

DISTRIBUTED CALIBRATION OF WIRELESS
MULTIMEDIA SENSOR NETWORKS

By

ROHIT SHRIKANT KADAM

Bachelor of Engineering in Instrumentation and Control

University of Pune

Pune, Maharashtra, India

2007

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December 2010

DISTRIBUTED CALIBRATION OF WIRELESS
MULTIMEDIA SENSOR NETWORKS

Thesis Approved:

Dr. Weihua Sheng

Thesis Adviser

Dr. Sohum Sohoni

Dr. Nazanin Rahnavard

Dr. Mark E. Payton

Dean of the Graduate College

ACKNOWLEDGMENT

I thank Dr Weihua Sheng (Committee Chair), for his invaluable guidance and constant motivation.

I would also like to thank Dr Sohum Sohoni and Dr Nazanin Rahnavard (committee members) for their invaluable inputs..

I would like to thank Department of Electrical and Computer Engineering.

I also thank Sijian Zhang, PhD student at Zhejiang University, China on sharing his insights on the topic.

I thank the lab group of ASCC laboratory

I thank my friends for standing by my side without waiver and giving endless encouragement through the many hours of work and sleepless nights.

Finally I thank my parents, Shrikant and Sujata Kadam for raising me to be the man that I am today. I dedicate this thesis to them

This project is supported by the DoD ARO DURIP grant 55628-CS-RIP.

Table of Contents

INTRODUCTION	4
1.1 MOTIVATION	4
1.2 RELATED WORK	6
OVERVIEW	13
2.1 PROBLEM FORMULATION	13
2.2 OVERALL APPROACH FOR DISTRIBUTED CAMERA CALIBRATION	15
2.3 MULTIDIMENSIONAL SCALING	18
2.3.1 <i>Introduction to Multidimensional Scaling</i>	18
2.3.2 <i>Types of Multidimensional Scaling</i>	21
2.3.3 <i>Algorithm for MDS based localization</i>	22
2.3.4 <i>Performance of MDS-Based Localization</i>	25
DETAILED ALGORITHMS FOR DISTRIBUTED CALIBRATION OF WMSN	27
3.1 LOCAL MAP GENERATION	27
3.1.1 <i>Vision map generation</i>	27
3.1.2 <i>ID map generation</i>	30
3.1.3 <i>ID Association Algorithm</i>	31
3.2 LOCAL MAP VALIDATION	33
3.3 MAP MERGING	34
SIMULATION RESULTS	36
4.1 SIMULATION SETUP	36
4.2 RESULTS	36
4.2.1 <i>Local Map Generation</i>	38
4.2.2 <i>Symmetry Correction Algorithm</i>	43
4.2.3 <i>Map Merging</i>	46
CONCLUSION AND FUTURE WORK	54
REFERENCES	56

LIST OF TABLES

Table	PAGE
Table 1: Inter-City Distance Matrix.....	19
Table 2 : True Mean Square Error Of The Locations In The Vision Map And The Id Map R_V=3.1	42
Table 3 : True Mean Square Error Of The Locations In The Vision Map And The Id Map R_V=4.1	43
Table 4 : Correction In Symmetry Error	46
Table 5 : % Of Good Nodes In The Local Map Of The Base Node 55.....	51
Table 6: % Of Good Nodes For Base Node 66.....	52
Table 7: Mean Square Error For Various Placement Errors (Noise) For A Selected Vision Range Of 3.1	53

LIST OF FIGURES

Figure	Page
Figure 1: The design of the multimedia sensor node.....	14
Figure 2: Omnidirectional Camera	14
Figure 3 : The self-calibration of sensor node with respect to node 'i'	15
Figure 4: Overall Approach	16
Figure 5 : Relative and absolute maps of cities of Europe	21
Figure 6 : A typical vision map with range and angle errors.....	29
Figure 7: A typical ID map	31
Figure 8: The ground truth of the sensor deployment (100 nodes, $R=1.99r$, placement error=10%).....	37
Figure 9: The ground truth for ID map (green stars) of the sensor nodes and the vision map obtained (black triangles).....	38
Figure 10: ID Association using pre-registration. The nodes in the vision map are brought closer to the nodes in the ID map for node ID 55	39
Figure 11: The map shows the correct ID association between the ID map and the vision map after performing the ICP refinement algorithm for node 55. The IDs shown in the figure indicate that they have the perfect match	39
Figure 12: Pre-registration example. Before the implementation of iterative closest point algorithm for node 66.....	40
Figure 13: The map shows the correct ID association between the ID map and the vision map after performing the ICP refinement algorithm for node 66.....	41
Figure 14: Pre-registration - Nodes in Vision map brought close to nodes in ID map for base node 33.....	41
Figure 15: Local Map generated for base node ID 33.....	41
Figure 16: Incorrect ID association. The MSE value is 3.83.....	44
Figure 17: Incorrect ID association. The MSE value is 2.79.....	44
Figure 18: Correct ID association after applying reflection and comparing the MSE values. The MSE value obtained is 0.07	45
Figure 19: Merging of local maps of node 55 with local map of node 37.....	47
Figure 20: Intermediate global maps for nodes 37, 33, 77 merged with local map of the base node 55.....	47
Figure 21: The global map for base node 55	48
Figure 22: Node ID 66 merged with Node ID 47	49
Figure 23: Intermediate global maps for nodes 47,25,63 merged with local map of the base node 66.....	49
Figure 24: Global map for base node 66.....	50
Figure 25: Global Map for base node 33	50

CHAPTER 1

INTRODUCTION

Wireless Multimedia Sensor Networks (WMSNs) are gaining popularity among researchers over the past few years. Knowledge of the geographic locations of the sensor nodes is very important in such sensor networks. Location calibration is a method that uses the connectivity information, the estimated distance information among the sensor nodes, as well as the vision images to find the location of the sensor nodes in WMSNs.

Chapter 1 introduces the emerging field of Wireless Multimedia Sensor Networks (WMSNs). First the motivation behind pursuing the research in WMSNs is discussed. Second the literature is reviewed on localization in less expensive short range based sensor networks. Then related work on the calibration problem in multimedia sensor nodes in WMSN is discussed which reviews both single camera calibration and network camera calibration. The chapter is summarized by highlighting the uniqueness of our methodology to perform distributed camera calibration.

1.1 Motivation

Wireless Multimedia Sensor Networks [1] are a research topic of growing interest over the years due to its wide applications. WMSNs are a set of embedded devices that have the capability of processing and communicating video and audio streams collected from the environment in a distributed fashion [1]. Wireless Multimedia Sensor Networks find application in surveillance systems against crime and terrorist attacks. They can also be used for traffic monitoring in cities and highways. They are also very useful in military applications to locate the targets of interest (such as enemy soldiers, tanks) in the battlefield and relay video images to the command center.

The emergence of such new sensors has paved the way for the development of a variety of effective, low power and low cost vision based sensing platforms [2, 3, 4, 5, 6, 7].

For most WMSN applications, it is imperative to have the knowledge of the location of the nodes in order to understand the multimedia data received. Therefore, there is a great need to develop a sensor node calibration algorithm which can be implemented in embedded sensor nodes with limited computational resources.

Traditionally calibration is the process of determining the internal parameters like focal length, distortion, skew coefficient and external parameters like position and orientation of a camera in an environment [8]. Calibration is required to correct the errors caused due to device imperfections and aging [9]. With calibration, a camera sensor node can maintain up to date information about its internal and external parameters in an environment. A wireless camera sensor network operates either as a centralized or as a distributed network. In the centralized mode, all the nodes of the network communicate their data to a central node. The central node runs computational algorithms on the received data and sends back the results to the respective nodes. In the distributed mode, each sensor node runs computational algorithms on its own processor and may also exchange its results with its neighboring nodes. In the case of a large scale sensor network operating in a centralized mode, data transmission to a central station by the other nodes increases the transmission overhead and consumes time. It may also increase the power consumption at the central node at the expense of the remaining nodes of the network. Failure of the central node will lead to the failure of the entire network. Distributed camera calibration [10, 11], on the other hand, has the advantage that a node need not depend on centralized nodes for its calibration. A distributed node can calibrate by itself and communicate its position and orientation information to its neighbors. The sensor nodes perform complex

mathematical calculations to carry out the calibration on their own processors. However, the growing need for low power and low cost sensing technology has limited the computational ability of sensor nodes. Hence, there is a need to develop lightweight calibration algorithms for the purpose of configuring and operating these distributed sensor nodes [3] so as to employ them for potential applications. The main aim of this thesis is to develop innovative distributed computational algorithm to calibrate a wireless multimedia sensor network. In this thesis we discuss the theoretical framework for distributed camera calibration based on vision data, local inter-node distances and local topology.

1.2 Related Work

One of the most straightforward localization techniques is Global Positioning System (GPS) based localization that relies on multilateration technique using time of arrival of signals. It has been operative since early 1990's. For localization in an outdoor environment, GPS works well. Unfortunately, the signal from the GPS satellites is too weak to penetrate most buildings, making GPS useless for indoor localization. Likewise it has many other shortcomings. Multipath effects, delayed signals, and complex clock synchronization requirements and others have limited the usage of GPS to limited applications. Adding to above, the GPS units are very expensive and this makes it almost useless in case of commercial applications where the overheads are mainly specified in terms of financial constraints. This shifted the focus towards less expensive, short ranged sensor network. In recent years, researchers have been developing different localization algorithms to localize these sensor networks.

One kind of technique is based on inter-node distance ranging. The range-based techniques rely on a method of finding the physical distance between any two nodes in a network that are within communication range. This process is called ranging. There are two basic techniques used to

perform ranging: received signal strength and signal propagation time. Received signal strength (RSS) is a way to do ranging by measuring the signal strength of a message at the receiver [12, 16]. The receiver then uses knowledge of the sender's signal power (this might be contained within the message) to determine the power loss. Finally the receiver applies its known model for signal propagation behavior to convert the power loss to a distance, thus estimating how far away the sender is. This is an inaccurate technique. Radio signal propagation behavior is highly dependent on the environment (obstacles, signal fading, metals), and hence they are highly variable. Savvides *et al.* [17] describes experiments that tried to get good results this way, but the results are unsatisfactory in most of the cases except for an extremely idealized one. In most real-world ad-hoc networks, ranging by received signal strength is not accurate.

The second method of ranging is possible by measuring the signal propagation time [13] and converting it back to inter-node distance with the knowledge of velocity of the signal transmitted. Time of arrival (ToA) [18] is one such measure where the time taken for wireless signals (or packets) to travel from transmitter to receiver is multiplied by the velocity of signal (almost equal to light velocity) to obtain the inter-node distances. Radio signals travel at the speed of light (essentially instantaneous arrival), so it is not plausible to measure this time without using a high resolution clock to measure the time of flight. This is very commonly used in GPS-based ranging where the GPS receiver estimates distances using ToA from different satellites which needs time synchronization. Given the inter-node distances, techniques like multilateration can be used to locate them. To avoid complex time synchronizations between the transmitter and receiver, we can consider return time of flight wherein the receiver retransmits the signal back to transmitter. The transmitter then calculates the ToA as half the return time of

flight. But the ToA parameter is affected by latency in receiver response which may be due to processing queue at the receiver.

Time difference of arrival (TDoA) [19] is a variation of time of arrival and it is a preferred way of measuring distance by measuring the propagation time of signals. A sending node will transmit a radio signal and an ultrasonic signal at the same time. Because the radio signal arrives essentially instantaneously and the ultrasonic signal takes much longer, the receiver can measure the time difference between the arrivals, and thus deduce the traveled distance. The Cricket [20] system uses RF based TDoA ranging. One problem with ultrasound signal propagation is that it is subject to multipath effects, and to variations in the environment. It is desirable to recalibrate TDoA measurements according to these variations. Savvides *et al.* give a way to perform this calibration, given enough redundancy in the distance data. Some researchers have described the Ad-Hoc Localization System (AHLoS) [17], an iterative way of discovering the absolute position of every node in a network. They assumed an ad-hoc network, in which anchors that know their own location at any given time form some percentage of the nodes. The focus is on two-dimensional localization, and the ranging method is TDoA. Signal processing methods have been developed for localizing a set of static sensor nodes and analyzing the error properties [21, 22, 23], using both TDOA and angle of arrival (AoA) measurements where ToA measures the distances and the AoA tells about the orientation apart from positioning.

Apart from the above mentioned techniques, range-free techniques have also been used widely. An RF based proximity method was developed by [24], in which the location of a node is given as a centroid generated by counting the beacon signals transmitted by a set of beacons pre-positioned in a mesh pattern. Other methods that do not rely on range measurements were also developed. For example, the count of hops is used as an indication of the distance to the beacon

nodes in some applications [25]. But the majority of the applications rely on range based localizations.

However, most of the literature is on localization of traditional wireless sensor networks and not much has been discussed on localization in wireless multimedia sensor networks, which have more sensing modalities than traditional ones.

