

Anchor Node Placement for Localization in Wireless Sensor Networks

by

Benjamin Tatham

A Dissertation submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Applied Science

Ottawa-Carleton Institute for
Electrical and Computer Engineering

Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada

December 2010

Copyright ©
2010 - Benjamin Tatham

The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the Dissertation

Anchor Node Placement for Localization in Wireless Sensor Networks

Submitted by **Benjamin Tatham**
in partial fulfilment of the requirements for the degree of
Master of Applied Science

Dr. Thomas Kunz, Supervisor

Dr. H.M. Schwartz, Department Chair

Carleton University

2010

Abstract

Applications of wireless sensor network (WSN) often expect knowledge of the precise location of the nodes. One class of localization protocols patches together relative-coordinate, local maps into a global-coordinate map. These protocols require some nodes that know their absolute coordinates, called anchor nodes. While many factors influence the node position errors, in this class of protocols, the placement of the anchor nodes can significantly impacts the error. Through simulation, using the Curvilinear Component Analysis (CCA-MAP) protocol, we show the impact of anchor node placement and a set of rules to ensure the best possible outcome, while using the smallest number of anchor nodes possible. Scientists and researchers are thus enabled to focus on the sensed data with confidence in the node localization results.

Dedicated to my wife and children who supported me through the long process of
this research.

Acknowledgments

I would like to thank Professor Thomas Kunz for his guidance and encouragement that made this research possible. Further, I would like to thank Li Li whom, along with Professor Kunz, performed the initial CCA research and wrote the basis for the Matlab simulations used in this thesis. Finally, my colleagues Shafagh Alikhani and Ben Gardiner who provided a tireless ear to listen to ideas throughout the process.

Table of Contents

Abstract	iii
Acknowledgments	v
Table of Contents	vi
List of Figures	vii
1 Introduction	1
1.1 Motivation	2
1.2 Thesis Contribution	5
1.3 Methodology	6
1.4 Thesis Organization	6
2 Overview of Wireless Sensor Networks and Localization	7
2.1 Wireless Sensor Networks	7
2.2 Localization Protocols	8
2.2.1 Ad Hoc Positioning System	9
2.2.2 MDS-MAP	10
2.2.3 CCA-MAP	10
3 Related Work on Anchor Node Placement	12

3.1	Empirical Evidence	12
3.2	Explicit Studies of Anchor Node Placement	13
3.3	Summary of Related Work	14
4	Optimal Anchor Node Placements	15
4.1	Measuring Location Error	15
4.2	Coordinate Transformation	16
4.3	Methodology	18
4.4	Anchor Node Placement Metrics	19
4.4.1	Anchor Node Error	20
4.4.2	Network Area Coverage	23
4.4.3	Anchor Node Triangle	25
4.5	Best Anchor Node Placement	29
5	Avoiding the Worst Anchor Node Placements	33
5.1	Effects of Procrustes Analysis	33
5.1.1	Transformation Reflection and Rotation	35
5.1.2	Transformation Scaling and Translation	37
5.1.3	Procrustes Dissimilarity	37
5.2	Outlier Indicators	41
6	Effects of Network Topology	45
6.1	Node Distance from the Anchors	45
6.2	Applicability to Varying Network Topologies	47
6.2.1	C-Shape Network Topology	47
6.2.2	Pipeline Network Topology	49
7	Other Factors and Future Work	55

8 Conclusion	59
8.1 Future Work	59
8.2 Summary	60
List of References	61
Appendix A Matlab©Simulation Code	64

List of Figures

1.1	Reasonable localization results	3
1.2	Poor localization results	4
4.1	A sample anchor triangle Procrustes analysis	17
4.2	The sample network used in this chapter	19
4.3	Number of anchor neighbors vs. location error	21
4.4	Mean of anchor node error vs. location error	22
4.5	Sum of Distances = $\sum_{i=0}^n \ x_i\ $	24
4.6	Sum of distance between anchors vs. location error	26
4.7	Minimum distance between anchors vs. location error	27
4.8	Area and height of a triangle	28
4.9	Area of anchor triangle vs. location error	30
4.10	Minimum height of anchor triangle vs. location error	31
4.11	Sum of distance between anchors vs. location error	32
5.1	Network map of localization performance at each node	34
5.2	Rotation and reflection versus location error	36
5.3	Transformation scalar component versus location error	38
5.4	Transformation translation component versus location error	39
5.5	Procrustes dissimilarity measure	40
5.6	Anchor triangle areas	42
5.7	Minimum anchor triangle heights	43

