right]^{1/2}$ <p>where $\vec{V} = P_1P_2$ and $\vec{V}' = P'_1P'_2$ then</p> $\mu(S, S') = \frac{2}{N*(N-1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right)$
Normalized Relative Euclidean Distance	<p>For each pair of nodes P_1 and P_2 in S,</p> $d(P_1, P_2) = \left[(\vec{V} ^2 + \vec{V}' ^2 - 2\vec{V} \cdot \vec{V}') \right]^{1/2} / (\vec{V} + \vec{V}')$ <p>where $\vec{V} = P_1P_2$ and $\vec{V}' = P'_1P'_2$ then</p> $\mu(S, S') = \frac{2}{N*(N-1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right)$
Spring Distance	<p>For each pair of nodes P_1 and P_2 in S,</p> $d(P_1, P_2) = 1/ \vec{V} * \left \vec{V} - \vec{V}' \right $ $+ 1/ \vec{V} * (dV_{s1} + dV_{s2}) * m$ $+ 1/ \vec{V} * \left[\vec{V} ^2 + \vec{V}' ^2 - 2\vec{V} \cdot \vec{V}' \right]^{1/2} * n$ <p>where $d\vec{V}_{s1} = P_1\vec{P}'_1$, $d\vec{V}_{s2} = P_2\vec{P}'_2$, $\vec{V} = P_1\vec{P}_2$, $\vec{V}' = P'_1\vec{P}'_2$, m is shift_sensitivity and n is rotation_sensitivity, then</p> $\mu(S, S') = \frac{2}{N*(N-1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right)$

Table 3.1: List of the metrics developed in study of this thesis. Details of the metrics are given in Section 3.2.

$$S = \{P_1(1, 3), P_2(3, 3)\} \text{ and } S' = \{P'_1(1, 2), P'_2(4, 3)\}$$

For this network, the vectors $\vec{V} = P_1\vec{P}_2$ and $\vec{V}' = P'_1\vec{P}'_2$ hold the relative positions of the nodes. Moving these vectors to origin, we get the setup in Figure 3.2.

3.2.1 Cumulative Vectorial Distance (CVD)

In this metric we propose, we thought about a way to include the distance as well as direction into the equation. In this regard, we record the distance between

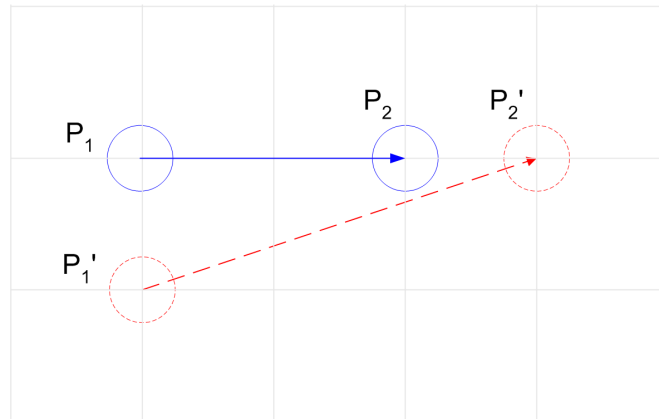


Figure 3.1: Test network setup with $P_1(1,3), P_2(3,3); P_1'(1,2), P_2'(4,3)$.

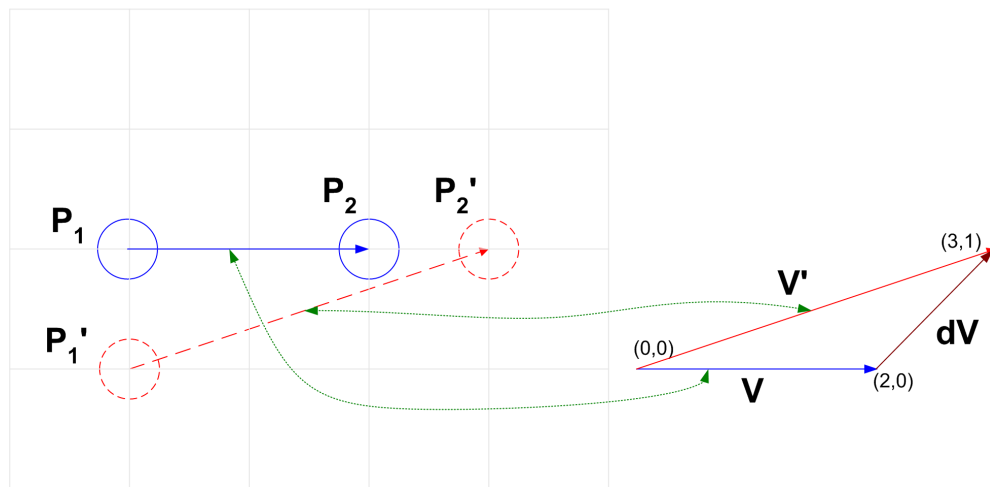


Figure 3.2: The vectors representing relative positions of the nodes. All examples are based on this network setup.

a real and corresponding estimated point as a vector. Then for all points in the network, we sum up these vectors to form the cumulative vector, which is recorded as the measure of distance. In particular,

$$\mu_{cvd}(S, S') = \frac{1}{N} \left| \sum_{i=1}^N \vec{V}_i \right| \quad \text{where } \vec{V}_i = P_i \vec{P}'_i \quad (3.1)$$

we know that

$$|\vec{V}_i + \vec{V}_j| = \left| \left((x'_i - x_i) + (x'_j - x_j) \right), \left((y'_i - y_i) + (y'_j - y_j) \right) \right| \quad (3.2)$$

$$= \sqrt{\left[(x'_i - x_i) + (x'_j - x_j) \right]^2 + \left[(y'_i - y_i) + (y'_j - y_j) \right]^2} \quad (3.3)$$

Substituting (3.3) into (3.1) we get

$$\mu_{cvd}(S, S') = \frac{1}{N} \left(\left[\sum_{i=1}^N (x'_i - x_i) \right]^2 + \left[\sum_{i=1}^N (y'_i - y_i) \right]^2 \right)^{1/2} \quad (3.4)$$

where the topologies S and S' contain N points.

Example: For the example topology shown in Figure 3.2, the distance can be computed using this method as:

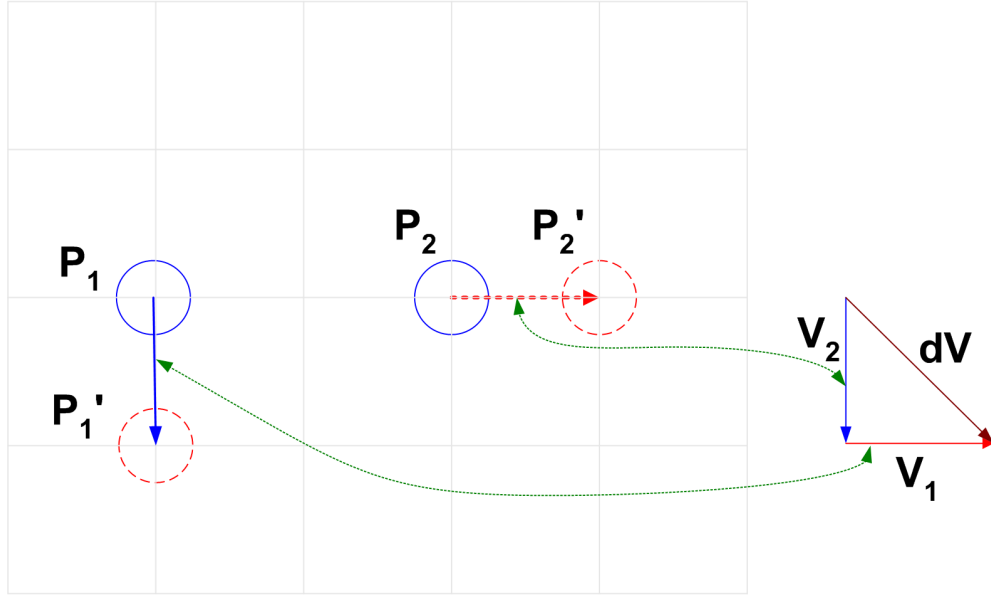


Figure 3.3: The distance between pair (P_1, P_1') and (P_2, P_2') are recorded as vectors V_1 and V_2 . Then, by adding these vectors we get distance dV representing the distance between two topologies.

$$\begin{aligned}
 \mu_{cud}(S, S') &= \frac{1}{N} \left(\left[\sum_{i=1}^N (x'_i - x_i) \right]^2 + \left[\sum_{i=1}^N (y'_i - y_i) \right]^2 \right)^{1/2} \\
 &= \frac{1}{2} \left(\left[\sum_{i=1}^2 (x'_i - x_i) \right]^2 + \left[\sum_{i=1}^2 (y'_i - y_i) \right]^2 \right)^{1/2} \\
 &= \frac{1}{2} \left(\left[(x'_1 - x_1) + (x'_2 - x_2) \right]^2 + \left[(y'_1 - y_1) + (y'_2 - y_2) \right]^2 \right)^{1/2} \\
 &= \frac{1}{2} \left([(1 - 1) + (4 - 3)]^2 + [(2 - 3) + (3 - 3)]^2 \right)^{1/2} \\
 &= \frac{1}{2} \left([0 + 1]^2 + [-1 + 0]^2 \right)^{1/2} \\
 &= \frac{1}{2} (1 + 1)^{1/2} \\
 &= \frac{1}{2} (2)^{1/2} \\
 &= 0,71
 \end{aligned}$$

3.2.2 Relative Euclidean Distance

Relative Euclidean Distance (RED) is a metric that we propose based on our observations on how Euclidean distance fails to capture the relative positions of a pair of nodes. Euclidean distance considers a point in reference to the origin which is a fixed point. With this metric, instead, we try to capture the relative positional difference between two sets of points: the actual positions set and the estimated positions set. We consider the positions in pairs. Each pair of positions (i.e. points) in a set is represented with a vector.

Considering two such points P_1 and P_2 , we first get the relative position of points in both sets as vectors and then move these vectors to have their starting point at the origin. We then compute the Euclidean distance between the end points of these two vectors. The process is illustrated in Figure 3.4. Note that RED, unlike Euclidean distance, allows directional errors to be caught as well as distance errors.

In particular, for each pair of nodes P_1 and P_2 in S , let $\vec{V} = P_1\vec{P}_2$ and $\vec{V}' = P_1'\vec{P}_2'$ then

$$d(P_1, P_2) = \left[\left(|\vec{V}|^2 + |\vec{V}'|^2 - 2\vec{V} \cdot \vec{V}' \right) \right]^{1/2}$$

Hence

$$\mu_{red}(S, S') = \frac{2}{N * (N - 1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right) \quad (3.5)$$

gives RED between the set S and its estimate S' .