The calibration of multimedia sensor nodes in WMSNs is very important. In existing literature, camera calibration has been widely researched in the computer vision community and photogrammetry. Self calibration is a technique that relies on the motion of a camera in a static scene. It does not make use of any object to calibrate a camera. The camera captures a number of images of a static scene while it is moved relative to the static scene. The rigidity of the scene from the captured images provides constraints on the internal camera parameters. With fixed internal parameters the correspondence between at least three of the captured images is sufficient to recover both the internal (focal length and distortion) and external parameters. For example, Zhang [8] proposes a flexible calibration technique in which, from the captured images of a planar pattern, feature points are extracted to determine the internal and external parameters of the camera.

Most camera calibration research is for a single camera and conducted in controlled laboratory environments. Calibration of a camera network in real environments has been emerging. In [26], Kulkarni *et al.* propose a technique to overcome the constraint of using landmarks. They propose an approximate initialization technique to determine the relative locations and orientations of camera sensors without the use of landmarks. In an environment with a network of distributed cameras, this technique determines the degree of overlap and the region of overlap of each camera sensor node with the other camera sensor nodes of the network. These parameters help in

achieving a desired probability of event detection and maintain reliability in tracking a moving object. A reference object of known size at random points in the environment. The camera sensor nodes are programmed to capture images of this reference object on a duty cycle basis. The nodes estimate the degree of overlap with various sensor nodes by processing their respective images and determining the reference points in their field of view. By determining a subset of reference points that are visible to two or more cameras, the region of overlap between the cameras is given by the union of cells containing the reference points visible to each camera. The region of overlap of the camera is used to estimate the target path, and hence provide reliable information in estimating the next sensor node which has to take over the tracking responsibility. Assuming that the origin of a reference frame is located at the center of the sensor node, simple optics is used to estimate the distance between the camera and the reference point. As the size of the object as well as the values of the internal parameters is known, the sensor nodes can calculate the orientation of the reference point with respect to their center. However, this calibration technique only helps in determining the relative locations of the reference point and the orientations of the camera sensors. In this technique, the tracked target is not localized with respect to a single measurement frame but is localized with respect to the reference frame of each camera. To simplify tracking, many applications require the measurements for target localization and camera calibration to be carried out in the same frame. This can be achieved only when the target and camera sensors are present in a single reference frame.

A single reference frame based calibration technique is presented in [27]. In [27], Lee and Aghajan present a collaborative technique to localize the nodes of a camera sensor network using opportunistic observations of a target. Each node uses lightweight in-node image processing to extract the coordinates of the center of the blob of an object it detects. A sensor node that detects

an object sends trigger signals to its neighboring nodes. A neighboring node, which also detects the object, forms a reference coordinate system with the triggering node. The triggering node and its neighbor (also called the helper node) act as reference nodes. The reference nodes along with more than one uncalibrated node can participate in tracking the object. Each node uses a pin-hole camera model to determine the angle between the optical axis of its camera and the line joining the center of its camera to the center of the detected blob. All the tracking nodes collaborate with one another and exchange the tracking information. Every node then uses Gauss–Newton method [28] on the exchanged information to determine its position and orientation in the reference frame. However, this calibration algorithm relies on the assumption that all the reference nodes and the uncalibrated nodes that are participating in the tracking must view the target simultaneously. In order to perform the calibration, this method requires a minimum of three sensor nodes and at least five observations of the target be taken by each sensor node. In [42] Vimal *et al.* propose a low power and low cost wireless camera sensor network platform to perform distributed camera calibration. The sensor node uses the imaging capabilities in cooperation with moving targets to determine their own positions and orientations.

The cooperative target-based self-calibration protocol in [30] uses the target coordinate information at four locations in the field of view of a camera to perform calibration. In [30], Liu *et al.* propose an automated wireless self calibration protocol for camera sensor networks. This work combines the principles of computer vision, optics and vectors with the internal parameters of a camera to estimate the location and orientation of the camera. The work assumes that a device equipped with an ultrasound based positioning sensor moves around the environment and generates a set of reference points in the field of view of each camera that it encounters. To perform calibration the camera is provided with the location information of four reference points

in its field of view by the moving object. The sensor node then determines the vectors joining its centre to each of the reference points. It then employs a non-linear optimizer to estimate the location of the camera. After determining its location, the sensor node uses a subset of three reference points to determine the orientation of its camera. The obtained external parameters of the camera are iteratively refined by using additional reference points in the field of view of the camera. The external parameters are now used to determine the region of overlap which helps in localizing and tracking the moving object.

Overall, the existing camera calibration algorithms are either time-consuming, fit only in laboratory environments, or require specific cooperative or non-cooperative targets, which may not be realistic in many applications. Though our work involves network camera calibration it is different than the proposed approaches mentioned above in several ways. The first difference is the multimedia sensor node makes use of local inter-node distances and network topology as well as vision information to determine the coordinates of location of the nodes. The second difference is that most authors need to have a moving target for calibration; our method needs no such target as the sensor nodes develop the ability to self organize themselves globally.

The remainder of this thesis is organized as follows. In Chapter 2, we discuss the design of the wireless multimedia sensor nodes we are developing in our laboratory and formulate the WMSN location calibration problem and propose an overall solution. In Chapter 3 we develop the detailed algorithm needed to solve such a challenging problem. We then discuss the simulation results in Chapter 4 and Chapter 5 concludes the thesis.

CHAPTER 2

OVERVIEW

Chapter 2 formulates the problem of distributed camera calibration, and presents the overall approach to it. Then it explains the concept of multidimensional scaling, the algorithm for multidimensional scaling based localization and investigate the factors that impact the performance of multidimensional scaling based localization technique.

2.1 Problem Formulation

We are developing a Wireless Multimedia Sensor Network (WMSN) for military surveillance applications. This surveillance network consists of a large set of sensor nodes which have omnidirectional vision, audio input and output, computation, and wireless communication capabilities. By forming a network they can provide battle-field awareness to military personnel. Below we will first introduce the design of the sensor node and then we will formulate the problem of location calibration in this WMSN.

As shown in Figure. 1, each sensor node is outfitted with an omnidirectional video camera, an array of microphones and speakers, a compass, modest computing resources (PDA-level CPU), and wireless communication capabilities (250 Kbits/sec data rate). The sensor nodes can be deployed by a soldier and/or a ground vehicle or airplanes. The sensors have the ability to self-organize which implies that the sensor calibration and the establishment of communications will be performed autonomously. The form-factor of the sensor is relatively small, and so they can be easily carried and deployed. The dimension of the sensor is close to that of a human fist and its weight is less than two pounds. The Figure 2 shows an example of the omnidirectional camera at ASCC lab.

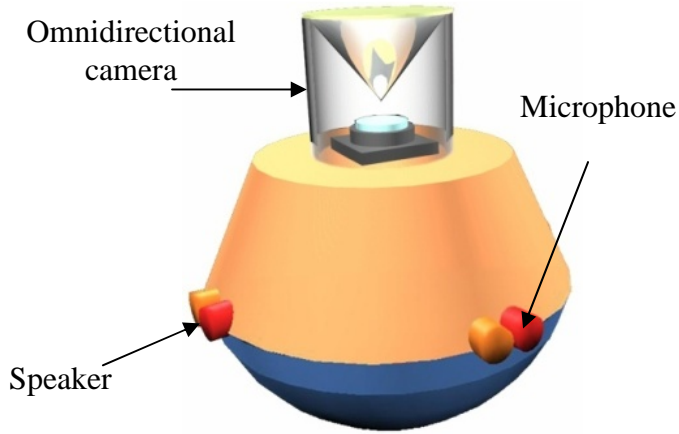


Figure 1: The design of the multimedia sensor node



Figure 2: Omnidirectional Camera

We consider n multimedia sensor nodes randomly deployed in a region as shown in Figure 3. The node N_i has two neighborhoods defined, as can be seen in Figure 3. N_i can communicate with its neighbors and form a *communication neighborhood*. Similarly, node N_i can see the neighbors within vision range R_v through its omnidirectional camera. These nodes form its *vision neighborhood*. We assume that each node can measure the distance to its one hop neighbor in the communication neighborhood. Such a capability can be realized through the Time Difference of Arrival (TDoA) technique, which is based on the time difference between an acoustic signal and a RF signal when they both travel from one node to the other.

The problem of location calibration in an WMSN is to find the relative locations and heading of all the nodes in the deployed sensor network. If we have at least three beacon nodes we can find the absolute locations of all the nodes in the deployed network.

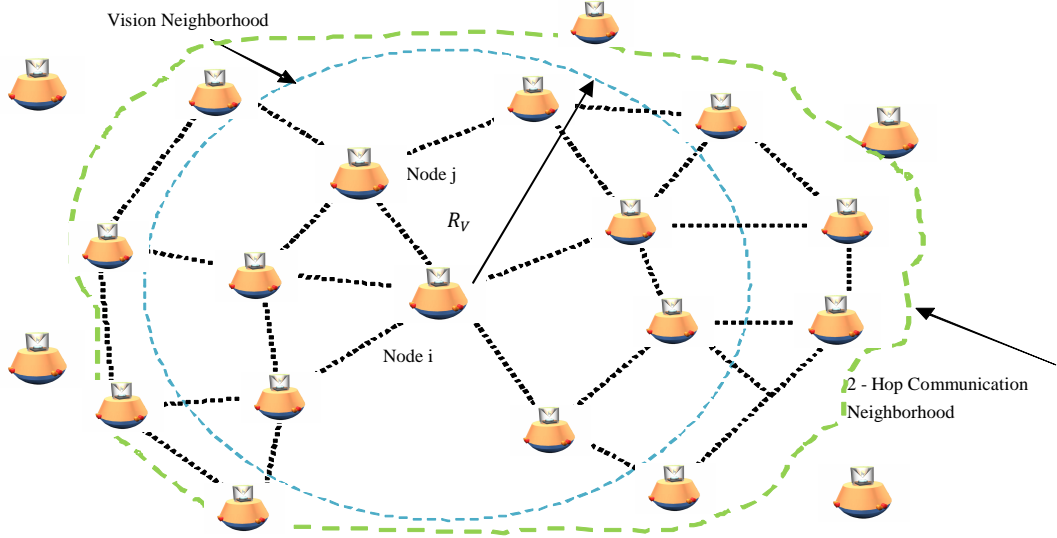


Figure 3: The self-calibration of sensor node with respect to node ‘i’

In this work we make the following assumptions:

1. There are no obstacles in the region described, so any sensor node within a distance R_v to N_i can be observed by N_i .
2. The nodes use a compass to know their heading.
3. The environment is considered to be flat and treated as a two dimensional space.

2.2 Overall Approach For Distributed Camera Calibration

In our proposed approach, distributed camera calibration algorithm consists of two main steps which involve local map generation and global map generation. First we generate the local maps for each sensor node. We then select a node with the least ID to start with and validate its local map to know whether it is a good base node to start with. We then select the next best node for merging which has the maximum unknown IDs and at least three common IDs. We then solve the map merging problem to get the global map of sensor locations.

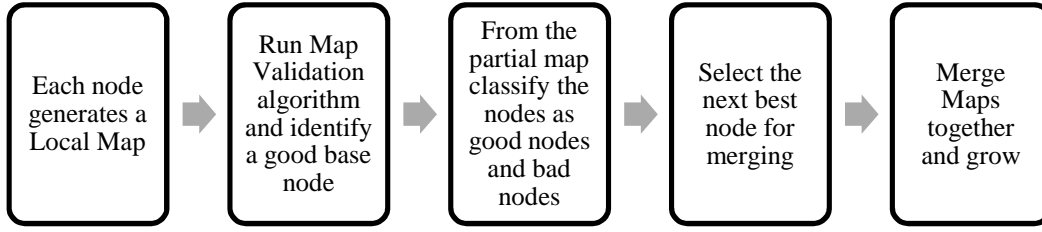


Figure 4: Overall Approach

As illustrated in the Figure 4 we now describe the overall approach for distributed camera calibration. In the local map generation problem, each node first generates a vision map from the omnidirectional image and then estimates the location of each neighboring node within the vision range R_v . However, since all the sensor nodes have the same shape and color, their IDs cannot be determined on this vision map, which means that the vision map provides us with the information of relative location of the nodes with a reasonable accuracy but with no ID information. Next, each node generates an ID map using the popular multidimensional scaling (MDS) algorithm [13, 31] as described in chapter 1. Since each node can measure the distance to its neighbors in the one-hop communication neighborhood, it can construct a distance matrix for its 2-hop communication neighborhood and feed it to the MDS algorithm to create a relative map of IDs. However, the MDS algorithm is not very accurate since it depends on the shape of the network topology and the density of the network. Therefore, the ID map provides us only with the information of the node IDs but not with accurate information of the relative location of the nodes. In this sense, the ID map and the vision map complement each other.

Therefore, the local map generation problem is essentially an ID association problem that is to associate the IDs in the ID map to nodes in the vision map. To solve the ID association problem, we propose an algorithm based on the iterative closest point (ICP) algorithm [32].