6.1	Localization error as a contour plot	46
6.2	Example of C-shape network topology	48
6.3	Anchor metrics for a C-Shape topology	50
6.4	Example of pipeline network topology	51
6.5	Sum of distances between anchor nodes vs. location error in a pipeline topology	52
6.6	Sum of distances between anchor nodes vs. location error in a pipeline topology	54
7.1	Good localization performance with same anchor set in two different networks	56
7.2	Poor localization performance with same anchor set in two different networks	57

Chapter 1

Introduction

Scientists, engineers, and researchers use wireless sensor networks (WSN) for a wide array of applications. Many of these applications rely on knowledge of the precise position of each node. While some may only require relative coordinates within the network, most biological, geophysical, and other scientific applications require coordinates on a global coordinate system. Perhaps the obvious solution is for each node in the network to be equipped with GPS or other location positioning services. However, constraints on cost, power consumption, as well as visibility of satellites dictate the need for an alternative solution.

Many protocols have been proposed [1–3] to calculate relative positions amongst the nodes of a network. They vary in the required network functionality in terms of radio ranging or range-free. Radio ranging involves specialized hardware to measure the distance between nodes based on physical data like signal strength or transmission delays. In order to convert from relative to global coordinates, some of the nodes require a local source of global coordinates. This can be achieved by operators recording the global coordinates during network deployment, by embedding a GPS receiver in a subset of the nodes, or some other source. We call these enhanced nodes anchors. Here, we explore the effect of anchor node placement within the network

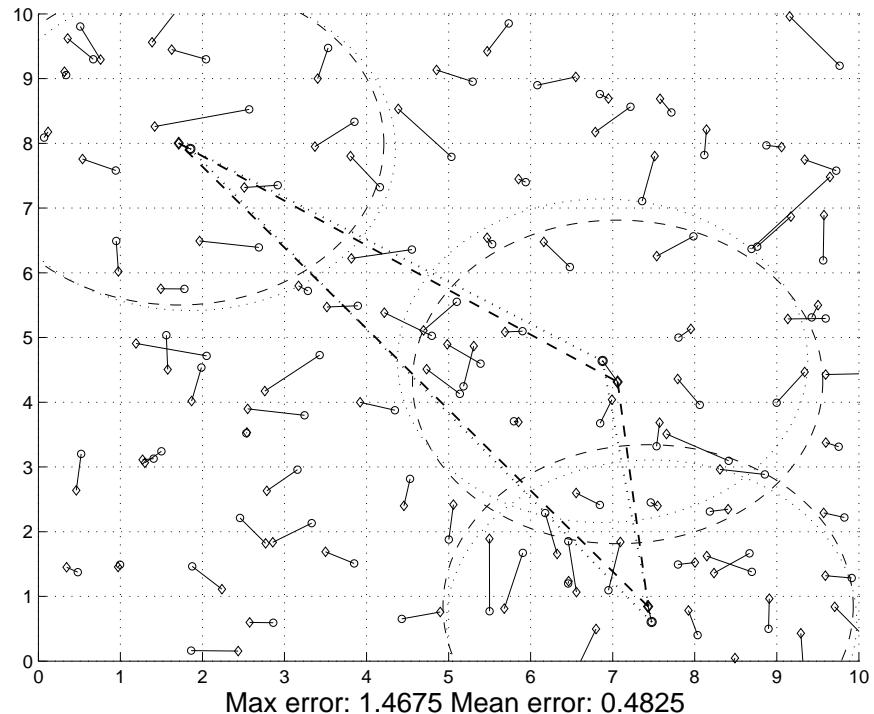
on the overall localization errors, on a network-wide basis. This provides network planners with a set of general rules to minimize the number of anchor nodes required while avoiding poor node localization, allowing scientists to assume a maximum position error during their own research. Further, based on application requirements of location accuracy, planners can minimize the cost of the network associated with anchor nodes by using the minimum number and best position.