Example:

$$d(P_1, P_2) = \left[\left(|\vec{V}|^2 + |\vec{V}'|^2 - 2\vec{V} \cdot \vec{V}' \right) \right]^{1/2}$$

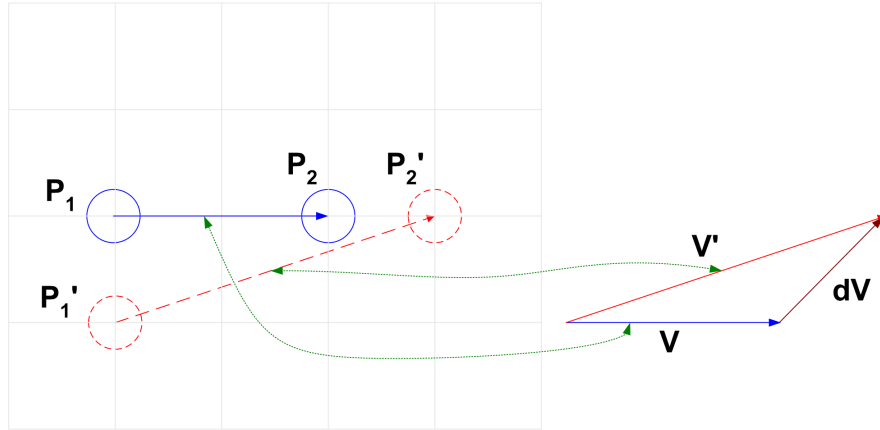


Figure 3.4: The distance between pair (P_1, P_2) and (P'_1, P'_2) are vectors V and V' . Relative Euclidean Distance (dV) is the distance between two vectors' end points when combined at a common starting point.

where $\vec{V} = P_1\vec{P}_2$ and $\vec{V}' = P'_1\vec{P}'_2$

$$\begin{aligned}
 |\vec{V}| &= (2^2 + 0^2)^{1/2} = 2 \\
 |\vec{V}'| &= (3^2 + 1^2)^{1/2} \cong 3.16 \\
 \vec{V} \cdot \vec{V}' &= |\vec{V}| * |\vec{V}'| * \cos(\alpha) = 3.16 * 2 * (3/3.16) = 6 \\
 d(P_1, P_2) &= [(|\vec{V}|^2 + |\vec{V}'|^2 - 2\vec{V} \cdot \vec{V}')^{1/2}] \\
 d(P_1, P_2) &= [(3.16)^2 + 2^2 - 2 * 6]^{1/2} \\
 d(P_1, P_2) &= 1.41
 \end{aligned}$$

then

$$\begin{aligned}
\mu_{red}(S, S') &= \frac{2}{N * (N - 1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right) \\
&= \frac{2}{2 * (2 - 1)} \left(\sum_{i=1}^2 \sum_{j=i+1}^2 d(P_i, P_j) \right) \\
&= \frac{2}{2 * 1} (d(P_1, P_2)) \\
&= \frac{2}{2 * 1} (d(P_1, P_2)) \\
&= \frac{2}{2 * 1} (1.41) \\
&= 1.41
\end{aligned}$$

3.2.3 Normalized Relative Euclidean Distance

Normalized Relative Euclidean Distance (NRED) is another metric we came up with based on our observations on previous techniques and sensor network application requirements. In this approach we start off as RED and then normalize the distance according to the length of the vectors. This is done by dividing the distance by the sum of vector magnitudes.

NRED is motivated by two observations. First one is that the topology is not just about the distance of individual points, but it is more about the relative position of the pairs that compose the network. Second one is that, direct distances may be misleading i.e. 10001 - 10003 and 1-3 both pairs have the same direct distance while 10001-10003 is closer in topological view.

For each pair of nodes P_1 and P_2 in S ,

$$d(P_1, P_2) = \left[\left(|\vec{V}|^2 + |\vec{V}'|^2 - 2\vec{V} \cdot \vec{V}' \right) \right]^{1/2} / (|\vec{V}| + |\vec{V}'|)$$

where \vec{V} is the vector that combines the actual positions of the nodes (i.e.

$\vec{V} = P_1\vec{P}_2$) and \vec{V}' is the vector that combines the estimated positions of the nodes (i.e. $\vec{V}' = P_1'\vec{P}_2'$).

Then

$$\mu_{nred}(S, S') = \frac{2}{N * (N - 1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right) \quad (3.6)$$

gives NRED between the set S and its estimate S' .

Example:

$$d(P_1, P_2) = \left[(|\vec{V}|^2 + |\vec{V}'|^2 - 2\vec{V} \cdot \vec{V}') \right]^{1/2} / (|\vec{V}| + |\vec{V}'|)$$

where $\vec{V} = P_1\vec{P}_2$ and $\vec{V}' = P_1'\vec{P}_2'$

$$\begin{aligned} |\vec{V}| &= (2^2 + 0^2)^{1/2} = 2 \\ |\vec{V}'| &= (3^2 + 1^2)^{1/2} \cong 3.16 \\ \vec{V} \cdot \vec{V}' &= |\vec{V}| * |\vec{V}'| * \cos(\alpha) = 3.16 * 2 * (3/3.16) = 6 \\ d(P_1, P_2) &= \left[(|\vec{V}|^2 + |\vec{V}'|^2 - 2\vec{V} \cdot \vec{V}') \right]^{1/2} / (|\vec{V}| + |\vec{V}'|) \\ d(P_1, P_2) &= [(3.16)^2 + 2^2 - 2 * 6]^{1/2} / (3.16 + 2) \\ d(P_1, P_2) &= 1.41/5.16 \\ d(P_1, P_2) &= 0.27 \end{aligned}$$

then

$$\begin{aligned}
\mu_{\text{red}}(S, S') &= \frac{2}{N * (N - 1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right) \\
&= \frac{2}{2 * (2 - 1)} \left(\sum_{i=1}^2 \sum_{j=i+1}^2 d(P_i, P_j) \right) \\
&= \frac{2}{2 * 1} (d(P_1, P_2)) \\
&= \frac{2}{2 * 1} (0.27) \\
&= 0.27
\end{aligned}$$

3.2.4 Spring Distance

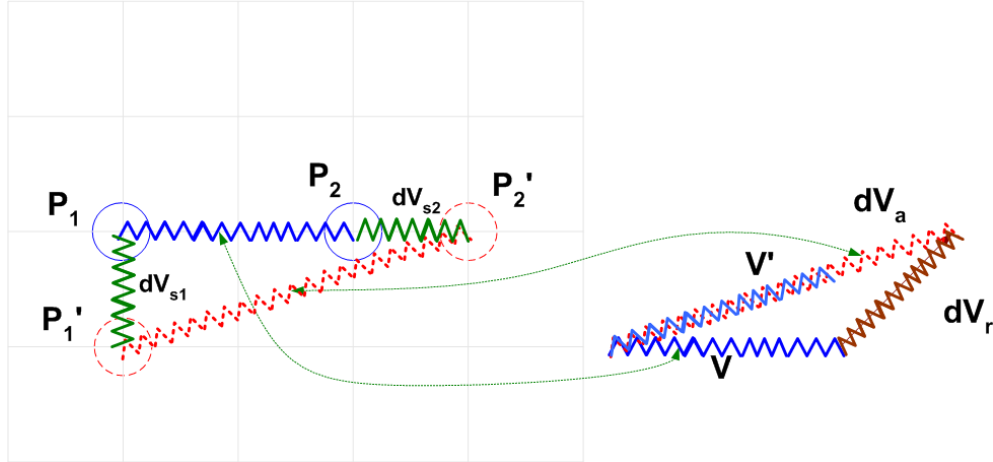


Figure 3.5: The vectors $\vec{V} : (P_1, P_2)$, $\vec{V}' : (P_1', P_2')$, $d\vec{V}_{s1} : (P_1, P_1')$ and $d\vec{V}_{s2} : (P_2, P_2')$ are made up of strings. $d\vec{V}_a$ which equals to $|\vec{V}'| - |\vec{V}|$ is the distance observed as absolute change. $d\vec{V}_r$ which equals to $|\vec{V}' - \vec{V}|$ is the distance observed as rotation change. $d\vec{V}_{s1}$ and $d\vec{V}_{s2}$ are the distance observed as shift changes.

Generally, the more force applied to a system, the more changes occur on it. Hence, having a physical model for a network and measuring the changes on it via the amount of the force, which is applied to make such a change on it, motivated us for this metric. We assumed that the vector representing the actual relative placement of two points is made out of a spring that we try to keep in the

same state after the estimates are completed. That is, if after the estimates are done, assuming perfect accuracy, there would not be any change in the new vector between the estimated coordinates. Yet, if the estimates are further apart than they should be, this means we have applied a force to stretch the spring. Similarly, if they are much closer to each other, that means we applied a force to compress the spring. With this motivation, we propose the spring distance measure as follows. The distance between two vectors, one representing the actual relative placement of two points and the other the estimated relative placement of these two points can be measured by the force applied to alter the position/spring.

Note that if all nodes are connected to each other with springs, moving the complete plane on which the topology resides would not flag any error. For this reason, we also assumed that each node is connected to the ground and axes by another two sets of springs such that absolute relocations and rotations can also be recorded. In order to accommodate whole changes, we assumed three sets of strings attached to each node.

The strings in the first set which connect nodes to each other are responsive for changes in relative distances and these strings' force constant is assumed to be one. In addition, the strings in the second set, which connect node positions to the ground are responsive for changes in absolute distances, so these strings' force constant is proportional to `shift_sensitivity` parameter. Moreover, the string in the third set, which connects node positions to axes are responsive for changes in direction, so these string's force constant is proportional to `rotation_sensitivity` parameter. In particular, we assume that the sensor network consists of nodes connected to each other with springs and also that each one is connected to the ground and axes with springs. We then calculated the force applied to these springs to end up in the topology suggested by the estimated positions.

The forces applied on the springs are then measured using Hook's law:

$$F = -\frac{c}{\lambda}x$$

where F is the restoring force exerted by the spring, c is the spring constant

or force constant of the spring, λ is the length of string, and x is the distance the spring is elongated by.

Name:	shift_sensitivity constant	rotational_sensitivity constant
Spring A Distance	0.5	0.5
Spring B Distance	1	0
Spring C Distance	0	1
Spring D Distance	0	0

Table 3.2: List of Spring Distance variations.

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, as listed in Table 3.2 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. Computed force on the strings quantifies the change in the network and is used as the distance between networks.

Formulation: For each pair of nodes P_1 and P_2 in S ,

$$\begin{aligned}
d(P_1, P_2) &= 1/|\vec{V}| * \left| |\vec{V}| - |\vec{V}'| \right| \\
&+ 1/|\vec{V}| * (|dV_{s1}| + |dV_{s2}|) * m \\
&+ 1/|\vec{V}| * \left[|\vec{V}|^2 + |\vec{V}'|^2 - 2\vec{V} \cdot \vec{V}' \right]^{1/2} * n
\end{aligned}$$

where $d\vec{V}_{s1} = P_1\vec{P}'_1$, $d\vec{V}_{s2} = P_2\vec{P}'_2$, $\vec{V} = P_1\vec{P}_2$, $\vec{V}' = P_1\vec{P}'_2$, m is shift_sensitivity and n is rotation_sensitivity, then

$$\mu_{spring}(S, S') = \frac{2}{N * (N - 1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right)$$

is Spring distance between the topologies S and S' .