The local map obtained is not always correct since it returns erroneous results when there is symmetry in the placement pattern. The symmetry in the placement of the nodes results in multiple solutions to the ID association problem and hence the global minima obtained may not be correct. However, we can solve this problem by introducing the reflection matrix and recalculate the global minima. We then use the map sharing algorithm to further validate local maps. Validation is needed since there are irregularities in the MDS resulting from the arbitrary noise in the translation matrix returned from the classical MDS result. We hence introduce the concept of good nodes. Local maps are said to be good when the Mean Square Error metric obtained from the map sharing algorithm is less than a predefined threshold value. Nodes with good local maps are then defined as good nodes.

We then solve the map merging problem. We do this by searching for the next best node from the set of good nodes, to merge with the initial arbitrarily selected node (base node) and thus expand its local map. The next best node is the one with a local map that has the maximum number of unknown IDs and a minimum of at least three common IDs. Then we merge these two maps together. This process is repeated until all the nodes in the network are covered.

The overall algorithm for the location calibration is summarized as follows:

- 1. Each node generates a local map*
- 2. A node with the least ID selects itself to be a base node N_b . The base node checks whether its local map is a good local map using the map validation algorithm.*

3. *If the base node local map is identified to be bad we discard the node and randomly select the next base node from the set of neighboring nodes and repeat step 2-3*
4. *Until a good base node is found, classify nodes as good and bad nodes from the base node's neighboring nodes.*
5. *Search for the next best node for merging, from the set of good nodes in the current partial map of N_b and, request for the local map from the node selected*
6. *Merge the local maps from node N_b and the node selected.*
7. *Update the current partial global map for N_b and repeat steps 5-6 until all the nodes in the network are localized.*

2.3 Multidimensional Scaling

Multidimensional scaling is a popular technique for localization in wireless sensor networks since it uses the network topology and inter-node distance information to generate relative position map of 2 dimensions. In this section we explain the concept of MDS, types of MDS, mathematical background of MDS and algorithm for MDS based localization. Lastly, we investigate the factors that impact the performance of MDS algorithm.

2.3.1 Introduction to Multidimensional Scaling

The roots of the Multidimensional Scaling [31, 33, 34] or MDS lie in the behavioral sciences like Psychometrics and Psychophysics wherein the personal traits of people are analyzed for important underlying distinctive characteristic features. Subsequently MDS proved to be an essential tool for many other researchers in diverse fields like Marketing, Sociology, Geography, and Psychology. Basically MDS is a data visualization algorithm that can describe the structure of the data. It involves multivariate statistical probing to describe proximity between the pairs of objects with the proximity data collected over time.

The term ‘proximity’ is an index defined over a pair of objects to quantify the degree to which the two objects are alike or different. Correlation coefficient, joint probabilities are two such examples of proximity measures which can explain the extent to which two objects show common attributes. A proximity measure helps in differentiating the objects and hence it can indicate either similarities or dissimilarities. Hence the term ‘proximity’ has varied contextual meanings based on the application in which the data visualization algorithms are employed. In general usage, the term proximity indicated by δ_{ij} indicates the dissimilarity between the objects i and j

The Multidimensional Scaling algorithm takes the proximity measures as the input. And the chief output is a spatial representation, consisting of a geometric configuration of points. Each point in the configuration corresponds to one of the objects and the configuration as a whole reflects the hidden structure in the data making them easier to comprehend. This implies larger the dissimilarity between the two objects in comparison, the farther apart they would be placed in the spatial map.

Table 1: Inter-city distance matrix

	1	2	3	4	5	6	7	8	9	10
1	0	569	667	530	141	140	357	396	570	190
2	569	0	1212	1043	617	446	325	423	787	648
3	667	1212	0	201	596	768	923	882	714	714
4	530	1043	201	0	431	608	740	690	516	622
5	141	617	596	431	0	177	340	337	436	320
6	140	446	768	608	177	0	218	272	519	302
7	357	325	923	740	340	218	0	114	472	514
8	396	423	882	690	337	272	114	0	364	573
9	570	787	714	516	436	519	472	364	0	755
10	190	648	714	622	320	302	514	573	755	0

MDS starts with a matrix representing the distances or dissimilarities between ‘n’ objects. The power of this algorithm lies in its ability to depict the dissimilarities or the proximities between the objects through a placement of points in a low dimensional plane where the Euclidean distances between the points resemble the actual proximities between the objects as closely as

possible. The best ever way to demonstrate the capabilities of MDS is by illustrating the classic example of cities and geographic map of Europe which is very widely used.

Consider the inter-city distance matrix shown in table 1. Let us suppose that we know the inter-city distances accurately for 10 popular cities of Europe. Essentially this would be a 10 by 10 symmetric distance matrix with the principal diagonal being zeros (distance of a city to itself is always zero). The actual distances in miles are given by the following matrix:

The numbers 1 through 10 correspond to each of the 10 cities shown on the Figure 5. This distance matrix is fed to the MDS. The distances are first scaled down suitably with a scaling factor. Now the 10 cities are placed in a 2-D coordinate axes in such a way that their Euclidean distances match very close to their scaled distances and the resulting centroid of the entire configuration of points is at the origin. This forms the Relative Map. Now we may also have specific idea on some of the cities. For instance, assume that we already know the geographical locations of at least three cities say Stockholm, Madrid and Rome. We can now translate and rotate the relative map in such a way that the relative locations of the above mentioned three cities conform very closely to the absolute locations. In this process, we find that the rest of the cities also reach their absolute locations. The accuracy depends on the precision of inter-city distances and the transformation of relative map to absolute map.

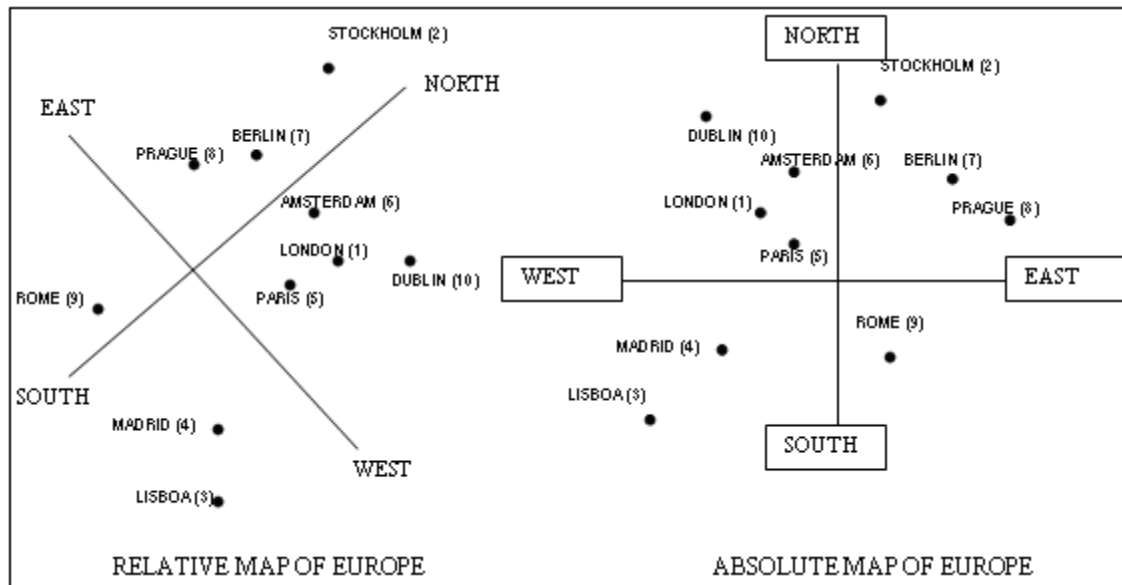


Figure 5 : Relative and absolute maps of cities of Europe

2.3.2 Types of Multidimensional Scaling

By now, we already understood that the MDS depends heavily on the proximity measure input in the dissimilarity matrix. In the above example, the inter-city distance was the proximity measure. While 'distances' are numerical figures, there are many other types of proximity measures also. Basically, the proximity variables can be divided into four broad categories:

1. Nominal Scale : Classificatory data with no comparisons possible .(Ex: Gender)
2. Ordinal Scale : Comparable data with no quantitative measure(Ex: Grades A–E)
3. Interval Scale: Difference in two values are meaningful but no zero (Celsius scale)
4. Ratio Scale: Same as Interval, but with defined zero.(Kelvin Scale)

Based on these four types of proximity measures, the classical Multidimensional Scaling is further classified into two types:

1. Metric Multidimensional Scaling
2. Non-Metric Multidimensional Scaling

Metric Multidimensional Scaling deals with the Interval and Ratio variables. This would include most of the models that deal with numerical scores, distances and other quantitative measures. This thesis work uses inter-subject distances and hence we will be using on Metric Multidimensional scaling only. The Non-metric Multidimensional scaling deals with the other two types of variables, nominal and ordinal. Mostly this model finds application in subjects involving abstract issues like behavior patterns, affinity determination and other issues which are hard to be quantified. This type of MDS is mostly used in Psychology, Marketing and other related fields. Now onwards we will be dealing with only Metric or Classical Multidimensional Scaling in which the Proximity variable refers to the distance between the subjects.

2.3.3 Algorithm for MDS-Based Localization

In the case of localization problem, the dissimilarity measure of N subjects is an N x N distance matrix. We now discuss the classical multidimensional scaling method first and then discuss the steps for localization using MDS. Let the proximities between any pair of nodes (r, s) be indicated by δ_{rs} where r, s = 1, 2...n and n is the total number of subjects. If the dissimilarities and the distances between the subjects are to be precisely Euclidean distances then Classical Multidimensional Scaling finds a configuration of points ensuring the equality

$$d_{rs} = \delta_{rs} \quad (1)$$

where d_{rs} is the distance between the two subjects in the configuration. Generally the above equation is not a strict equality and through the configuration of points, MDS always tries to minimize the loss function given by

$$\text{Loss Function} = ((\sum (d_{rs} - \delta_{rs})^2) / \sum (d_{rs}^2))^{1/2} \quad (2)$$

Given the above constraints on the inputs, the MDS can then plot these points with origin as the centroid. To get a perceivable output, there must be just 2 or 3 dimensions which are good enough to contain most of the information. Hence singular value decomposition is carried out on the distance matrix and only those dimensional are preserved which convey most of the information. In mathematical terms, these are the dimensions which are associated with correspondingly largest eigen values.

In summary, the localization problem can be addressed by the following steps using MDS [35]

1. The shortest path between the pairs of nodes is computed. The distance measurement capacity of a node is limited by its communication range. A node can measure distances to its neighbors only and for the rest of the nodes which fall outside the communication range, an “infinite” value is assigned as the distance. (Typically “Infinite” takes the values of few tens of thousands so that it is always contextually large figure)
2. Floyd’s Algorithm is used to compute the shortest paths between any pair of nodes using the connectivity information.

Floyd’s Shortest Path Algorithm

This algorithm, which is also referred to as Floyd-Warshall’s algorithm [36], compares all possible paths through the connected graph between each pair of vertices. It does so by incrementally improving an estimate on the shortest path between two vertices, until the estimate is known to be optimal. Floyd’s shortest path algorithm uses a technique called Dynamic Programming [36] to solve the all-pair shortest path problem. The following explains the procedure involved.

The first step is to create an Adjacency matrix which is computed for any paid of nodes (i, j) as follows:

$$A(i, j) = \begin{cases} 0 & \text{if } i = j \\ d_{ij} & \text{if } i \neq j \end{cases} \quad (15)$$

Note that $d_{ij} = \infty$ when i and j are more than 1-hop away

The adjacency matrix entries are recursively updated by the following function that searches exhaustively for all possible paths and picks the shortest path. The variable k indicates the possible number of iterations and there could be at the most $k-1$ intermediate nodes in between i and j in any iteration. The recursive function is given by the following expression with k equal to $(1, 2 \dots n)$

$$d_{ij} = \begin{cases} A(i, j) & \text{when } k = 0 \\ \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}) & \text{when } k > 0 \end{cases} \quad (16)$$

This leads to a final updated version of Adjacency matrix with respective shortest path distances as the matrix elements. The computation complexity is given by n^3 .

Dijkstra's algorithm is another popular shortest path algorithm that can be used in this application. But Floyd's algorithm is more robust and involves lesser computational overhead in large networks. Moreover practical experience also indicates that Floyd's algorithm is faster than Dijkstra's algorithm in MATLAB simulation. [35]

3. The symmetric distance matrix obtained in the above step is input to the Classical MDS algorithm. As mentioned earlier, the CMDS does singular value decomposition and eliminates dimensions corresponding to non-significant eigen values, thereby constructing a relative map with 2 or 3 dimensions. An optional refinement step involving least-squares minimization can be included to best conform the inter-distances of the nodes to the measured distances.

4. The relative map obtained can be transformed into an absolute map, if provided with the minimum number of anchor nodes, (3 nodes for a 2-D and 4 nodes for a 3-D networks). First, a transformation function is created by mapping the relative coordinates of the beacons with their known absolute coordinates. This might involve some translations and rotations. The obtained transformation function is then applied to the rest of the nodes. An optional refinement step involving least-squares minimization can be included to conform the inter-distances of the nodes to the measured distances.

2.3.4 Performance of MDS-Based Localization

The performance of classical Multidimensional Scaling based localization is determined fundamentally by the network topology parameters. It was observed that the density of the network has direct relationship with the performance. Simulation results show that the denser networks exhibit less mean localization error. The second distinctive performance parameter is the shape of the network topology. If the nodes were deployed in a uniform pattern, the results were better. Contrastingly, irregular deployment increases the error and especially c-shaped networks yield highly unsatisfactory results.

In the view of the parameters identified, the MDS-based localization procedure is realizable in most of the cases where the nodes are deployed densely and regularly in a static network. Once deployed, the nodes do not change their locations and hence, the regularity of the network can possibly be addressed by a proper initial deployment of nodes. But the density is fixed and moreover this approach cannot be extended to a dynamic network where the nodes move around randomly and might end up forming an irregular network in due course of time.

The performance of the MDS is restrained by the density of network and regularity in the locations of the deployed nodes. Hence this situation calls for an algorithm which comes into

picture once the nodes are deployed in a grid based network. It should be able to accommodate the issues of density and regularity to the best, though irregular networks are always a problem. These two issues are addressed in the Chapter 3.

CHAPTER 3

DETAILED ALGORITHMS FOR DISTRIBUTED CALIBRATION OF WMSN

Chapter 3 develops the detailed theoretical framework for performing the distributed calibration of wireless camera sensor nodes. First we discuss the local map generation algorithm which involves vision map generation and ID map generation. Secondly we examine the map validation algorithm and lastly we explore the map merging algorithm.

As described in the chapter 2 we find the global minimum angle to generate the local maps by fusing the vision map and ID map and further refining the results using the iterative closest point algorithm (ICP) [26]. Then the node with the least ID selects itself as the base node and confirms whether it is a good node to start with. Once confirmed the node searches for the next best node to merge with and then performs the map merging algorithm to grow its local map to cover the entire environment. Also further analysis is done to account for the uncertainties in the MDS algorithm due to network topology and density of the network and thus develop algorithms for addressing the same innovatively.

3.1 Local Map Generation

3.1.1 Vision map generation

The original idea of the omnidirectional vision sensors (ODVS) using a mirror in combination with a conventional imaging system has been proposed by Rees [37]. The idea makes use of a hyperboloidal mirror for acquiring an omnidirectional image that has a single center of

projection. That is, the omnidirectional image can be transformed into normal perspective images.

1990's, saw the emergence of computer technologies which enabled real-time process of vision data and researchers made several types of omnidirectional vision sensor as vision systems for computers and robots. Yagi and Kawato [38] made an omnidirectional vision sensor using a conical mirror. Hong *et al.* [39] made an omnidirectional vision sensor using a spherical mirror. Their purpose was to navigate mobile robots with the omnidirectional vision sensor. The omnidirectional vision of a robot is convenient for detecting moving obstacles around the robot and for localizing itself. Then, Yamazawa and others [40] made again an omnidirectional vision sensor by using a hyperboloidal mirror. By utilizing the merit of the hyperboloidal mirror that the omnidirectional image can be transformed into perspective images, they proposed a monitoring system with the ODVS.

Nayar and Baker [(Baker, 1997)] theoretically analyzed imaging systems using mirrors and developed an ideal omnidirectional vision sensor using a parabolic mirror and a tele-centric lens. The omnidirectional vision sensor using a hyperboloidal mirror can generate an image taken from a single center of projection in combination with a standard perspective camera. However, it has a disadvantage that one of the two focal points of the hyperboloid has to be set on the camera center [42]. This disadvantage makes it difficult to design the ODVS. On the other hand, the imaging system proposed by Nayar and Baker does not have such a disadvantage since it is using the tele-centric lens. As known well, the parabolic mirror has a focal point for light which is parallel to the main axis of the parabolic mirror. Further, the imaging system is superior in acquisition of non-blurred images and it can eliminate internal reflections of a hollow cylindrical or spherical glass which supports the mirror.

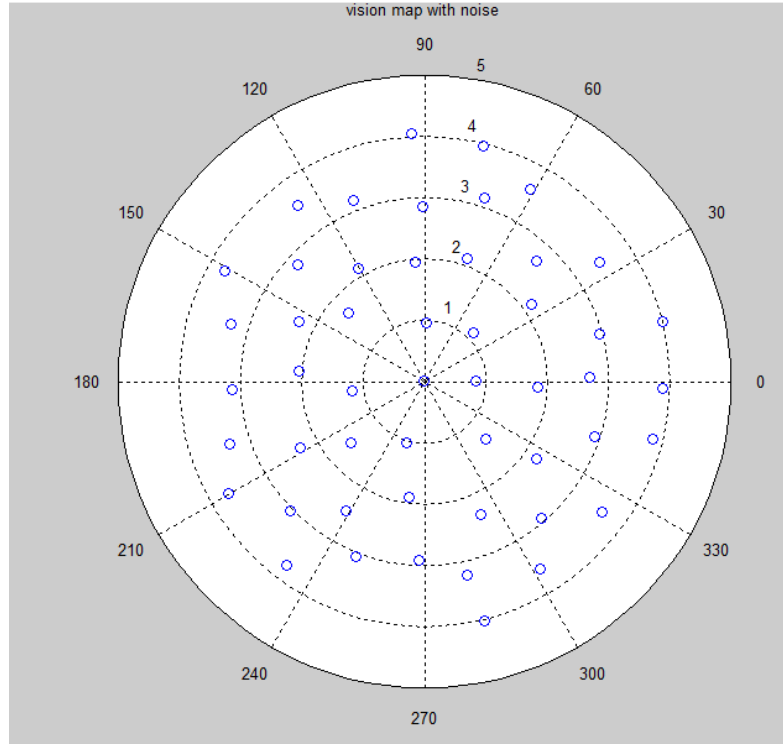


Figure 6 : A typical vision map with range and angle errors

In the proposed system each sensor node takes an image of the environment using its embedded omnidirectional camera. Figure 6 shows a simulated vision map. We provide random orientation to the vision maps in order to simulate different orientation for each vision map.

From this image each node gets the relative distance and angle information (x , y , and θ for the camera) of each of its vision neighboring nodes. The distance information can be calculated based on the size of the image of the sensor node [43]. We assume that the distortion correction on the omnidirectional images has been done. We further assume that in the omnivision camera the mirror is facing downwards and hence we can see the neighbors in the vision map. Based on the above information, we can estimate the location of each node within the vision neighborhood.

3.1.2 ID map generation

As explained in chapter 2 Multidimensional Scaling (MDS) algorithm is a popular statistical tool used to depict the dissimilarities or proximities between the objects through a placement of points in a low-dimensional plane where the Euclidean distances between the points resemble the actual proximities between the objects as closely as possible. We generate a distance matrix which is a dissimilarity measure for the sensor nodes within a two hop communication neighborhood. The distance matrix which is fed to the MDS algorithm must be a symmetric matrix with zeros on its principal diagonal. Given the above constraints on the inputs, the MDS can then plot these points with origin as the centroid.

Based on the classical MDS algorithm we discussed in Chapter 2 section 2.3.4, the method to generate the ID map consists of four steps which involves collecting the inter-node distance information, finding the shortest path using the Floyd's algorithm, input this distance matrix to the MDS algorithm to generate a relative map of 2 dimensions and perform an optional refinement step to further conform to the inter-node distances. Figure 7 illustrates a typical ID map.

Through the above steps, an ID map is generated in the two-hop neighborhood, which will be associated with the vision map of that node.

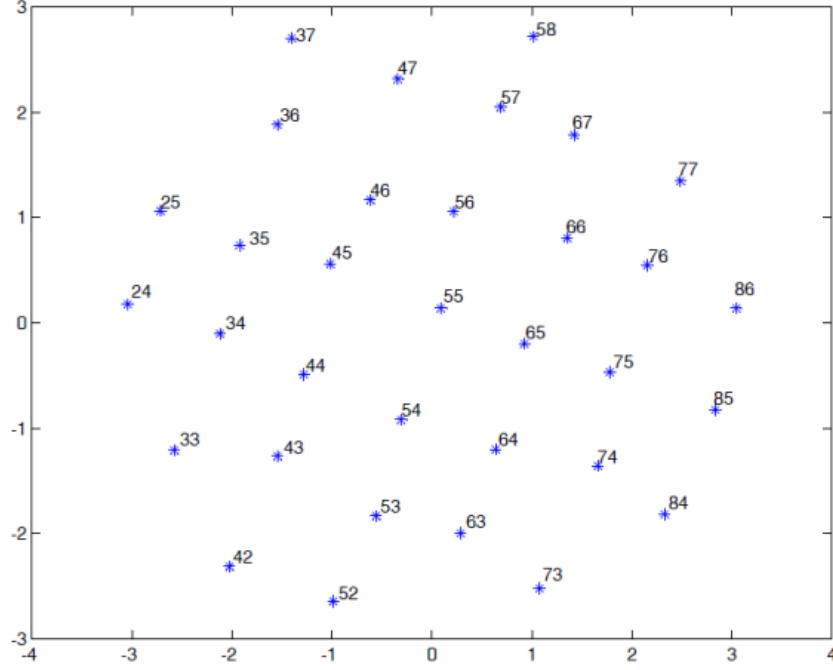


Figure 7: A typical ID map

3.1.3 ID Association Algorithm

We propose a map registration approach to solve the ID association problem. This approach has two steps. In step 1, we conduct a pre-registration to roughly align the ID map with the vision map. In step 2, we refine the registration with the ICP algorithm.

Step 1: Pre-registration

In this step we associate the nodes in the vision map (x_v, y_v) and ID map (x_m, y_m) to find the best possible correspondence. This is done by searching the global minimum angle which can give the rough association.

To find the global minimum angle, we perform the following steps:

1. Translate the ID map in order to have an overlapped centroid with the vision map.
2. For $\theta = 1:360$ with increment of 1 degree do:

Rotate the ID map by θ degree;

Calculate the corresponding pairs between the nodes in the ID map and in the vision map based on the closest distance.

Calculate the sum of the square of the distances between the corresponding pairs and find the minimum sum and its \emptyset .

$$\min \sum ((x_v - x_m)^2 + (y_v - y_m)^2)$$

3. *Calculate the rotation matrix:*

$$R_{pre} = [\cos\emptyset \sin\emptyset, -\sin\emptyset \cos\emptyset]$$

4. *For reflection due to symmetry*

$$R_{flip} = [\cos\theta \sin\theta, -\sin\theta \cos\theta],$$

where $\theta = 180$ degree

$$IDmap = IDmap * R_{flip} \text{ (this gives } x = -x, y = y)$$

5. *Recalculate the rotation matrix:*

$$R_{prenew} = [\cos\theta \sin\theta, -\sin\theta \cos\theta],$$

where $\theta = 0:360$ with increment of 1degree

6. *Recalculate the sum of the square of the distances between the corresponding pairs and find the minimum sum and its θ .*

7. *Compare the minimum sum for R_{prenew} and R_{pre} using θ or \emptyset and select the appropriate rotation matrix for the new ID map generation.*

8. *Generate the new ID map:*

$$IDmap_{new} = (R_{pre}) \text{ or } (R_{prenew}) * IDmap$$

Through the above steps, the ID map is rotated, reflected and brought close to the vision map, which will be further rotated in Step 2 for refinement.

Step2: ICP based refinement

After the nodes in the vision map and in the relative ID map have been roughly associated in the above step, we run the iterative closest point algorithm [26] to refine the correspondence of the two maps to a higher degree of accuracy.

The ICP algorithm finds the rotation and translation matrix between the preregistered ID map and the vision map by minimizing the least square errors among the closest node pairs. Based on the refined registration, we can improve the ID association between the nodes in the ID map and in the vision map.

3.2 Local Map Validation

After the ID association and the ICP registration algorithm, the nodes need to know that their local maps are correct. This ambiguity is mainly due to the uncertainty in the MDS result.

In order to solve this problem we propose map validation technique.

The node validating its local map requests for the local maps of the neighboring nodes. The common IDs in the local maps are subject to rotation, reflection, scaling and translation through the use of procrustes function [18]. We then calculate the MSE value for the node validating its local map and its neighbor. If the MSE value obtained is less than $1r$ units we define the node as a good node, where 'r' is the normalizing unit distance. If the MSE value is more than $1r$ units.

We can summarize the algorithm in the steps below:

1. *Select the node which needs to validate its local map.*
2. *Run the ID association algorithm to calculate the local map*
3. *To check the local maps do:*

- *Neighboring nodes share the local maps with the node that needs to validate its local map*
- *Compare the Mean Square Error metric between the original IDs in the new local map and the node that needs to validate its local map*
- *The mean square error metric is calculated between the original IDs and the IDs in the local map of the node that needs to validate its local map*
- *Check the magnitude of MSE value. if it is less than $1/r$ units, where 'r' is the normalizing factor then we can declare the node as a good node. A good node is a node that has a good local map.*

3.3 Map Merging

Each node knows whether it has a good local map after solving the local map generation problem and then running map validation algorithm as discussed above. Then we address the problem of base node selection and merging or growing the local maps by obtaining the best linear transformation from the set of good nodes to transform the coordinates of the common nodes in the base node local map to those in the neighboring nodes local maps. This enables us to localize all the nodes in the network. We call this *the map merging algorithm*.