Example:

$$\begin{aligned}
 d(P_1, P_2) &= 1/|\vec{V}| * \left| |\vec{V}| - |\vec{V}'| \right| \\
 &+ 1/|\vec{V}| * (|dV_{s1}| + |dV_{s2}|) * m \\
 &+ 1/|\vec{V}| * \left[|\vec{V}|^2 + |\vec{V}'|^2 - 2\vec{V} \cdot \vec{V}' \right]^{1/2} * n
 \end{aligned}$$

where $d\vec{V}_{s1} = P_1\vec{P}'_1$, $d\vec{V}_{s2} = P_2\vec{P}'_2$, $\vec{V} = P_1\vec{P}_2$, $\vec{V}' = P'_1\vec{P}'_2$, m is shift_sensitivity and n is rotation_sensitivity,

$$\begin{aligned}
 |d\vec{V}_{s1}| &= 1 \\
 |d\vec{V}_{s2}| &= 1 \\
 |\vec{V}| &= (2^2 + 0^2)^{1/2} = 2 \\
 |\vec{V}'| &= (3^2 + 1^2)^{1/2} \cong 3.16 \\
 \vec{V} \cdot \vec{V}' &= |\vec{V}| * |\vec{V}'| * \cos(\alpha) = 3.16 * 2 * (3/3.16) = 6 \\
 d(P_1, P_2) &= 1/|\vec{V}| * \left| |\vec{V}| - |\vec{V}'| \right| \\
 &+ 1/|\vec{V}| * (|dV_{s1}| + |dV_{s2}|) * m \\
 &+ 1/|\vec{V}| * \left[|\vec{V}|^2 + |\vec{V}'|^2 - 2\vec{V} \cdot \vec{V}' \right]^{1/2} * n \\
 d(P_1, P_2) &= 1/2 * |2 - 3.16| + 1/2 * (1 + 1) * 0.5 \\
 &+ 1/2 * [(2^2 + (3.16)^2 - 2 * 6)]^{1/2} * 0.5 \\
 d(P_1, P_2) &= 0.5 * 1.16 + 0.5 * 2 * 0.5 + 0.5 * 1.41 * 0.5 \\
 d(P_1, P_2) &= 0.58 + 0.5 + 0.35 \\
 d(P_1, P_2) &= 1.43
 \end{aligned}$$

then

$$\begin{aligned}
&= \mu_{spring}(S, S') = \frac{2}{N * (N - 1)} \left(\sum_{i=1}^N \sum_{j=i+1}^N d(P_i, P_j) \right) \\
&= \frac{2}{2 * (2 - 1)} \left(\sum_{i=1}^2 \sum_{j=i+1}^2 d(P_i, P_j) \right) \\
&= \frac{2}{2 * 1} (d(P_1, P_2)) \\
&= \frac{2}{2 * 1} (d(P_1, P_2)) \\
&= \frac{2}{2 * 1} (1.43) \\
&= 1.43
\end{aligned}$$

This section concludes the introduction and detailed description of existing metrics and new metrics that we proposed. We have proposed four new distance metrics. The distance metrics we described can be used to evaluate localization algorithms. This is a critical issue. How a metric can help in evaluating various localization algorithms may depend on the application scenario and on the type of errors that are tolerable by the users of the location data. Therefore, metrics too need to be evaluated. In the next section, we will evaluate the various existing and new metrics we discussed in the thesis. We will try to identify the cases and circumstances under which a particular metric is more useful compared to other ones.

Chapter 4

Experiments and Results

In this chapter we will study some basic topology scenarios for comparing the metrics presented in the previous chapters. For each topology and metric we will discuss the impact of errors and the expected accuracy values for some sample applications.

4.1 Topology Scenarios

We run simulations for three scenarios. Each of these scenarios represents a certain topological change. Simulation inputs are:

- S (Actual Network),
- S' (Estimated Network),
- M (Set of metrics).

Here, actual network S is deployed randomly with certain densities. Estimated network S' is generated by localization functions.

Following localization functions are used for simulation scenarios and each of them is considered as a topological change.

1. $\delta_{shift(n)} : \forall i, p_i \in S \wedge p'_i \in S' \rightarrow p'_{i_x} = p_{i_x} + n \wedge p'_{i_y} = p_{i_y} + n$
2. $\delta_{rotate(\alpha)} : \forall i, p_i \in S \wedge p'_i \in S' \rightarrow p'_{i_x} = rotatepoint_\alpha(p_{i_x}) \wedge p'_{i_y} = rotatepoint_\alpha(p_{i_y})$ where $rotatepoint_\alpha(k)$ rotates point k around center point of S by α degrees and returns the resulting point.
3. $\delta_{arbitrary(n)} : \forall i, p_i \in S \wedge p'_i \in S' \rightarrow p'_{i_x} = p_{i_x} \pm n \wedge p'_{i_y} = p_{i_y} \pm n$ where value of \pm determined randomly.

The metric list, M , consists of the metrics in the literature and the metrics developed by us. Therefore, the list consists of:

- Euclidean Distance,
- Hamming (Manhattan) Distance,
- Tanimoto Distance,
- Cosine Distance,
- CVD,
- RED,
- NRED,
- Spring Distance.

4.2 Simulation Parameters

The simulations are performed over 20×20 unit sized square area. For each density, 100% (400 nodes), 90% (360 nodes) down to 10% (40 nodes), nodes are randomly distributed over this area. In this manner, 10 networks (actual network S) are generated. Then, each localization function is applied to these networks, as a result we get 10 networks (estimated network S') for each localization algorithm.

Subsequently, each metric is applied to each pair (S, S') of these networks and the results are recorded.

4.3 Simulation Environment

In the community of wireless sensor networks, proposals are usually supported by simulation. As a result of this, many simulation tools specialized to wireless sensor networks are available in public domain i.e. NS-2 [9]. These simulation tools contain many features, and reduce the overhead of rewriting numerous well-known algorithms and protocols. However, for simulations we designed, there is no need for such complex simulators. We just deploy sensor networks, then apply localization algorithms to compute estimated networks, and then execute a set of metrics on the networks. Therefore, we designed and implemented a simulation program with visual support which is specialized to this work and focused on our needs. The program randomly deploys a network, applies estimation algorithms to generate estimated networks, and computes the distances based on a given set of metrics.

4.4 Evaluation Based on the Metrics

In this section, we show the simulation results classified by metrics and try to figure out metrics behavior against the changes in the network.

4.4.1 Euclidean Distance Behavior

In Figure 4.1(a), the metric value is drawn against the distortion amount, increasing from 1 to 10. Having more distorted topology, the metric suggests more error, thus Euclidean distance is distortion sensitive and it is linearly proportional to the magnitude of distortion. The metric value is drawn against the shift amount increasing from 1 to 10 in Figure 4.1(b). Here, the metric value is proportional

to shift amount, thus we conclude that Euclidean distance demonstrates similar behavior for shifted topologies and for distorted topologies. In case of rotated topologies, presented in Figure 4.1(c) where metric value is drawn against the rotation, Euclidean distance reports greater error when angle increases from 0 to 180 and smaller error when angle increases from 180 to 360.

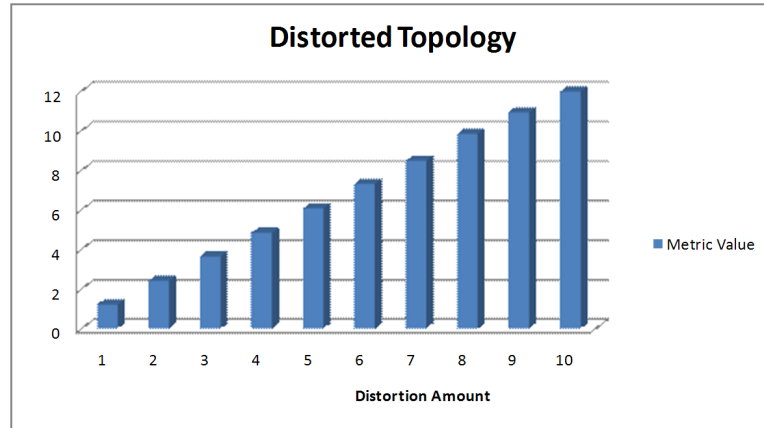
Simulation results show that Euclidean distance metric does not tolerate topology preserving localization errors. Especially, the similarity in its behavior against distorted and shifted topologies illustrates how it is unaware of network topology. The behavior of Euclidean distance metric can be stated as:

- Shift Sensitive,
- Rotation Sensitive.

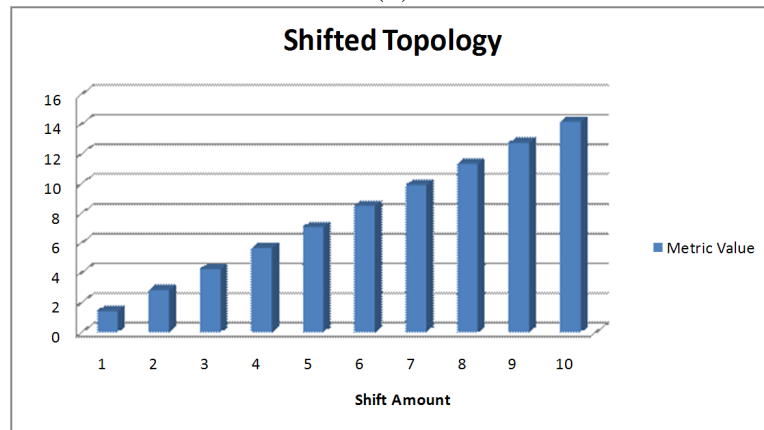
4.4.2 Hamming (Manhattan) Distance Behavior

Figures 4.2(a), (b) and (c) show us that the Hamming distance is distortion, shift and rotation sensitive. Comparing Figure 4.2(a) and (b), we find out that the metric behavior is similar for distorted and shifted topologies. Simulation results indicate that Hamming distance metric does not tolerate topology preserving localization errors. Especially, the similarity in its behavior against distorted and shifted topologies illustrates how it is unaware of network topology, similar to Euclidean distance. The behavior of Hamming distance metric can be declared as:

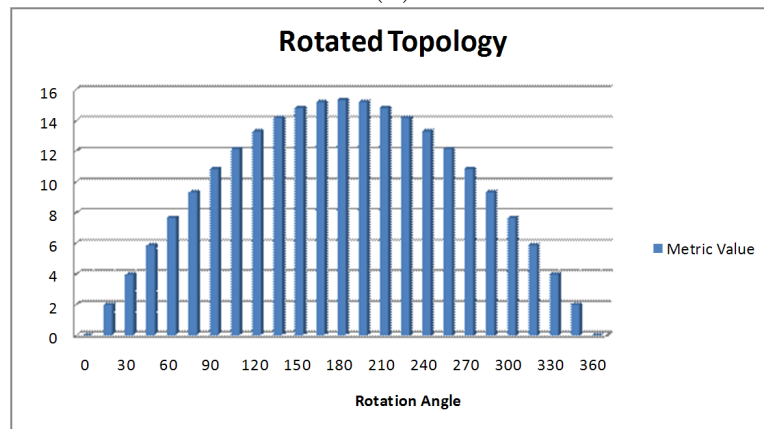
- Shift Sensitive,
- Rotation Sensitive.