We select the node with the least ID (node with ID 1) as a base node to start with. We then run the map validation algorithm to know whether it is a good base node to start. If we validate the local map as a good node then we select node 1 as the base node. If we are not able to validate, we randomly select another node from the set of neighboring nodes of node 1 until we find a good node to start with.

We define the following variables:

$M_b : \{ (x, y)_b, ID_b \}$ The partial global map of node N_b .

ID_b : The IDs in the local map of N_b

$(x, y)_b$: The local map coordinates of node N_b

N_j : The next best node selected for merging

$M_j : \{ (x, y)_j, ID_j \}$ The local map of node

ID_j : The IDs in the local map for the next best node N_j

$(x, y)_j$: The local map coordinates for the node N_j .

$(x, y)_j^b$: The local map coordinates for node N_j in node N_b coordinate frame.

The following steps are involved in the map merging algorithm:

1. For base node N_b , find the neighboring node that has a local map with the maximum number of unknown IDs and minimum number of three common IDs.
2. Request for the local map from the selected node.
3. Find the linear transformation by using the Procrustes function. Transform the next best node map M_j into node N_b 's coordinate frame to get M_j^b using:

$$(x, y)_j^b = (x, y)_j * \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix} + (T_X \quad T_Y)$$

where, R is a combination of the rotation and reflection factor and T is the translation factor

4. Update the partial global map M_b using

$$M_b = M_b \cup M_j^b$$

5. Repeat steps 1-4 and thus update the partial global map until it contains all the nodes.

CHAPTER 4

SIMULATION RESULTS

In Chapter 4 we first present in detail the simulation environment. We then discuss the results obtained and evaluate the performance of the proposed framework. Different scenarios have been discussed which investigate the local map generation, base node selection, and map merging concepts developed previously in Chapter 3.

4.1 Simulation Setup

We conducted computer simulation to evaluate the proposed approach. The simulation is carried out in MATLAB. The PC used is a standard Dell Inspiron with an INTEL core 2 Duo T5750 @ 2GHz processor and a 3GB RAM.

In our simulation, nodes are placed in a square area. We model the placement errors as Gaussian noises. In this way, we can create a random sensor distribution mimicking, for example, airplane deployment. One hundred nodes are placed on a $10r \times 10r$ grid; with a unit edge distance r . Figure 8 shows an example of the deployment with 10% placement error. The radio range is $R = 1.95r$, which leads to an average connectivity of 8.14. In the graph, the points represent the sensor nodes and the edges represent the one-hop connectivity between neighbors.

4.2 Results

In the results section we examine the performance of the algorithms discussed in chapter 3. First we examine the results for local map generation which includes pre-registration and refinement due to ICP algorithm. Also we tabulate the results of ID association in local maps for different vision ranges. In the second subsection we evaluate the map validation algorithm. We tabulate

the results for the same. In the third subsection we investigate the map merging algorithm and observe several simulations illustrating the same. We then tabulate the results calculating the accuracy metric at each stage of map merging. We have performed simulation tests on the above described irregular grid network.

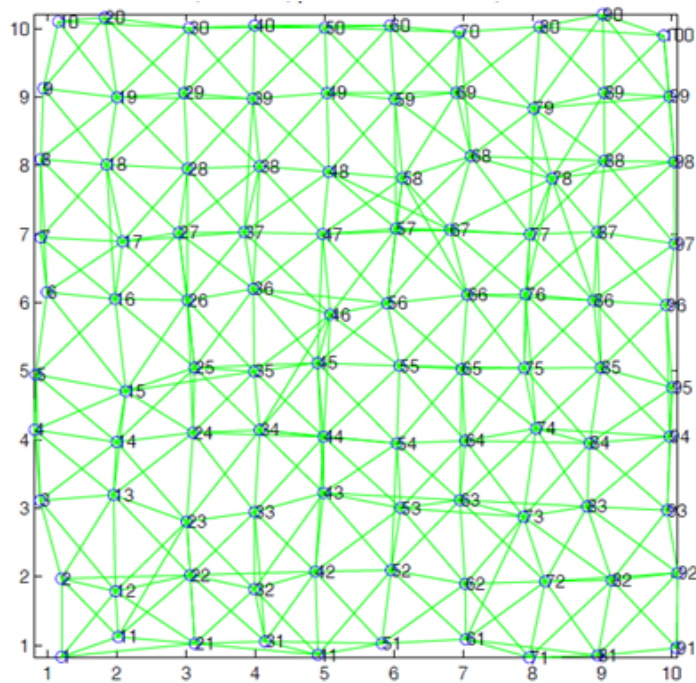


Figure 8: The ground truth of the sensor deployment (100 nodes, $R=1.99r$, placement error=10%)

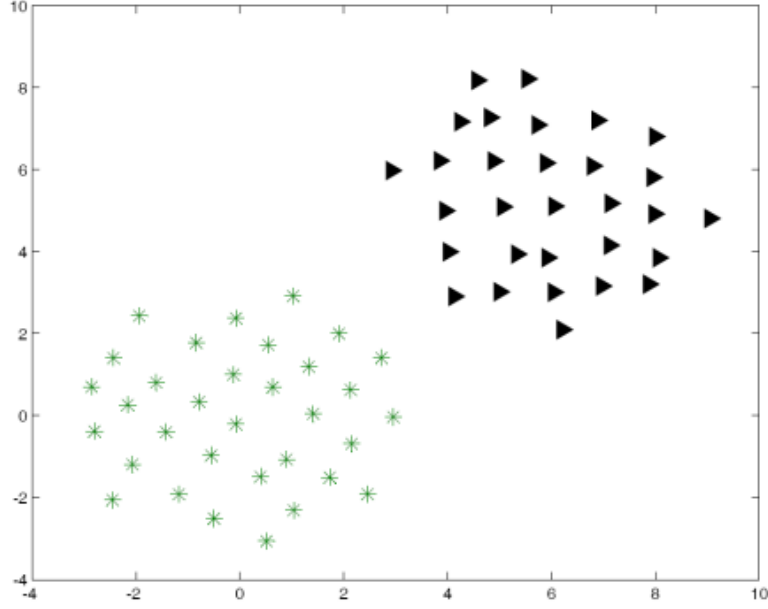


Figure 9: The ground truth for ID map (green stars) of the sensor nodes and the vision map obtained (black triangles).

4.2.1 Local Map Generation

We now observe the results for local map generation algorithm. In the first evaluation scenario the node 55 was selected as the good base node to start with. Figure 9 shows the result of the ID map (green stars) based on the two-hop communication neighborhood of the base node. It can be seen that the center of the ID map is the origin (0, 0), and it has a different orientation than the original network.

Figure 10 shows the result after performing pre-registration and associating the IDs in both the ID map and in the vision map for node 55. It can be seen that there is still a significant distance error between the corresponding pairs in the two maps for base node ID 55. In Figure 11, the result shows the IDs in the ID map match the ground truth IDs of vision map after performing the ICP refinement algorithm for base node ID 55.

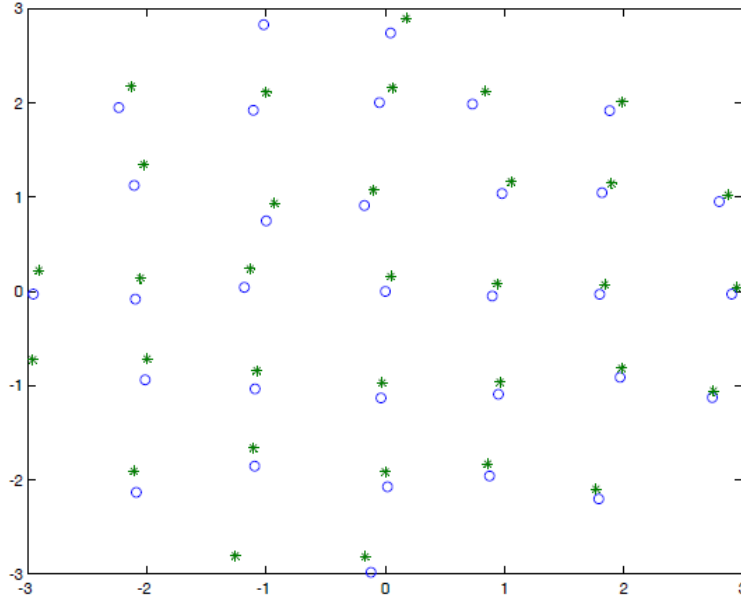


Figure 10: ID Association using pre-registration. The nodes in the vision map are brought closer to the nodes in the ID map for node ID 55

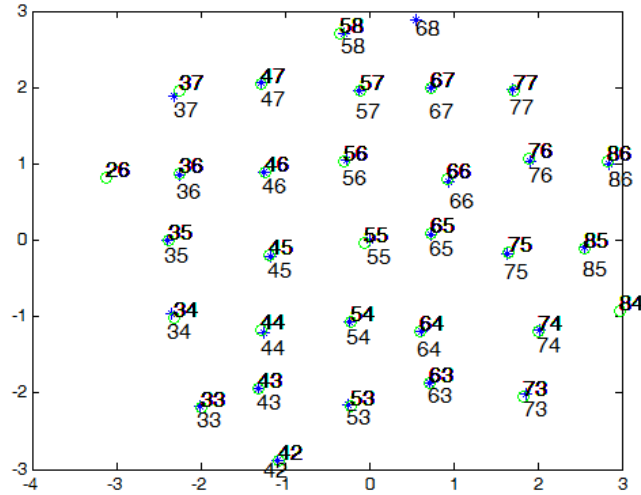


Figure 11: The map shows the correct ID association between the ID map and the vision map after performing the ICP refinement algorithm for node 55. The IDs shown in the figure indicate that they have the perfect match.

We now consider a different example for the local map generation algorithm. In the Figure 12 we illustrate the pre registration step for node ID 66 and in Figure 13 we demonstrate the refinement step due to ICP algorithm.

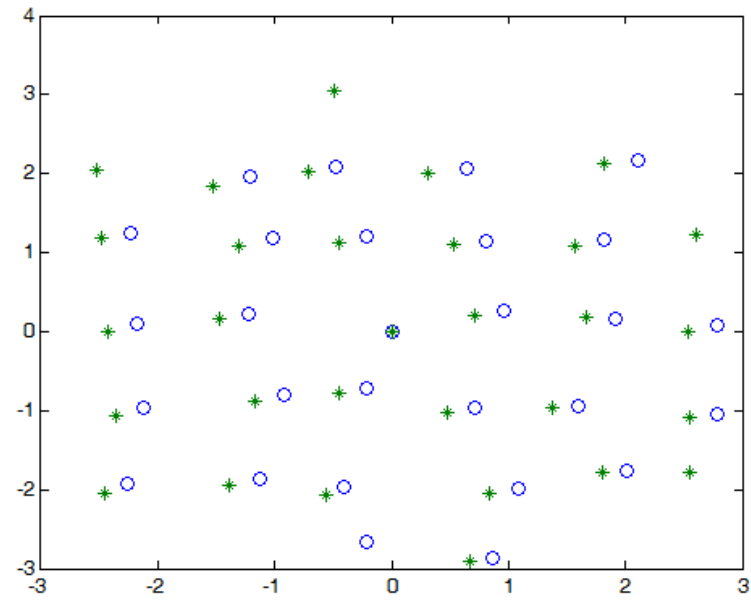


Figure 12: Pre-registration example. Before the implementation of iterative closest point algorithm for node ID 66

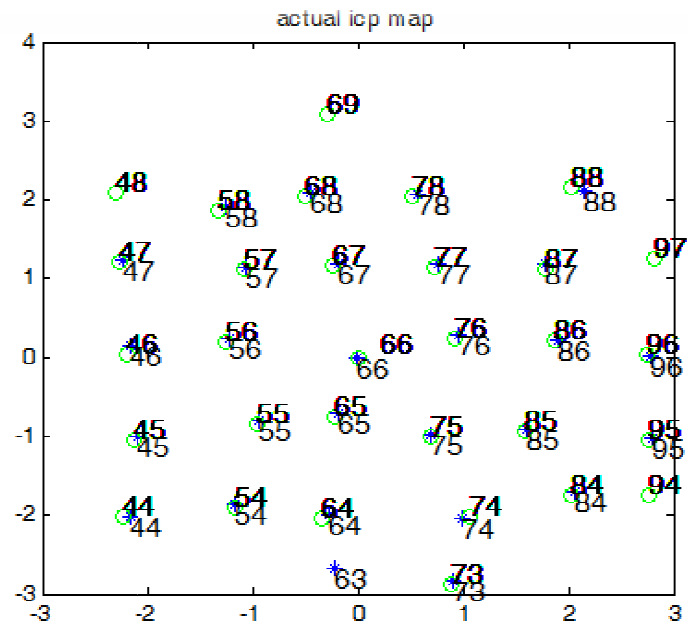


Figure 13: The map shows the correct ID association between the ID map and the vision map after performing the ICP refinement algorithm for node ID 66.

In the third scenario the node ID 33 was selected as a good base node to start with. The pre-registration step in obtaining the local map is illustrated in Figure 14 and the result after ICP refinement is shown in Figure 15.

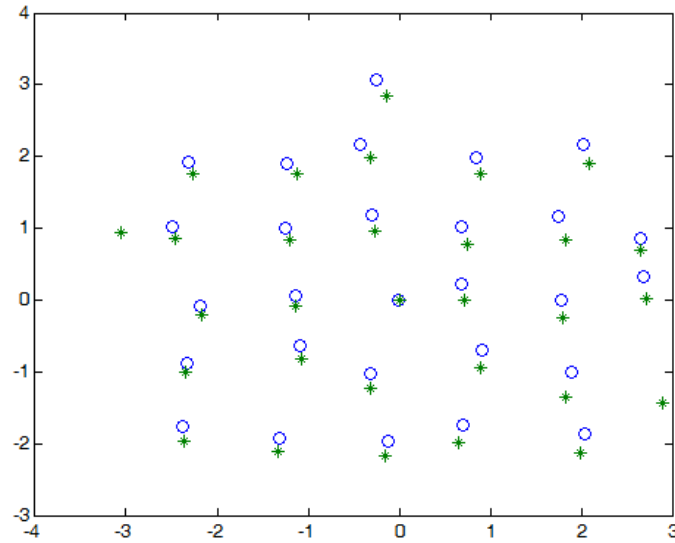


Figure 14: Pre-registration - Nodes in Vision map brought close to nodes in ID map for base node 33

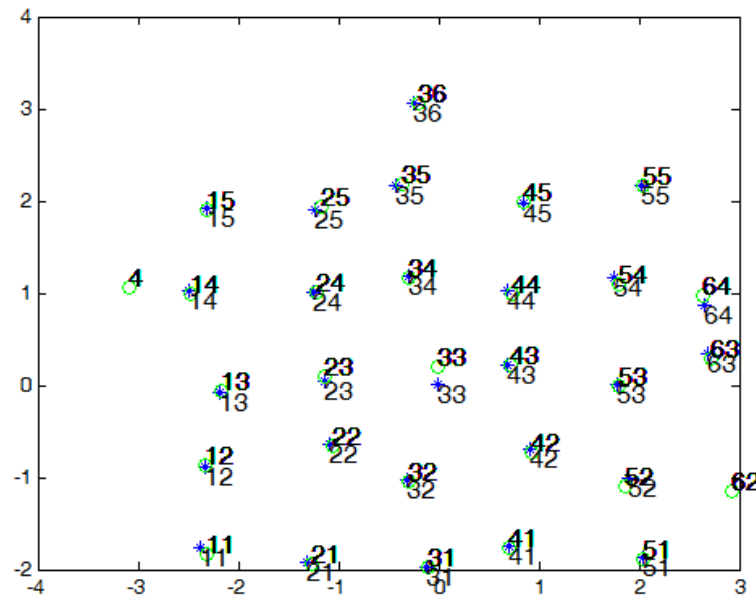


Figure 15: Local Map generated for base node ID 33

We now compare the results for various radio range and vision range. The average degree of connectivity in all four cases is greater than 5 and the noise in the placement error is 10% Gaussian noise. The table shows that with the increase of the radio communication range, the ID map results will be better and the registration error will be smaller. We can also observe that the radio range needs to be greater than $1.5R$ for correct ID association. The number of neighbors covered for a typical 2 hop neighborhood with $1.5R$ is 25% of the total number of nodes deployed in a network. In Table 2, we have listed the true mean square error (MSE) of the locations in the vision map and in the ID map for base node ID 55. This table summarizes the results for various radio ranges.

TABLE 2 : TRUE MEAN SQUARE ERROR OF THE LOCATIONS IN THE VISION MAP AND THE ID MAP $R_V=3.1$

Noise	10%	10%	10%	10%	10%
Radio Range	2R	1.75R	1.5R	1R	0.75R
True MSE	0.0176	0.0988	0.1057	3.885	4.605
ID association	correct	correct	correct	incorrect	incorrect

In the Table 3 we discuss the true mean square obtained for a vision map range of 4.1 for decreasing range of radio error starting from 2R. We can observe that the error for the radio range of less than $1.5R$ is larger. We now discuss the results for incorrect ID association due to symmetry and the correction applied. Figure shows the incorrect ID association resulting due to

symmetry. 10% noise in the placement of nodes results in 30% asymmetry, due to the 3σ noise in the deployment of the nodes.

TABLE 3: TRUE MEAN SQUARE ERROR OF THE LOCATIONS IN THE VISION MAP AND THE ID MAP $R_V=4.1$

Noise	10%	10%	10%	10%	10%
Radio Range	2R	1.75R	1.5R	1R	0.75R
True MSE	0.0293	0.1288	0.157	4.47	5.18
ID association	correct	correct	correct	incorrect	incorrect

4.2.2 Symmetry Correction Algorithm

In this section we examine the map validation algorithm. The ambiguity in local map generated basically results due to the irregularities in the MDS algorithm hence there is a need to validate the local maps further.

We have performed simulations for different vision ranges while keeping the radio range constant. The figure 16 and figure 17 illustrate the incorrect ID association scenarios for node ID 37. It can be observed that the True MSE value is higher than the threshold value of 1r units and hence the local map obtained is incorrect. After improving the algorithm with the investigation for reflection factor, the magnitude of the MSE value is found to be less than the threshold of 1r units. The figure 18 clearly shows the improved local map for node ID 37.

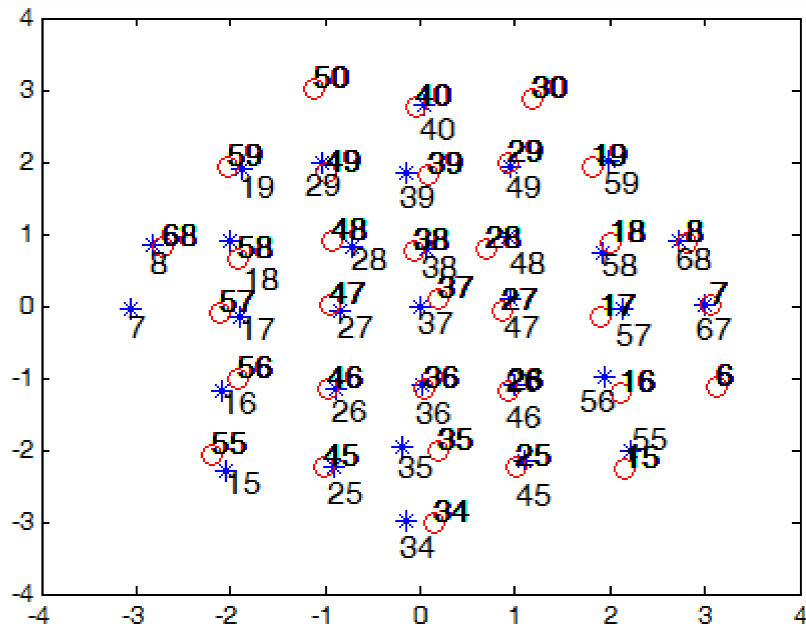


Figure 16: Incorrect ID association. The MSE value is 3.83

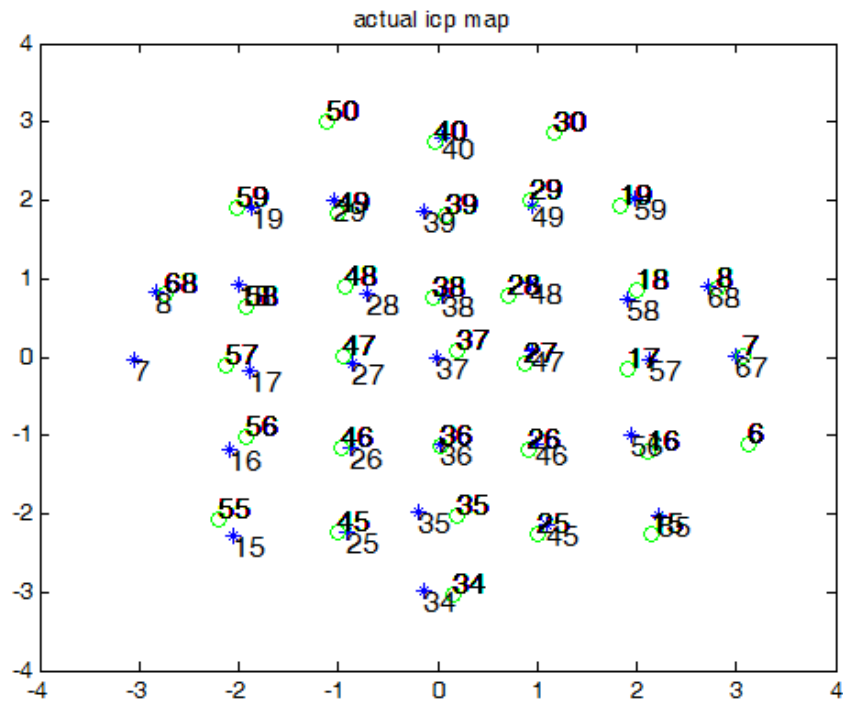
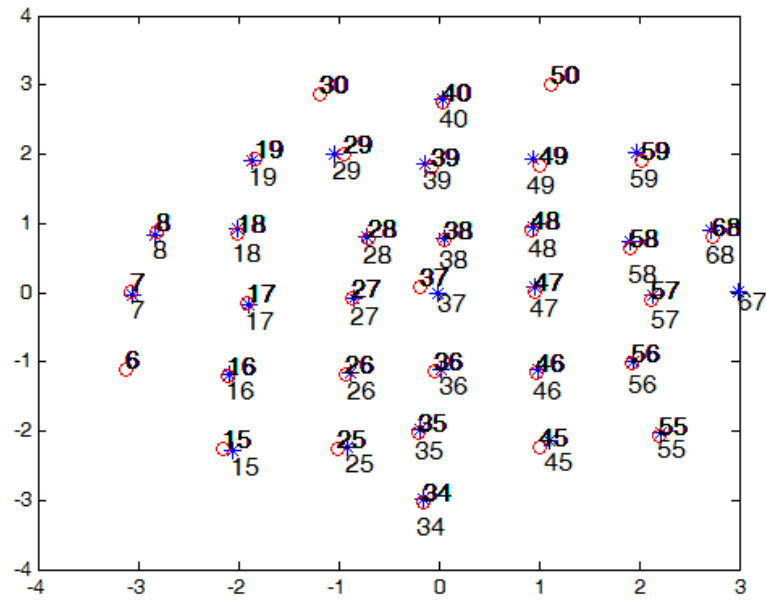


Figure 17: Incorrect ID association. The MSE value is 2.79



**Figure 18: Correct ID association after applying reflection and comparing the MSE values.
The MSE value obtained is 0.07**

In Table 4 we have tabulated the true MSE metric for incorrect ID association due to symmetry. The true MSE is a measure of how accurately the algorithm can correct the symmetry error. The radio range is kept constant at $1.95R$. The vision range is varied from $2.1r$ units to $3.1r$ units. For the vision range below $1.5r$ units the density of the nodes is very less and hence is not appropriate for simulation.

TABLE 4: CORRECTION IN SYMMETRY ERROR

Radio range	1.95R	1.95R	1.95R
Noise (Gaussian)	10%	10%	10%
Range error (MDS)	2%	2%	2%
Vision Radius	4.1	3.1	2.1
True MSE symmetry	2.7961	2.4263	1.398
ID Association	Incorrect	Incorrect	Incorrect
True MSE	0.0748	0.0648	0.348
ID Association	correct	correct	correct

4.2.3 Map Merging

In the map merging algorithm we examine the simulation results for merging of base node 55. Then we illustrate similar results for base node 66 and base node 33. We then tabulate our results by observing the accuracy metric at each stage of merging for node 55 and node 66. We further simulate the map merging for different noisy environments keeping the other parameters constant for base node 55 and tabulate the true MSE metric for the global map obtained in each case.

Figure 19 and Figure 20 illustrate the intermediate nodes selected for merging for base node 55. The nodes 37, node 33 and node 77 are the intermediate nodes selected. Figure 21 illustrates the global map obtained after performing successful map merging algorithm. The vision range for scenarios is kept at 3.1r units and the radio range id 1.95R units.