(a)

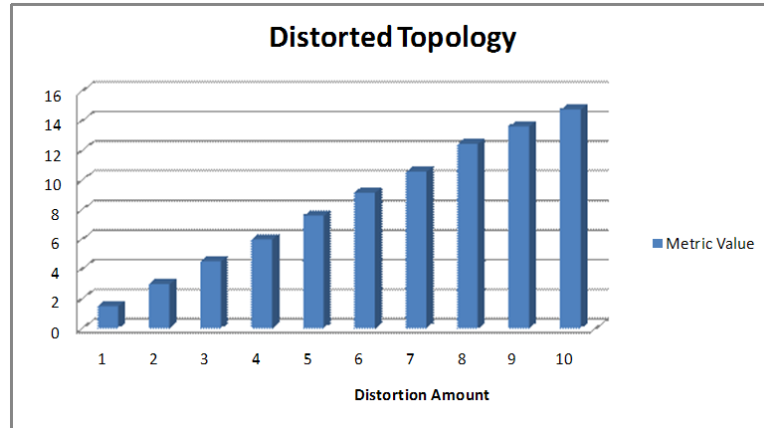


(b)

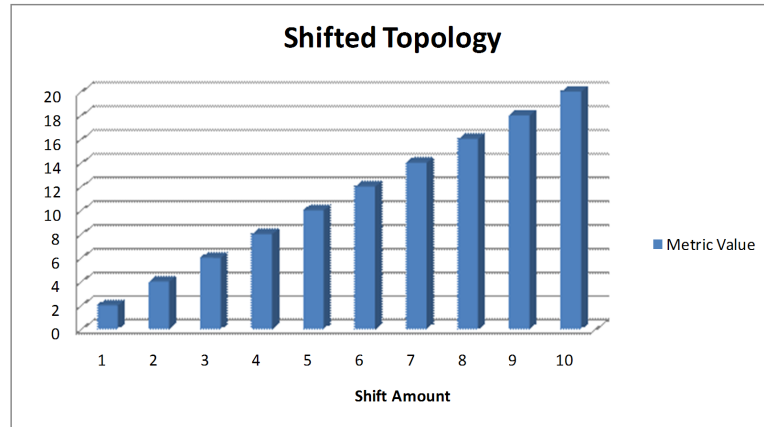


(c)

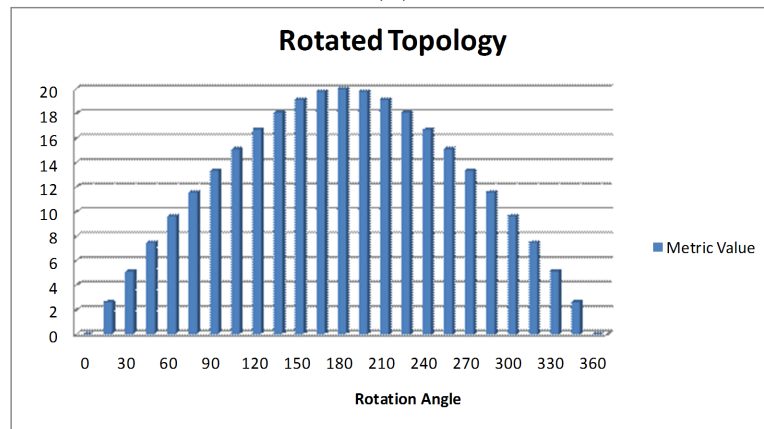
Figure 4.1: Euclidean distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle.



(a)



(b)



(c)

Figure 4.2: Hamming distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle.

4.4.3 Tanimoto Distance Behavior

Figures 4.3(a) and (c) show us that the Tanimoto distance is distortion and rotation sensitive, while Figure 4.3(b) indicates that it is shift insensitive. The metric shows totally different behavior against distorted and shifted topologies. Simulation results indicate that Tanimoto distance metric tolerates topology preserving shift class localization errors. On the other hand, it behaves similar to Euclidean and Hamming distance metrics in case of rotated topologies. The behavior of Tanimoto distance metric can be reported as:

- Shift Insensitive,
- Rotation Sensitive.

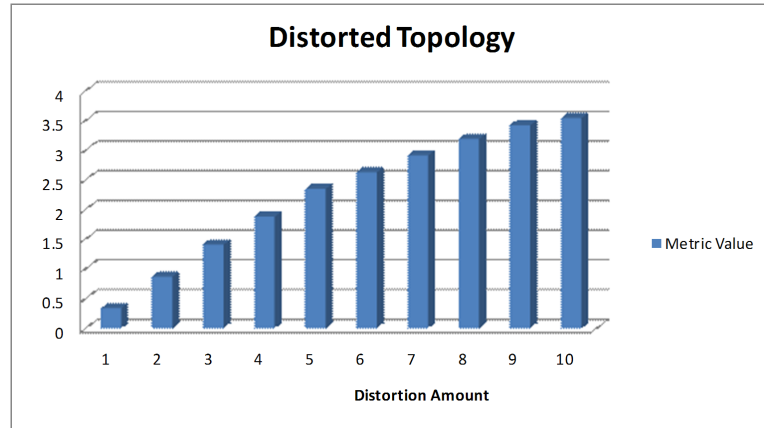
4.4.4 Cosine Distance Behavior

Figures 4.4(a) and (c) show us that the Cosine distance is distortion and rotation sensitive, however Figure 4.6(b) shows that it is shift insensitive. The metric shows completely different behavior against distorted and shifted topologies. Simulation results point out that Cosine distance metric tolerate topology preserving shift class localization errors. In case of rotated topologies, it behaves similar to Euclidean, Hamming and Tanimoto distance metrics. The behavior of Cosine distance metric is:

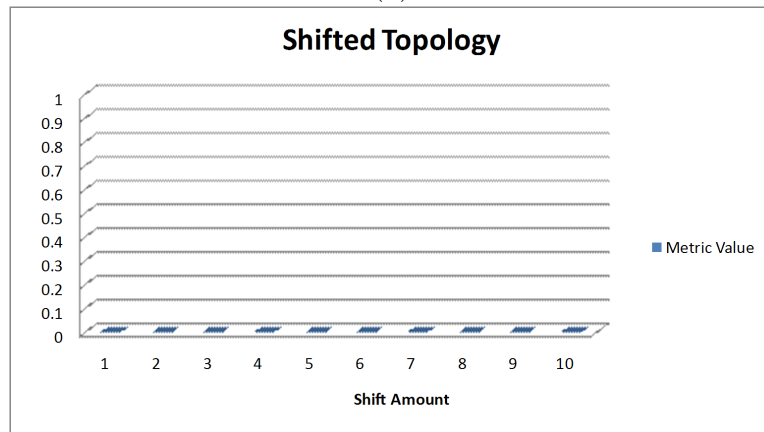
- Shift Insensitive,
- Rotation Sensitive.

4.4.5 CVD Behavior

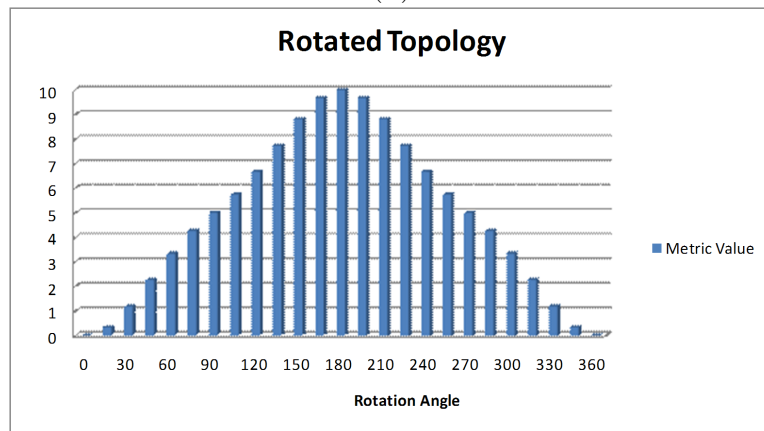
Figures 4.5(a), (b) and (c) show us that the CVD is distortion, shift and rotation sensitive. Comparing Figure 4.5(a) and (b), we find out that the metric behavior



(a)

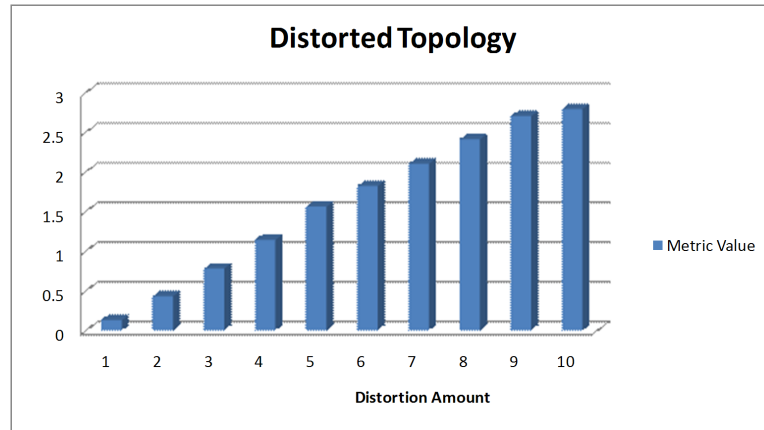


(b)

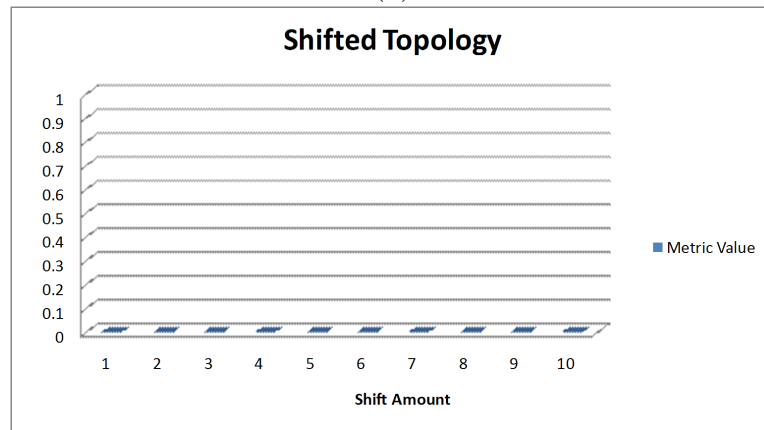


(c)

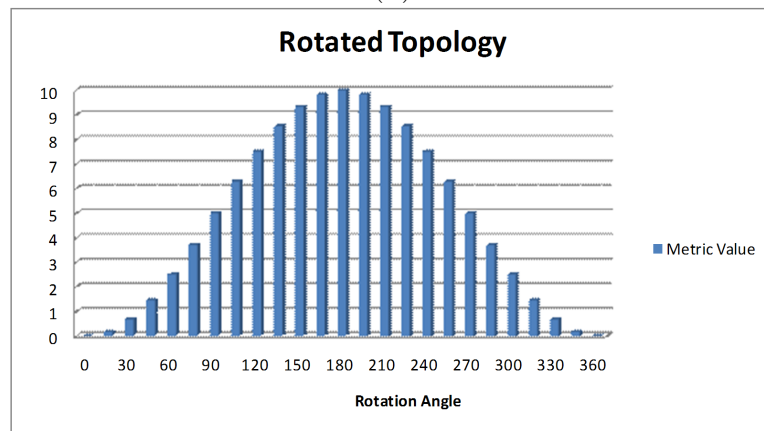
Figure 4.3: Tanimoto distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle.



(a)



(b)



(c)

Figure 4.4: Cosine distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle.

is similar for distorted and shifted topologies. Simulation results indicate that CVD metric does not tolerate topology preserving localization errors.

The behavior of CVD metric can be reported as:

- Shift Sensitive,
- Rotation Sensitive.

4.4.6 RED Behavior

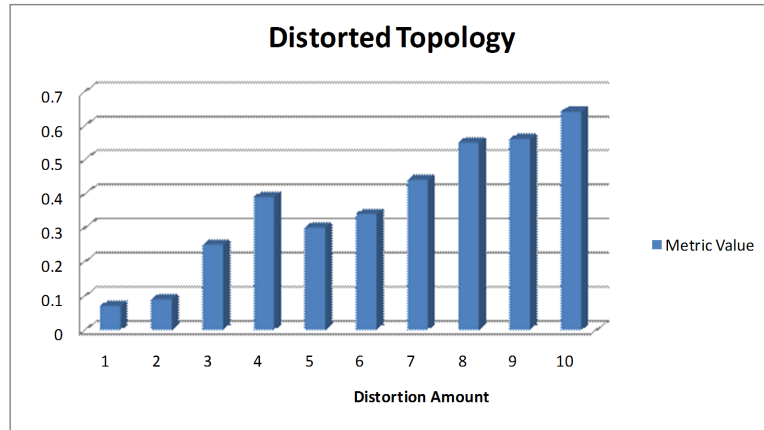
Figures 4.6(a) and (c) show us that the RED is distortion and rotation sensitive, however Figure 4.6(b) implies that it is shift insensitive. The metric shows completely different behavior against distorted and shifted topologies. Simulation results point out that RED metric tolerate topology preserving shift class localization errors. In case of rotated topologies, it behaves similar to Euclidean, Hamming, Tanimoto and CVD metrics. The behavior of RED metric is:

- Shift Insensitive,
- Rotation Sensitive.

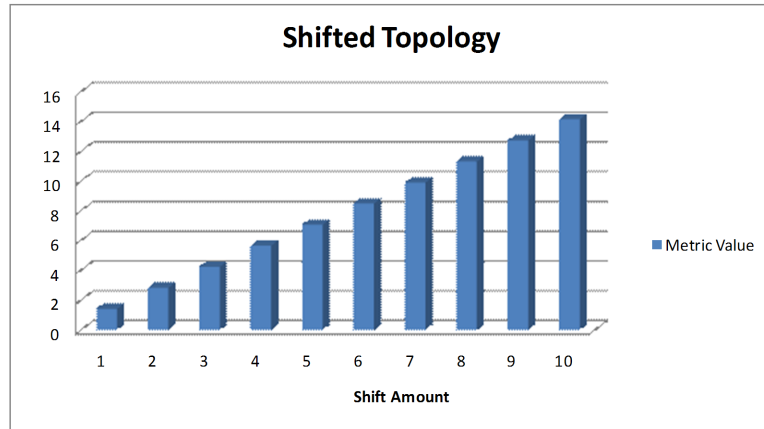
4.4.7 NRED Behavior

Figure 4.7(a) and (c) show us that the NRED is distortion and rotation sensitive. On the other hand Figure 4.7(b) implies that it is shift insensitive. The metric's behaviors are similar to RED. The behavior of NRED metric can be summarized as:

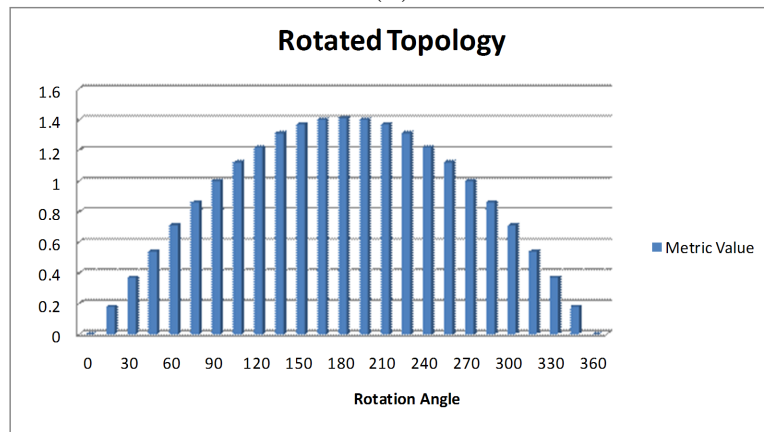
- Shift Insensitive,
- Rotation Sensitive.



(a)

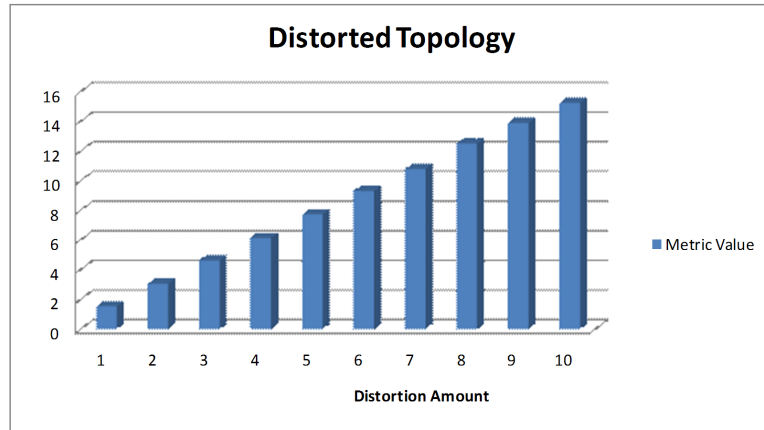


(b)

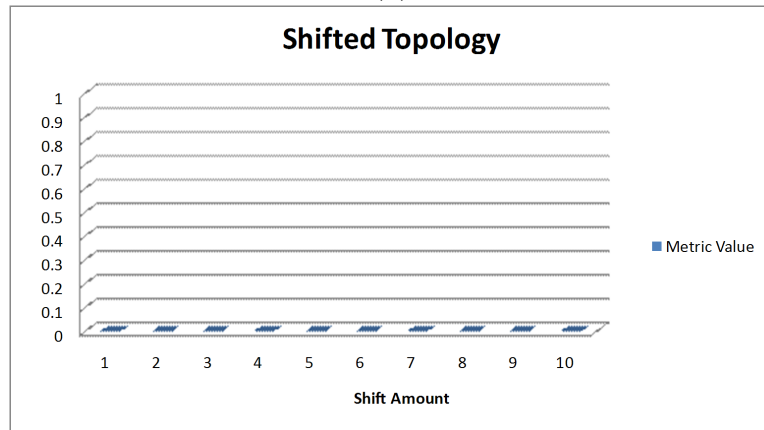


(c)

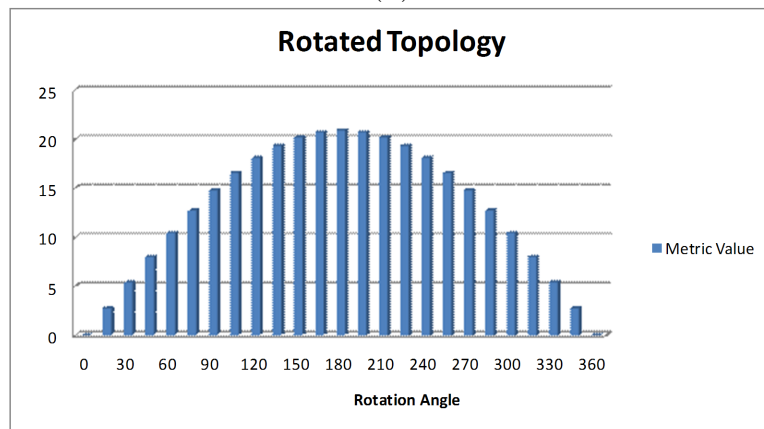
Figure 4.5: Cumulative Vectorial distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle.



(a)

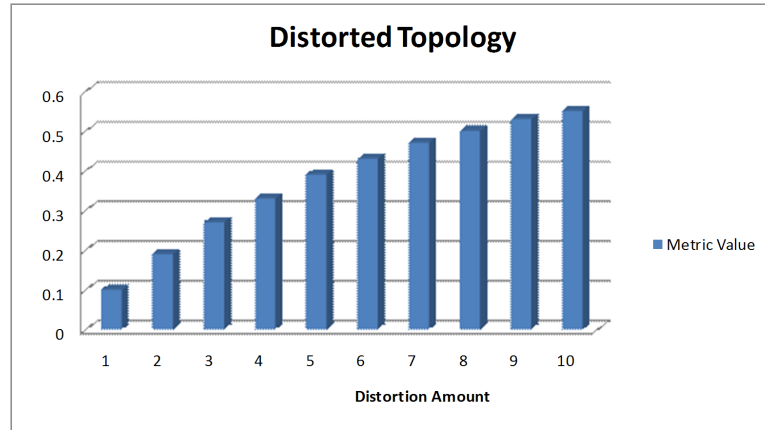


(b)

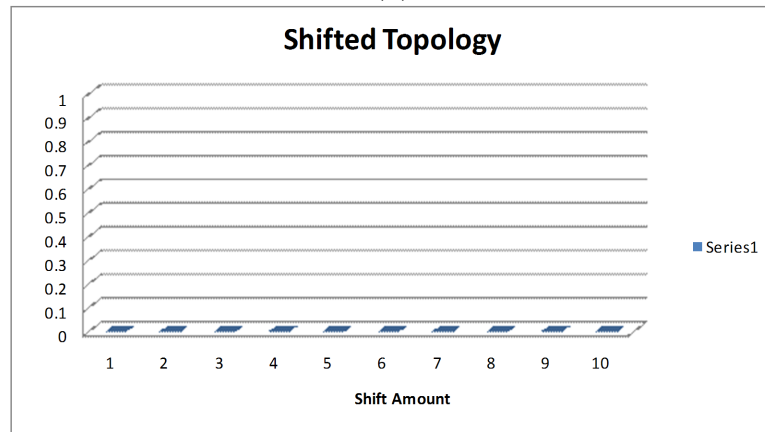


(c)

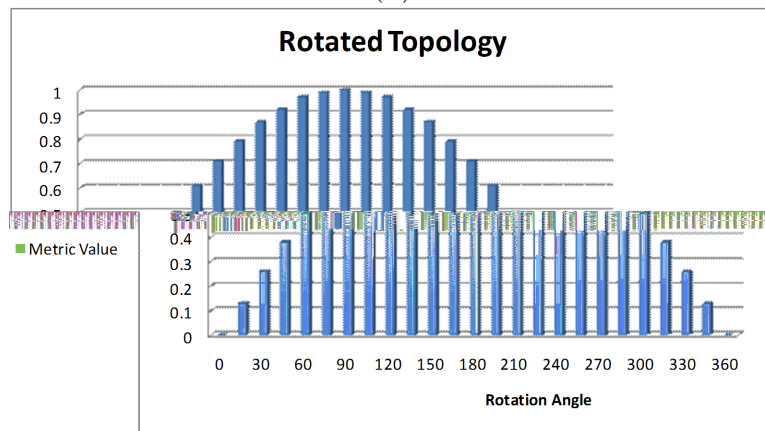
Figure 4.6: RED metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle.



(a)



(b)



(c)

Figure 4.7: NRED metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle.

4.4.8 Spring Distance Behavior

4.4.8.1 Spring A Distance Behavior

Figures 4.8(a), (b) and (c) show us that the Spring A distance is distortion, shift and rotation sensitive. Comparing Figure 4.8(a) and (b), we realize that the metric behavior is similar for distorted and shifted topologies. Thus Spring A metric does not tolerate topology preserving localization errors. Specifically, its behavior against distorted and shifted topologies demonstrate how it is unaware of topological changes similar to some other distance metrics including Euclidean and Hamming distance metrics. The behavior of Spring A distance metric can be declared as:

- Shift Sensitive,
- Rotation Sensitive.