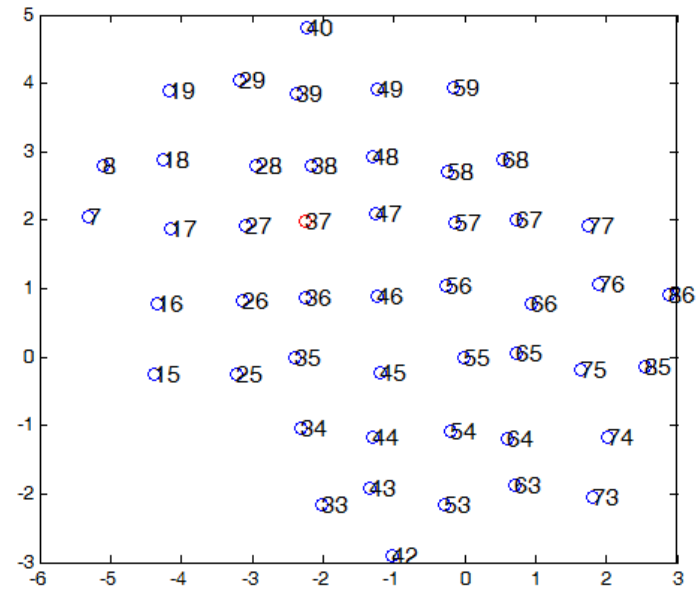


Figure 19: Merging of local map of node 55 with local map of node 37

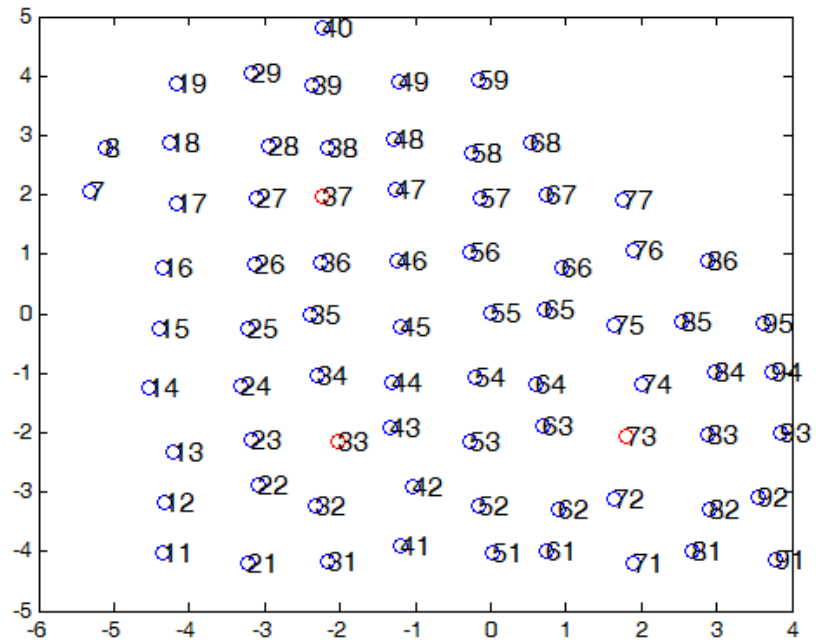


Figure 20: Intermediate global maps for nodes 37, 33, 77 merged with local map of the base node 55

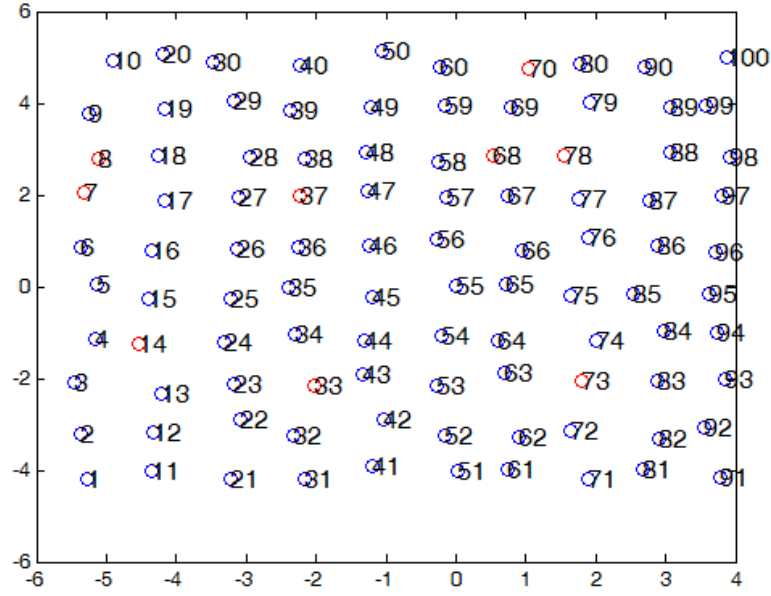


Figure 21: The global map for base node 55

Figure 21 shows the global map for the randomly selected node 55 chosen as the base node in this case.

We discuss another example wherein we started with the least ID as a base node and finally selected node 66 as the node to start with. As seen in Figure 22 node 47 is selected as the best node for merging with the base node 66 with 22 unknown IDs for a vision range of 3.1 and radio range of 1.95R. The Figure 23 illustrates the intermediate step where nodes 47, 25, 63 have been selected with unknown IDs 22, 18, 12. In the Figure 24 shows the global map obtained for node 66 for the vision range of 3.1.

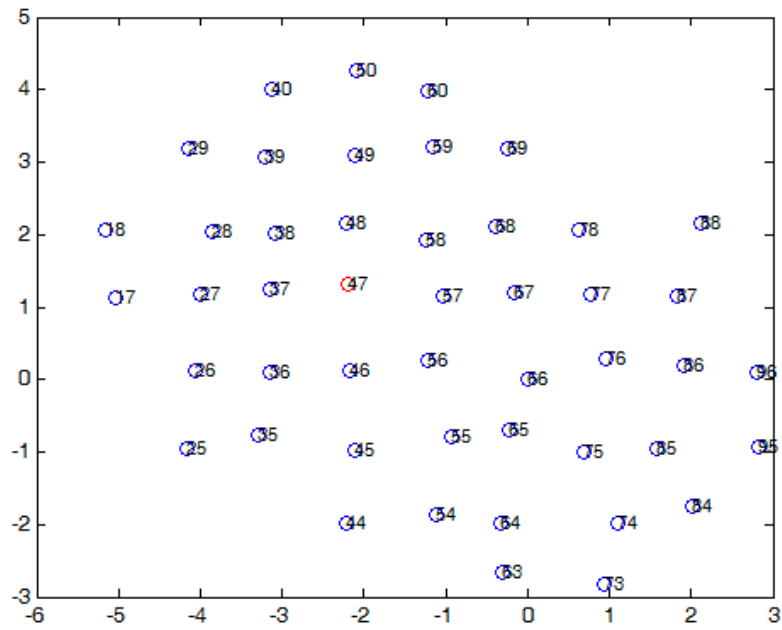


Figure 22: Base node 66 merged with node 47

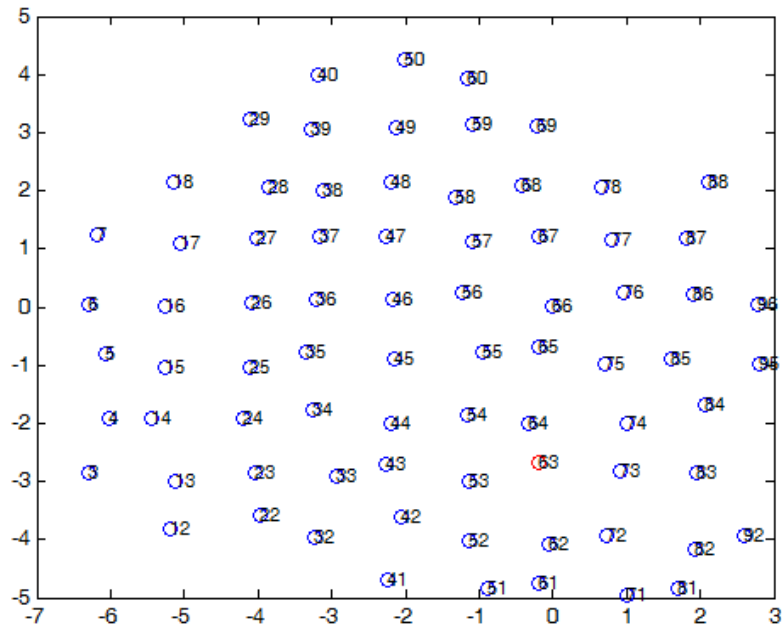


Figure 23: Intermediate global maps for nodes 47, 25, 63 merged with local map of the base node 66

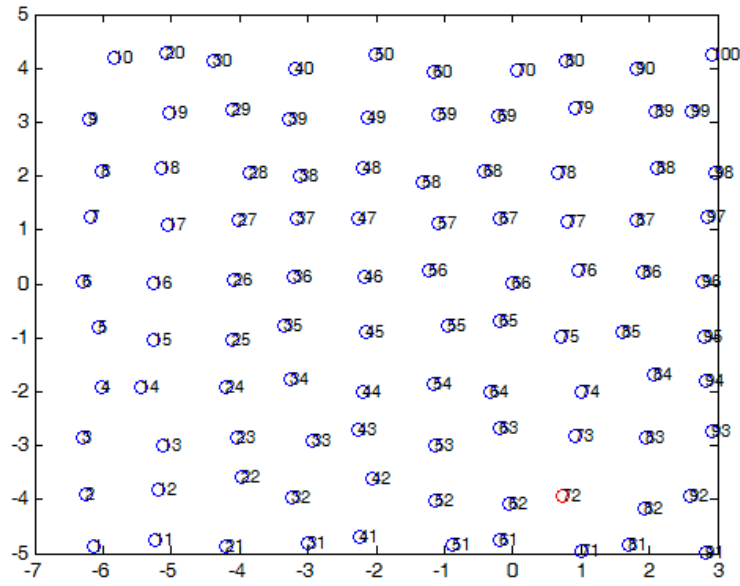


Figure 24: Global map for base node 66

The Figure 25 illustrates another example of base node local map for selected node ID 33 wherein the nodes selected for merging are IDs 55 58 73 28 78 12 3 70 with a vision radius of 3.1 and radio range of 1.99R and noise 10% ($3\sigma = 30\%$).

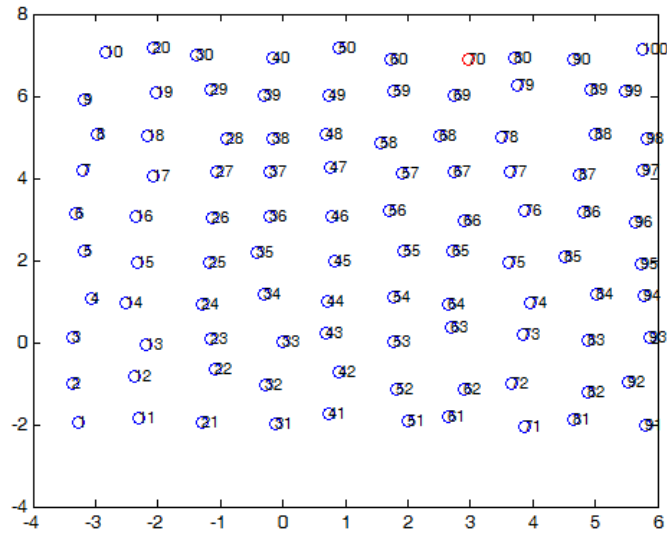


Figure 25: Global Map for base node 33

Table 5 shows the MSE for variations introduced in the placement errors. The mean square error for the placement error increases as more noise is introduced in the placement of the nodes. With increase in the gaussian noise above 20% the MDS results are observed to be incorrect which results in an incorrect global map. From our simulations, we also find that the ID association results are not good for sparse networks. When the average degree of connectivity is below 5 the ID map results will be bad, which will cause wrong pre-registration and wrong ID association. This will then lead to an incorrect map merging, and the global map obtained would be erroneous.

TABLE 5: % OF GOOD NODES IN THE LOCAL MAP OF THE BASE NODE 55

Next Node Selected For Merging	Number Of Good Nodes	Number Of Good Nodes (symmetry correction)	Total Nodes	% (good nodes with symmetry correction / total nodes)
37	19	36	48	75
58	31	42	61	68.85
28	38	57	74	77.07
77	38	60	86	69.76
2	40	65	94	69.14
69	48	74	100	74

The Table 6 shows the number of good nodes selected at each step of merging with base node 66. The table illustrates the improvement in the accuracy due to symmetry detection and

correction. The number of good nodes can be seen to be around 80% of the total number of nodes. This implies that there are 80% of nodes with correct local maps.

TABLE 6: % OF GOOD NODES FOR BASE NODE 66

Next Node Selected For Merging	Number Of Good Nodes	Number Of Good Nodes (symmetry correction)	Total Nodes	% (good nodes with symmetry correction / total nodes)
47	15	38	48	80
25	28	52	64	81.25
63	32	55	77	71.42
88	41	58	85	68.23
18	46	67	91	73.62
12	48	74	96	74
72	49	79	99	79.79

Table 7 shows the MSE metric for variations introduced in the placement errors. The mean square error for the placement error increases as more noise is introduced in the placement of the nodes. With increase in the Gaussian noise above 20% the MDS results are observed to be incorrect which results in an incorrect global map.