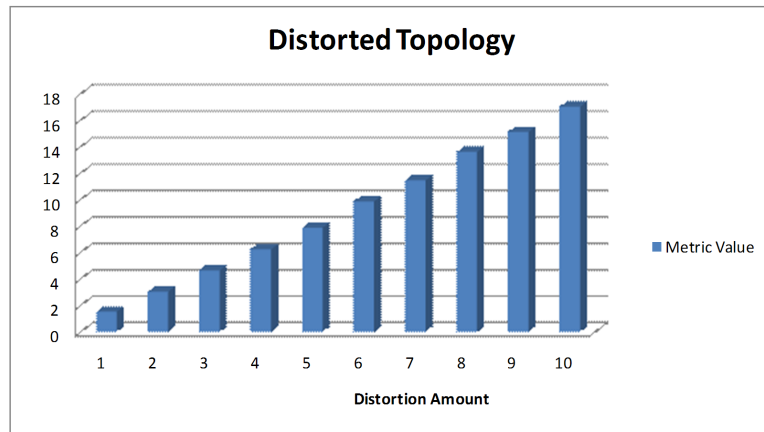
4.4.8.2 Spring B Distance Behavior

Figures 4.9(a), (b) and (c) show us that the Spring B distance is distortion, shift and rotation sensitive. Comparing Figure 4.8 and Figure 4.9, we realize that Spring A and Spring B show similar behavior. The behavior of Spring B distance metric can be listed as:

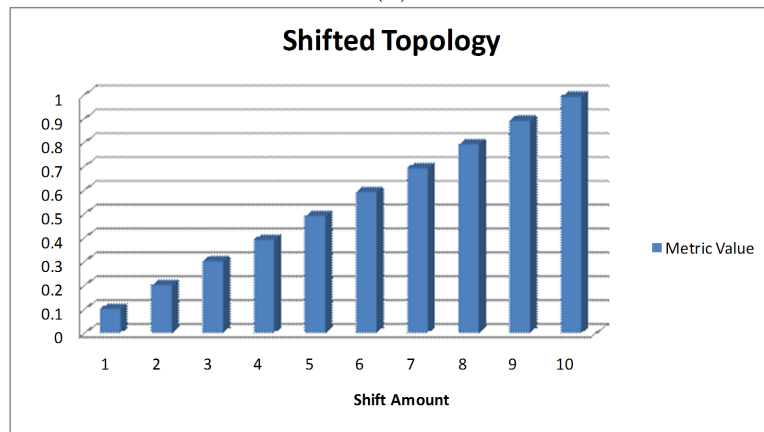
- Shift Sensitive,
- Rotation Sensitive.

4.4.8.3 Spring C Distance Behavior

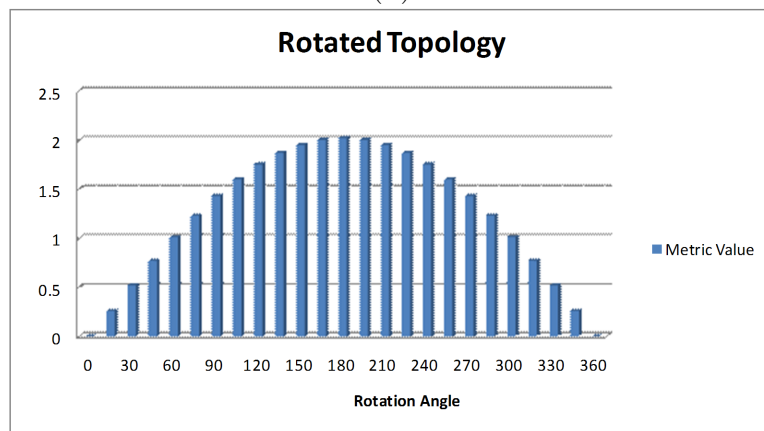
Figures 4.10(a) and (c) show us that the Spring C distance is distortion and rotation sensitive. However, Figure 4.10(b) indicates that it is shift insensitive.



(a)

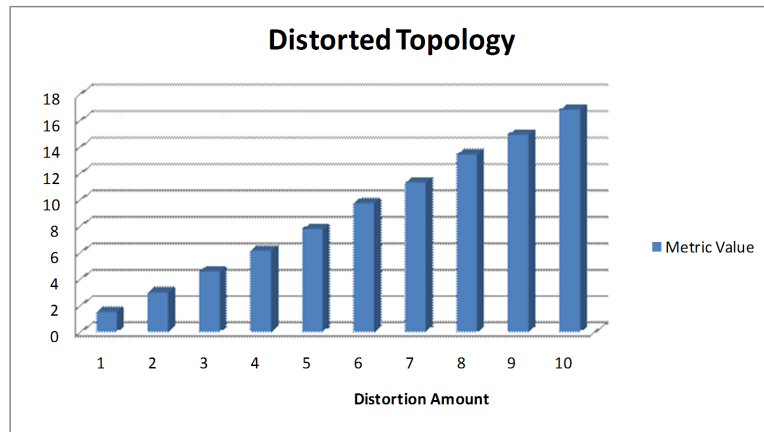


(b)

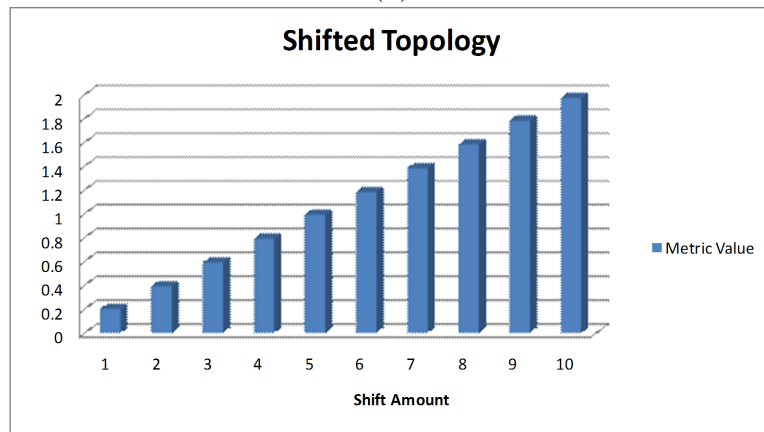


(c)

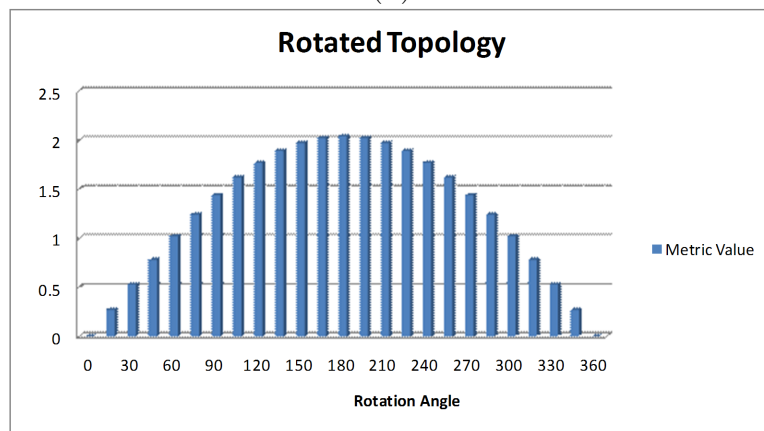
Figure 4.8: Spring A distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle.



(a)



(b)



(c)

Figure 4.9: Spring B distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle.

The metric behavior for distorted and shifted topologies is different. As simulation results indicate, Spring C distance metric tolerate topology preserving shift class localization errors, however, it shows similar behavior to other distance metrics in case of rotated topologies. The behavior of Spring C distance metric can be listed as:

- Shift Insensitive,
- Rotation Sensitive.

4.4.8.4 Spring D Distance Behavior

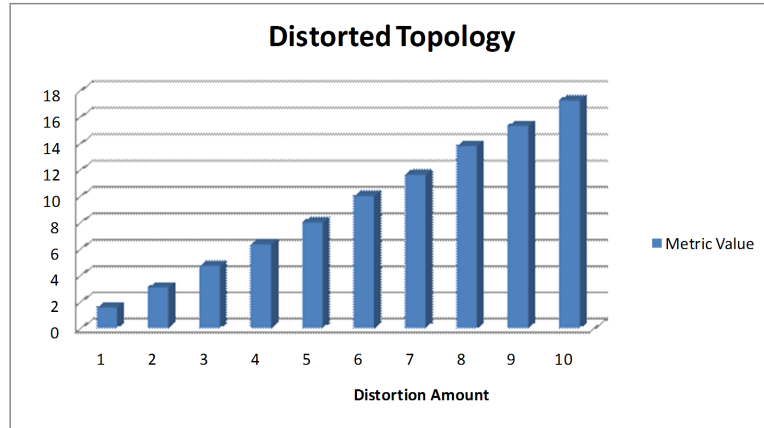
For our last metric, in Figure 4.11(a) we find out that the Spring D distance is only distortion sensitive. Moreover, Figure 4.10(b) indicates that it is shift insensitive and Figure 4.10(c) reveals that the metric is rotation insensitive. Spring D distance metric tolerates both shift class and rotation class localization errors. This behavior is unique to this metric. The behavior of Spring D distance metric can be listed as:

- Shift Insensitive,
- Rotation Insensitive.

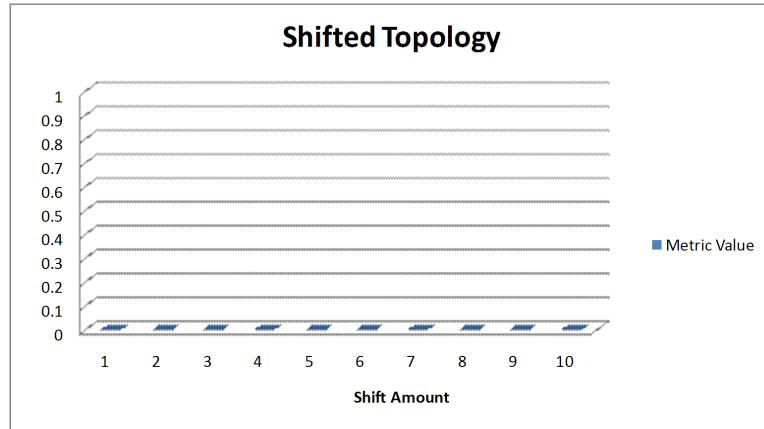
4.5 Evaluation Based on the Topologies

4.5.1 Rotated Topologies

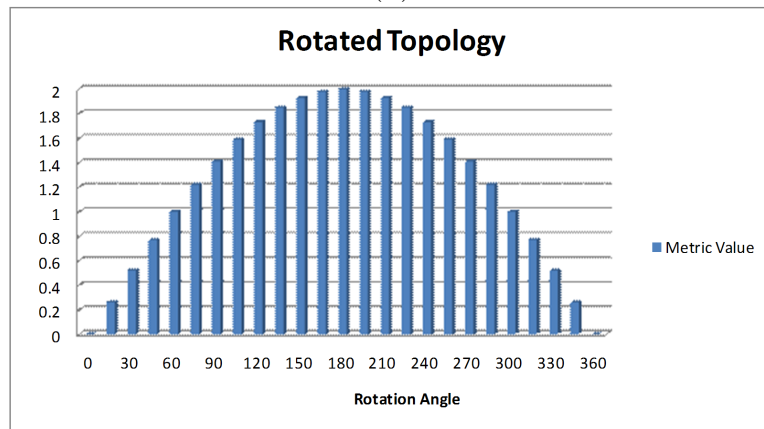
In this section we study topologies that are rotated with respect to the coordinate system. For instance, we place all nodes on a moving plane, and then rotate the plane such that the distance between nodes stay exactly the same while the



(a)

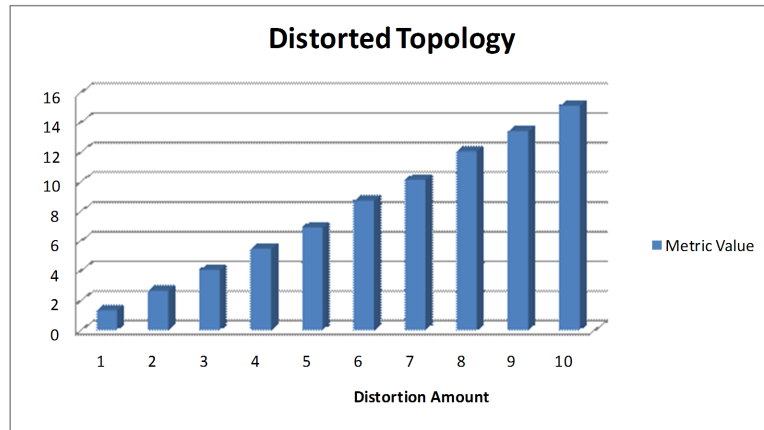


(b)

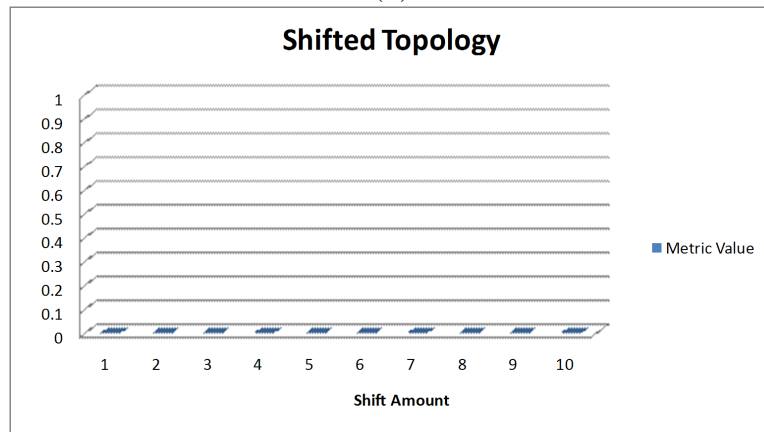


(c)

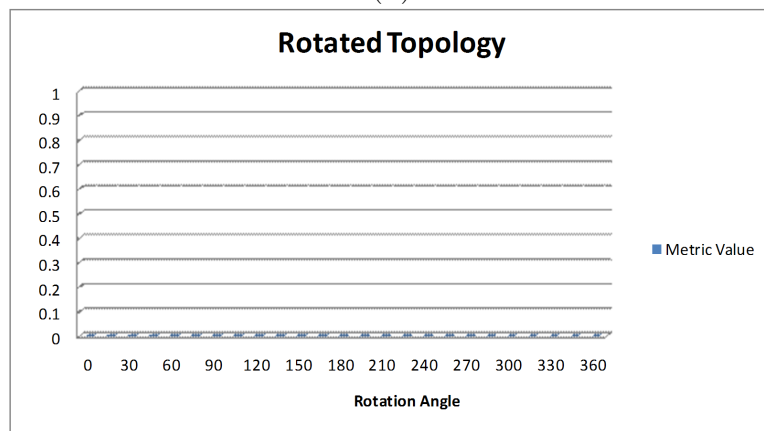
Figure 4.10: Spring C distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle.



(a)



(b)



(c)

Figure 4.11: Spring D distance metric behavior against certain changes in topology. In (a) the metric value is drawn against the distortion amount, in (b) the metric value is drawn against the shift amount, and in (c) the metric value is drawn against the rotation angle.

overall alignment differs. This simple scenario is presented to provide a better understanding of the metrics being studied.

In this scenario S' is generated by using rotate localization functions. i.e. $S' \leftarrow \delta_{rotate(\alpha)}(S)$ for $\alpha = 0 \dots 360$ degrees

For this purpose we run simulations using this scenario, and Figure 4.12 demonstrates the results as the plane is rotated at an increasing angle. In the figure, the distance between the original and estimated topology is plotted for various metrics and for increasing angles. As shown in the figure, all metrics, except Spring D, report an increasing diversion from the original topology as the angle is increased up to 180. The behavior is fully symmetric for all these metrics peaking at 180 and returning back to normal as further increased to 360, which is the original topology. However, Spring D reports the distance zero independent of rotation angle.

In traditional pattern matching problems we would expect similarity degrees to be high 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. In this regard, among all metrics RED 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.

4.5.2 Shifted Topologies

The second scenario we will study is a topology with a perfect shift. That is all nodes in the network are subject to an absolute distortion in a particular direction. For instance, all nodes deployed on a lake surface are moved 2 meters northeast by wind effects. In some cases, tolerable 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.

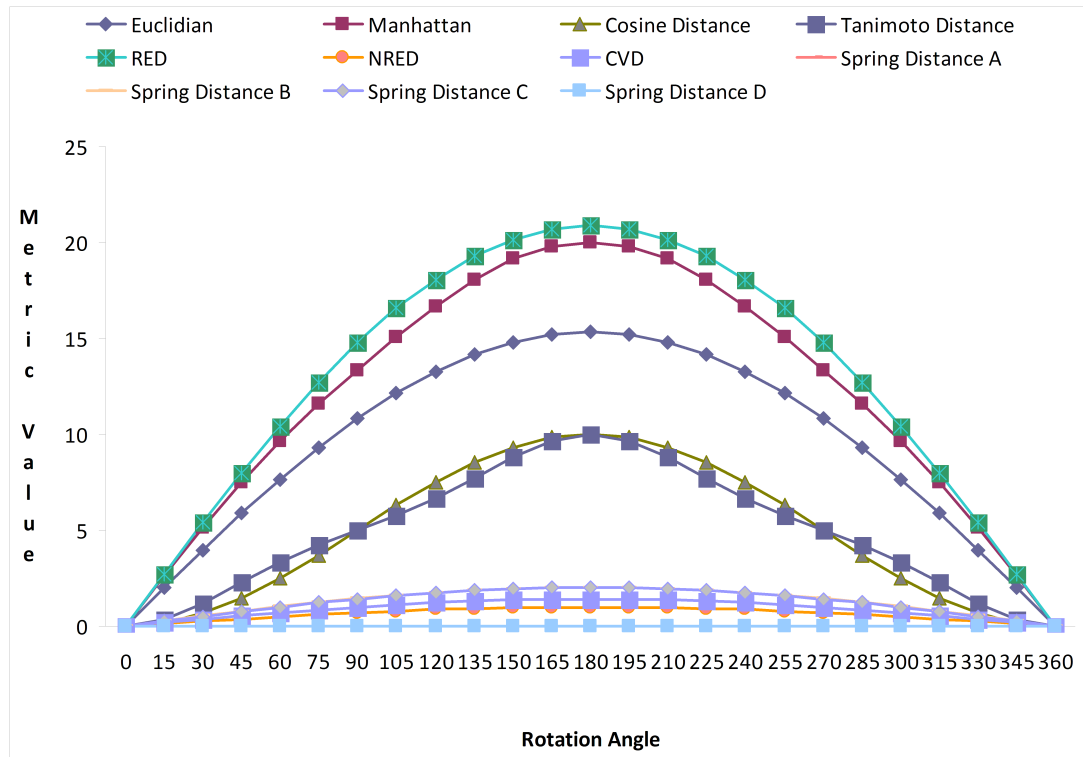


Figure 4.12: Estimation errors in case of estimates that are simply rotated replicas of the original topology: On the x -axis the rotation angle is increased from 0 to 360 and the metric value comparing the original and the rotated topology is reported on the y -axis.

To study this type of topology changes, we imposed various shifts and evaluated the resulting topologies using each metric. Here we will present results from a case where we applied a position shift such that each node will have the coordinate $(x + n, y + n)$, where (x, y) is the original coordinate of the node and n is a positive number representing the shift amount, between 1-10. Even though this is a rather simplified version, i.e., in practice some node can move more than others, the scenario will help us observe the behavior of metrics for the general case of shifted topologies.

In Figure 4.13, we demonstrate the metric values to reflect the distance of the original and the shifted topology, as the magnitude of the shift (n) is increased on the x -axis. As demonstrated, even though the complete topology (graph representation of the network) is preserved perfectly, Euclidean distance, Manhattan

distance, CVD, Spring A distance and Spring B distance are not able to capture this fact. All these metrics report an error that is directly proportional to the shift amount.

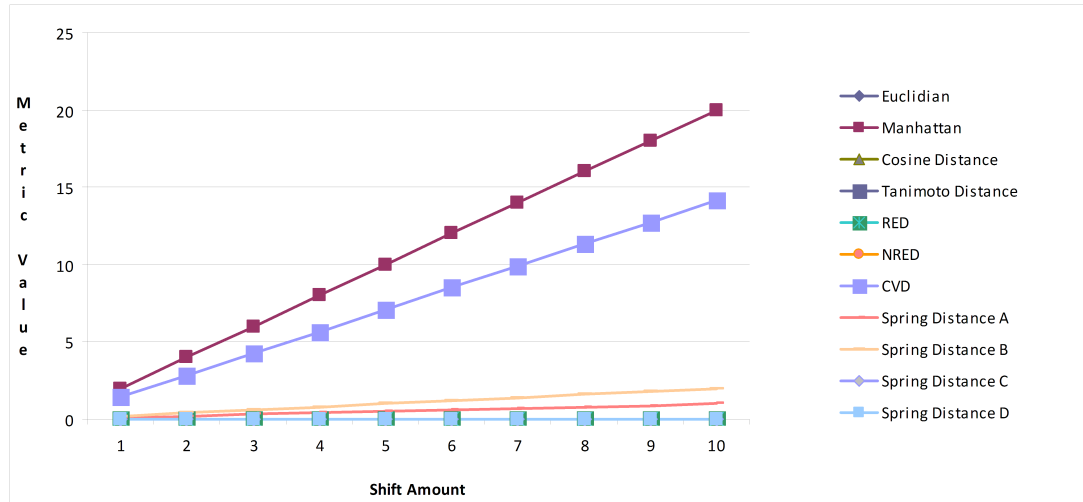


Figure 4.13: Estimation errors in case of topology shifts. On the x -axis the shift amount is increased from 1 to 10.

4.5.3 Distorted Topologies

Distorted topologies are the category that represents arbitrary errors made in position estimates. In this category we study a scenario where node positions are shifted along arbitrary directions within a distance of n from the node. In this scenario we apply an independent shift to each node such that the resulting topology will have some relative accuracy errors.

Since all localization algorithms are de facto distortion sensitive, it is not defined in Section 2.2 as a separate class. However, it is significant to know the magnitude of sensitivity

In Figure 4.14 we plot the metric values as the maximum distance from the original position is increased on the x -axis. Note that Manhattan distance and Euclidean distance report relatively smaller errors in this arbitrary topology case since they do not consider the relative positioning of nodes. Spring distances

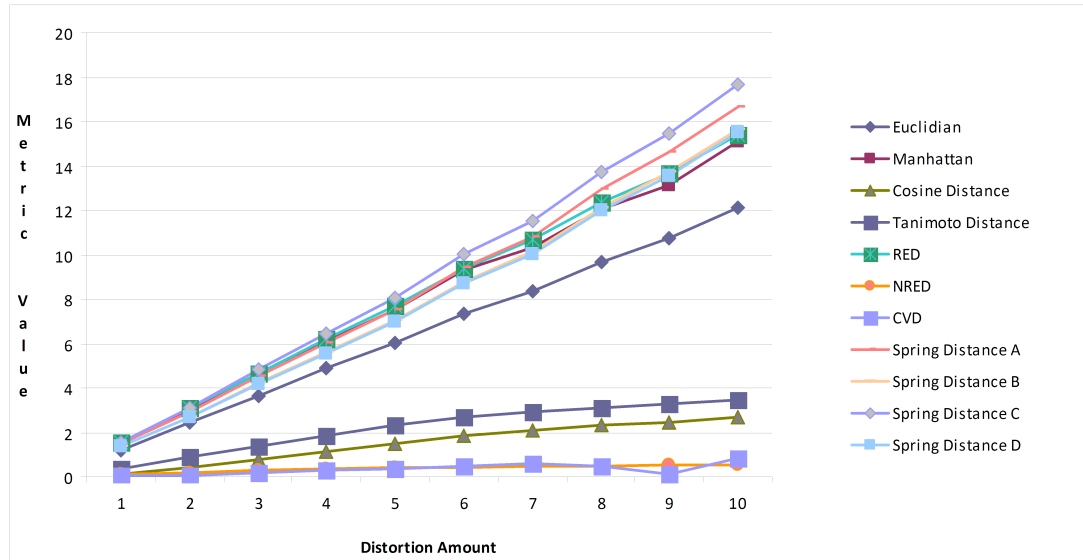


Figure 4.14: Estimation errors in case of topology distortions. On the x -axis the distortion amount is increased from 1 to 10.

and RED, on the other hand, which were reporting the topology to be similar in the previous section, now report significant divergence in this arbitrary topology change scenario.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

There are various applications of wireless sensor networks some of which require precise location information, e.g. sniper localization [31], while some other applications may only need relatively accurate location information, e.g. ZebraNet [13] where behaviors of animals were observed. The impact of errors in location information to the applications mostly depends on 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.