**TABLE 7: MEAN SQUARE ERROR FOR VARIOUS PLACEMENT ERRORS (NOISE)
FOR A SELECTED VISION RANGE OF 3.1**

Gaussian Noise	MEAN SQUARE ERROR	GLOBAL MAP OBTAINED
(%)	BETWEEN THE COMMON ID	
5	0.03	Correct
10	0.031	Correct
15	0.033	Correct
20	0.046	Correct
25	2.38	Incorrect

CHAPTER 5

CONCLUSION AND FUTURE WORK

The thesis work provides a solution to the network camera localization calibration problem in a distributed fashion. In chapter 1 we introduced the emerging field of wireless multimedia sensor networks and we discuss the related work starting from GPS based localization to range based localization techniques. Then we explored the single camera calibration techniques and the problems associated with the same. This led us to shift our focus to network camera calibration and we investigate the techniques for the same. Chapter 2 focused on developing the problem statement and the overall proposed solution which is a two step process involving local map generation and map merging algorithm. The detailed theoretical framework of local maps was put forth in chapter 3. As explained in the third chapter, in the local map generation algorithm we explained the ID map generation and ID association algorithm taking into account the refinement due to ICP algorithm. We then discussed the map validation algorithm to check the uncertainties due to classical MDS. Here the node checks whether its local map is correct by sharing its local map with the neighboring nodes and comparing the mean square error with a predefined threshold value and classifies itself as a good node. In the second part, we explained the map merging algorithm. The map merging algorithm consists of finding the next best node, and then merging the partial global map and the selected local map from the set of good nodes by finding the best possible linear transformation between the coordinates of these maps. The simulation results stand in accordance to all of the specified conclusions over a variety of radio and vision range.

The major contribution of our thesis lies in developing the theoretical framework for distributed camera calibration based on vision data, local inter-node distances and local topology. Moreover

our work does not need any moving targets to perform distributed camera calibration. We further develop innovative algorithms for local map generation and map merging. We evaluated the proposed approach through computer simulations. This location calibration algorithm can thus be used to develop self-organized wireless multimedia sensor networks.

Our thesis work caters to wide variety of applications which include surveillance systems during crime and terrorist attacks. They are also useful for traffic monitoring in cities and highways. Military applications include locating the targets of interest (such as enemy soldiers, tanks) in the battlefield and relay video images to the command center.

The errors due the network topology and grid based deployment of the sensor nodes are not completely eliminated and hence not all nodes have a good local map. Future work could involve investigating the key factors contributing to the same. Also the future work would include finding the heading of the wireless camera sensor nodes without using the compass. The method described in the thesis is performed only in simulation environment, and the performance of the algorithm needs to be evaluated in real time camera sensor networks.

REFERENCES

- [1] A. Rowe, C. Rosenberg, I. Nourbakhsh, "A low cost embedded color vision system." IEEE/RSJ Intl. Conf Intelligent Robots and Systems (IROS), Lausanne, Switzerland, October 2002.
- [2] A. Savvides, C. C. Han, and M. Srivastava. "Dynamic fine-grained localization in ad-hoc networks of sensors." 2001.
- [3] A. Savvides, W. L. Garber, R. L. Moses, and M. B. Srivastava. "An analysis of error inducing parameters in multihop sensor node localization." IEEE Transactions on Mobile Computing, 4(6). November/December 2005.
- [4] Aghajan, D. Stephan Hengstler and H.K. "Wisnap: A wireless image sensor network application platform." 2nd International Conference on Testbeds & Research Infrastructures for the DEvelopment of NeTworks & COMmunities, Barcelona, Spain 2006.
- [5] Aghajan, H. Lee and H. "Collaborative node localization in surveillance networks using opportunistic target observations in VSSN '06: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks." Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks, pp.9-18, 2006.
- [6] Atallah, M. J. "Algorithms and Theory of Computation Handbook." CRC Press, 1998.
- [7] Baker, S. K. Nayar and S. "Catadioptric image formation." DARPA Image Understanding Workshop, pp. 1431-1437, 1997
- [8] Beutel, J. "Geolocation in a PicoRadio Environment." Master Thesis. UC Berkeley, 1999
- [9] Borg, Ingwer. Modern multidimensional scaling : theory and applications. Springer, 2006
- [10] C. Savarese, J.M. Rabaey, and J. Beutel. "Locationing in distributed ad-hoc wireless sensor networks." . In Proc. 2001 Int'l Conf. Acoustic Speech, and Signal Processing (ICASSP 2001), volume 4. pp. 2037–2040, May 2001.
- [11] "Locationing in distributed ad hoc wireless sensor networks." Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP 2001), volume 4, 2037–2040. May 2001.
- [12] Cox, Trevor F., "Multidimensional scaling ." Chapman & Hall. , Chapters 1, 2 , 3 , 4, 2001

- [13] D. Lymberopoulos, A. Barton-Sweeney, and A. Savvides,. "Sensor Localization and Camera Calibration using Low Power Cameras." Technical Report 08050. 2005.
- [14] G. Sharp, S. Lee and D. Wehe. " ICP registration using invariant features ." PAMI , 24,1,2002.
- [15] G. Werner-Allen, P. Swieskowski, and M. Welsh,. "MoteLab: A wireless sensor network testbed ." 4th International Conference onInformation Processing in Sensor Networks (IPSN '05) . pp. 483–488, April 2005.
- [16] G.Fryer, T. A. Clarke and J. "The development of camera calibration methods and models." The photogrammetric Record, vol. 16, no. 91,pp. 51–6, 1998
- [17] Georgiy Pekhteryev, Zafer Sahinoglu, Philip Orlik, and Ghulam Bhatti. " “Image transmission over IEEE 802.15.4 and zigbee networks”." IEEE International Symposium on Circuits and Systems (ISCAS 2005), volume 4, 3539–3542, May 2005.
- [18] Guibas, F. Zhao and L. " Wireless Sensor Networks: An Information Processing Approach." Elsevier and Morgan Kaufmann Publishers, 2004
- [19] "Wireless Sensor Networks: An Information Processing Approach." Elsevier and Morgan Kaufmann Publishers. 2004
- [20] Huang Lee, Hamid Aghajan. ""Collaborative Node Localization using opportunistic Target Observations”." Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks. . 9-10, October 2006
- [21] Ian F. Akyildiz, Fellow IEEE, Tommaso Melodia. "Wireless Multimedia Sensor Networks: Applications and Testbeds." Proceedings of IEEE, Volume 96, Issue 10, . Oct. 1588 – 1605, 2008 .
- [22] ISHIGURO, Hiroshi. "Development of Low-Cost Compact Omnidirectional Vision Sensors." Panoramic Vision, chapter 3, pp. 433-439, 1998.
- [23] J. Feng, S. Megerian, and M. Potkonjak. " “Model-based calibration for sensor networks”." Sensors. 737 – 742, October 2003.
- [24] J. Hong, and others. " Image-based homing." Proc. Int. Conf.Robotics and Automation. 1991.
- [25] J.A. Costa, N. Patwari, and A. Hero. "Distributed weighted multidimensional scaling for node localization in sensor networks." ACM Transactions on Sensor Networks. February 2006.

- [26] K. Yamazawa, Y. Yagi and M. Yachida. " Omnidirectional imaging with hyperboloidal projection." Proc. Int. Conf. Robots and Systems. 1993.
- [27] Kawato, Y. Yagi and S. "Panoramic scene analysis with conicprojection." Proc. Int. Conf. Robots and Systems. 1990.
- [28] M. Paskin S. Funiak, C. Guestrin and R. Sukthankar. " "Distributed localization of networked cameras"." ISPN. 2006.
- [29] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. "'The cricket location-support system", ." MobiCom. 2000.
- [30] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low cost outdoor localization for very small devices. University of Southern California, Los Angeles, CA: Computer science department, 2000.
- [31] GPS-less low cost outdoor localization for very small devices. Technical Report 00-729. University of Southern California,. Los Angeles, CA, 2000.
- [32] N. Patwari, J. N. Ash, A. O. Hero, R. Moses, and N. S. Correal. " Locating the nodes." IEEE Signal Processing Magazine, 54, July 2005.
- [33] P. Kulkarni, P. J. Shenoy, and D. Ganesan. " Approximate initialization of camera sensor networks" EWSN, pp. 67–82, 2007.
- [34] Priyantha, N. B, A Chakraborty and H and Balakrishnan. "The cricket location-support system." Proceedings of the Annual International Conference on Mobile Computing and Networking, pp. 67–82, MOBICOM. 2000.
- [35] R. L. Moses, D. Krishnamurthy, and R. Patterson. " A self-localization method for wireless sensor networks. ." Paper on Applied Signal Processing. March 2003.
- [36] Canon's Omnidirectional Camera Is a 50 Megapixel Eye That Looks Everywhere at Once, Gizmodo at www.gizmodo.com
- [37] Rees, D. W. Panoramic television viewing system. United States: Patent 3, 505,465. Apr. 1970.
- [38] Savvides, D. Lymberopoulos and A. " "XYZ: A Motion Enabled Power Aware Sensor Node Platform for Distributed Sensor Network Applications"." Information Processing In Sensor Networks,. 795-825, April 2005..
- [39] Shang, Y., W. Ruml and Y.,Fromherz, M. P. J. Zhang. " Localization from mere connectivity." Fourth International ACM Symposium on Mobile Ad Hoc Networking and Computing. Annapolis; MD. NY, 201-212, June 1-3,2003.

- [40] Sogo, Takushi. "Localization of Sensors and Objects in Distributed Omnidirectional Vision." PhD Thesis. 2001.
- [41] Tsai, R. "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off the-shelf tv cameras and lenses." IEEE J. Robotics Automat., vol. 3, no. 4. 323–344, 1987.
- [42] V Mehta, W Sheng, T Chen and Q Shi. " Development and Calibration of a Low Cost Wireless Camera Sensor Network." IROS. St. Louis, Missouri, USA, 2009.
- [43] W.Ruml, Y. Shang and. "Improved MDS based Localization." Proceedings of IEEE InfoCom'04, 2004.
- [44] X. Liu, P. Kulkarni, P. J. Shenoy, and D. Ganesan. "Snapshot: A self-calibration protocol for camera sensor Networks." BROADNETS. 2006.
- [45] X. Nguyen, M.I. Jordon and B. Sinopli. "A kernel-based learning approach to ad hoc sensor network localization." ACM Transactions on Sensor Networks, 134-152, 2005.
- [46] Zhang, Z. " A flexible new technique for camera calibration." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11. 1330–1334, 2000.

VITA

Rohit Shrikant Kadam

Candidate for the Degree of

Master of Science

Thesis: DISTRIBUTED LOCATION CALIBRATION IN WIRELESS MULTIMEDIA
SENSOR NETWORKS

Major Field: Electrical Engineering

Biographical:

Personal Data:

Born on October 16th 1985 in Satara, the city of seven hills in Maharashtra, India

Education:

Pursued Bachelor of Engineering in Instrumentation and Control in University of Pune in June 2007

Completed the requirements for the Master of Science in Electrical and Computer Engineering at Oklahoma State University, Stillwater, Oklahoma in December, 2010

Experience:

Worked as a Teaching Assistant at Oklahoma State University for an undergraduate level course (ECEN 4213) for a period of three semesters. Also worked as a Research Assistant in Laboratory for Advanced Sensing, Computation and Control.

Name: Rohit Shrikant Kadam

Date of Degree: December 2010

Institution: Oklahoma State University

Location: Stillwater, Oklahoma

Title of Study: Distributed Location Calibration in Wireless Multimedia Sensor Networks

Pages in Study: 59

Candidate for the Degree of Master of Science

Major Field: Electrical Engineering

Scope and Method of Study: Wireless Multimedia Sensor Networks (WMSNs) are gaining popularity among researchers over the past few years. Knowledge of the geographic locations of the sensor nodes is very important in such sensor networks. Location calibration is a method that uses the connectivity information, the estimated distance information among the sensor nodes, as well as the vision images to find the location of the sensor nodes in WMSNs. We generate local maps for nodes in immediate vicinity and merge them together to get a global map. Local maps are prone to be incorrect mainly due to distribution symmetry in a grid based network and uncertainties in the classical multidimensional scaling. Performance measures to calculate and study the same have been developed and described in detail

Findings and Conclusions: In this research we propose a new algorithm which corrects the error to improve the location calibration in WMSN. The major contribution of our thesis lies in developing the theoretical framework for distributed camera calibration based on vision data, local inter-node distances and local topology. Moreover our work does not need any moving targets to perform distributed camera calibration. We further develop innovative algorithms for local map generation and map merging. We evaluated the proposed approach through computer simulations. This location calibration algorithm can thus be used to develop self-organized wireless multimedia sensor networks. We demonstrate the effectiveness of our proposed approach through computer simulation. We demonstrate the proposed approach for different base node and successfully build the global map. The percentage of accuracy obtained demonstrate that the 80% of the nodes can have a good local map, however the global map obtained can accurately localize all the nodes in the network. This brings a completion to the scope of this thesis, the framework developed further provides an opportunity to extend this algorithm to real time wireless multimedia sensor networks

Publication:

"Multidimensional Scaling Based Location Calibration for Wireless Multimedia Sensor Networks", Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan 2010

ADVISOR'S APPROVAL: Dr Weihua Sheng