As shown in Figure 5.1, for a planned wireless sensor network application, we suggest identifying and listing the characteristics of the required location data based on its sensitivity to shift and rotation errors. Then, appropriate metric can be chosen by looking up the Table 5.1, in which we suggest metrics according to application requirements on location data. 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.

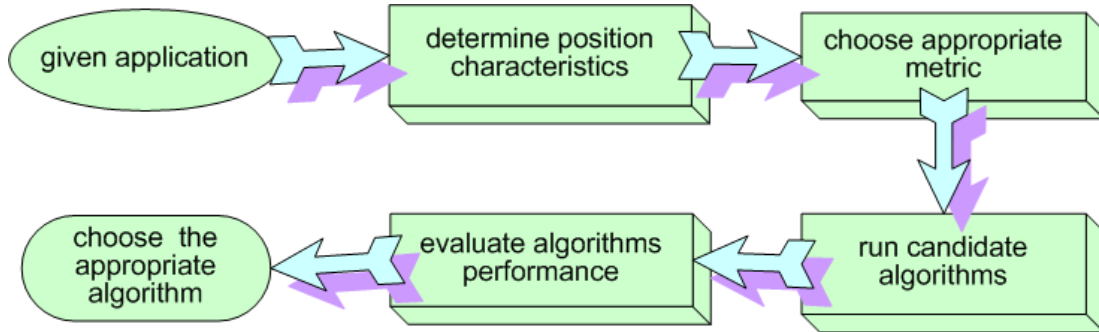


Figure 5.1: Suggested steps to choose appropriate algorithm for a planned wireless sensor network application.

SHIFT	ROTATE	SUGGESTED METRICS
sensitive	sensitive	Spring A, Euclidean
sensitive	insensitive	CVD, Spring B
insensitive	sensitive	Spring C, RED, Cosine
insensitive	insensitive	Spring D

Table 5.1: Metric suggestions based on location characteristics.

Localization (i.e. estimating location of sensor nodes), is a hot research area. A number of algorithms have been proposed in this regard. Yet, the evaluation of these algorithms traditionally depends on fairly simplistic metrics based on the original and the estimated position of each node in isolation to the rest of the network. In this work we discussed implications of errors considering the expectations of the end user. We discuss that there is a need for a new metric that will consider the relative positioning of each node with respect to the original topology for accurate data analysis. With this motivation we studied and also proposed alternative metrics to evaluate the localization errors of algorithms. We studied these metrics for various simplistic scenarios to provide a better understanding of the issues. We discussed the advantages of one metric to the other for specific applications. We suggest a methodology that is summarized into a table for metric selection based on the localization needs of a wireless sensor network application.

5.2 Future Work

In this work, we point out the need for topology and application aware distance metrics. We, then, try to understand the behavior of existing metrics and the metrics we proposed. Our study is based on some certain topological changes: shifted topologies and rotated topologies. Here, the topology concept might be studied further. A proposal for a different topology definition in context of wireless sensor network will make it easier to evaluate and understand the metrics.

We propose some new metrics to be used in evaluation of localization algorithms. These metrics are designed to satisfy some certain needs of specific wireless sensor network applications, e.g. rotation insensitivity for ZebraNet which is discussed in Section 2.2.2. As a result of these application dependent metric proposals, for each wireless sensor network application it is necessary to determine the application objectives for location sensitivity, and then choose the most appropriate metric to be used in selection of localization algorithm. A topological distance metric, which can be used for general sensor network applications, will be a good contribution to the localization studies in wireless sensor networks.

The performance of localization algorithms is determined by use of different metrics one of which is distance metric. Having designed a novel topological distance metric, a new localization algorithm that tries to minimize distance error reported by this metric would be a proper use of it. To sum up, the future work is composed of:

- a proposal for a different topology definition in context of wireless sensor networks,
- a topological distance metric that can be used for general sensor network applications,
- a new localization algorithm that tries to minimize distance error reported by the topological distance metric.

Bibliography

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102 – 114, August 2002.
- [2] A. Bharathidasan and V. A. S. Ponduru. Sensor networks: An overview. IEEE Infocom, 2004.
- [3] M. Boushaba, A. Hafid, and A. Benslimane. HA-A2L: angle to landmark-based high accuracy localization method in sensor networks. In *IWCMC '07: Proceedings of the 2007 international conference on Wireless communications and mobile computing*, pages 475–480, New York, NY, USA, 2007. ACM.
- [4] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *Technical report 00-729, Computer science department, University of Southern California*, April 2000.
- [5] R. Crepaldi, P. Casari, A. Zanella, and M. Zorzi. Testbed implementation and refinement of a range-based localization algorithm for wireless sensor networks. In *Mobility '06: Proceedings of the 3rd international conference on Mobile technology, applications & systems*, page 61, New York, NY, USA, 2006. ACM.
- [6] M. Ding, F. Liu, A. Thaeler, D. Chen, and X. Cheng. Fault-tolerant target localization in sensor networks. In *EURASIP Journal on Wireless Communications and Networking*, 2007.

- [7] S. O. Dulman. *Data-centric architecture for wireless sensor networks*. PhD thesis, University of Twente, 2005.
- [8] M. Erol, L. F. M. Vieira, and M. Gerla. Localization with dive'n'rise (dnr) beacons for underwater acoustic sensor networks. In *WuWNet '07: Proceedings of the second workshop on Underwater networks*, pages 97–100, New York, NY, USA, 2007. ACM.
- [9] K. Fall and K. Varadhan. The ns manual (formerly ns notes and documentation). In *The VINT Project, A Collaboration between researchers at UC Berkeley, LBL, USC/ISI and Xerox PARC*, November 2005.
- [10] G. Giorgetti, S. K. S. Gupta, and G. Manes. Wireless localization using self-organizing maps. In *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 293–302, New York, NY, USA, 2007. ACM.
- [11] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. Range-free localization schemes in large scale sensor networks. *MobiCom*, 2003.
- [12] R. Iyengar and B. Sikdar. Scalable and distributed GPS free positioning for sensor networks. In *Proc. of IEEE ICC 2003*. Anchorage, Alaska, May 2003.
- [13] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. In *ASPLOS-X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, pages 96–107, New York, NY, USA, 2002. ACM.
- [14] N. D. Lane, H. Lu, and A. T. Campbell. Ambient beacon localization: using sensed characteristics of the physical world to localize mobile sensors. In *EmNets '07: Proceedings of the 4th workshop on Embedded networked sensors*, pages 38–42, New York, NY, USA, 2007. ACM.
- [15] L. Lazos and R. Poovendran. Serloc: secure range-independent localization for wireless sensor networks. In *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*, pages 21–30, New York, NY, USA, 2004. ACM.

- [16] M. Li and Y. Liu. Rendered path: range-free localization in anisotropic sensor networks with holes. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 51–62, New York, NY, USA, 2007. ACM.
- [17] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, New York, NY, USA, 2002. ACM.
- [18] Merrill, W. Newberg, F. Sohrabi, K. Kaiser, and W. Pottie. Collaborative networking requirements for unattended ground sensor systems. In *Aerospace Conference*, pages 5_2153– 5_2165. IEEE, 2003.
- [19] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proc. 2nd ACM SenSys*, pages 50–61, Baltimore, MD, November 2004.
- [20] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proc. 2nd ACM SenSys*, pages 50–61, Baltimore, MD, November 2004.
- [21] R. Moses, D. Krishnamurthy, and R. Patterson. A self-localization method for wireless sensor networks. *EURASIP Journal on Applied Signal Processing*, 2002.
- [22] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In F. Zhao and L. Guibas, editors, *Information Processing in Sensor Networks*, Lecture Notes in Computer Science 2634, pages 333–348, 2003.
- [23] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery. Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Comput. Surv.*, 39(3), 2007.
- [24] D. Niculescu and B. Nath. Error characteristics of ad hoc positioning systems (aps). In *MobiHoc '04: Proceedings of the 5th ACM international symposium*

- on Mobile ad hoc networking and computing*, pages 20–30, New York, NY, USA, 2004. ACM.
- [25] R. Peng and M. L. Sichitiu. Probabilistic localization for outdoor wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(1):53–64, 2007.
- [26] M. Perkins, N. Correal, and R. O’Dea. Emergent wireless sensor network limitations: A plea for advancement in core technologies. *IEEE Sensors*, 2:1505–1509, June 2002.
- [27] M. Rudafshani and S. Datta. Localization in wireless sensor networks. In *IPSN ’07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 51–60, New York, NY, USA, 2007. ACM.
- [28] C. Savarese, J. Rabaey, and J. Beutel. Locationing in distributed ad-hoc wireless sensor networks. *Proceedings of the ICASSP*, May 2001.
- [29] C. Savarese, J. Rabaey, and J. Beutel. Locationing in distributed ad hoc wireless sensor networks. In *Proc. 2001 Int’l Conf. Acoustics, Speech, and Signal Processing (ICASSP 2001)*, volume 4, pages 2037–2040. IEEE, Piscataway, NJ, May 2001.
- [30] Y. Shang and W. Ruml. Improved mds-based localization. *Infocom 2004*, 2004.
- [31] G. Simon, M. Maróti, Ákos Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In *SenSys ’04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 1–12, New York, NY, USA, 2004. ACM.
- [32] N. R. C. Staff. *Embedded Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. National Academy Press, Washington, DC, USA, 2001.
- [33] R. Stoleru, T. He, J. A. Stankovic, and D. Luebke. A high-accuracy, low-cost localization system for wireless sensor networks. In *SenSys ’05: Proceedings*

- of the 3rd international conference on Embedded networked sensor systems*, pages 13–26, New York, NY, USA, 2005. ACM.
- [34] R. Stoleru, J. A. Stankovic, and S. H. Son. Robust node localization for wireless sensor networks. In *EmNets '07: Proceedings of the 4th workshop on Embedded networked sensors*, pages 48–52, New York, NY, USA, 2007. ACM.
- [35] R. Stoleru, P. Vicaire, T. He, and J. A. Stankovic. Stardust: a flexible architecture for passive localization in wireless sensor networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 57–70, New York, NY, USA, 2006. ACM.
- [36] S. T. V, A. K. Katti, and A. V. S. Analysis of mobile beacon aided in-range localization scheme in ad hoc wireless sensor networks. In *IWCMC '06: Proceedings of the 2006 international conference on Wireless communications and mobile computing*, pages 1159–1164, New York, NY, USA, 2006. ACM.
- [37] A. Youssef, M. Younis, M. Youssef, and A. Agrawala. On the accuracy of multi-hop relative location estimation in wireless sensor networks. In *IWCMC '07: Proceedings of the 2007 international conference on Wireless communications and mobile computing*, pages 481–486, New York, NY, USA, 2007. ACM.
- [38] J. Zhang, T. Yan, J. A. Stankovi, and S. H. Son. Thunder: towards practical, zero cost acoustic localization for outdoor wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(1):15–28, 2007.