1.1 Motivation

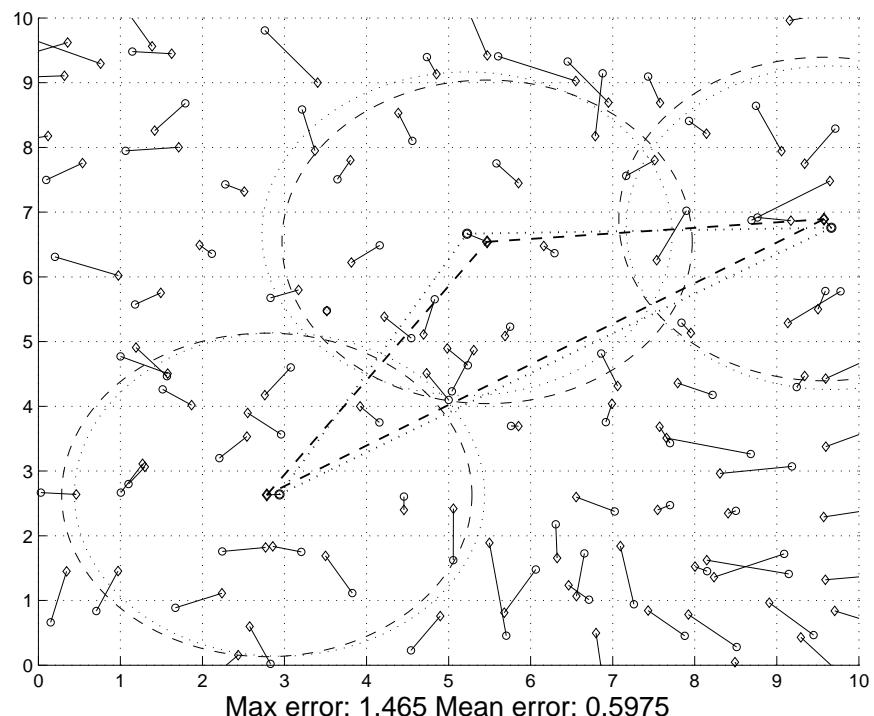
During previous work designing localization protocols [4, p. 11] [2, p.2], researchers often choose anchors at random within the network. Frequently, they simulate the network multiple times with different anchors in order to statistically exclude anchor node placement from their results.

Our initial investigations and simulations demonstrate that indeed the placement of anchor nodes in the network does have an often dramatic effect on the location error. The four plots shown in Figures 1.1 and 1.2 graphically establish that anchor node position does make a difference. Each plot shows the same network with a different choice of three anchors. A line is drawn between the actual and calculated position of each node to visualize the localization error. The circles show the radio range of each anchor, and a triangle is drawn between the three anchors for clarity.

The four plots shown are taken from a set of 100 randomly chosen anchor sets. While the first two choices, shown in Figure 1.1, have reasonable errors, the third choice, in Figure 1.2a, has an error more than twice the mean error of the first. Further, the fourth choice, in Figure 1.2b, has an extremely poor performance, more than ten times that of the first choice. The four anchor set triangles shown do not immediately show an obvious progression that could explain this dramatic change in

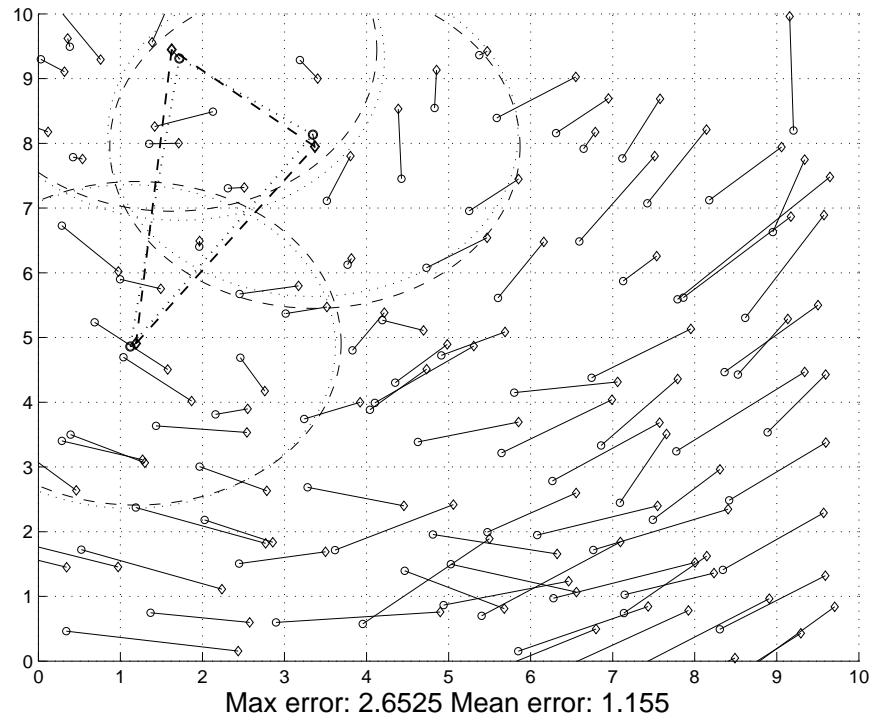


(a) Anchor Set 1

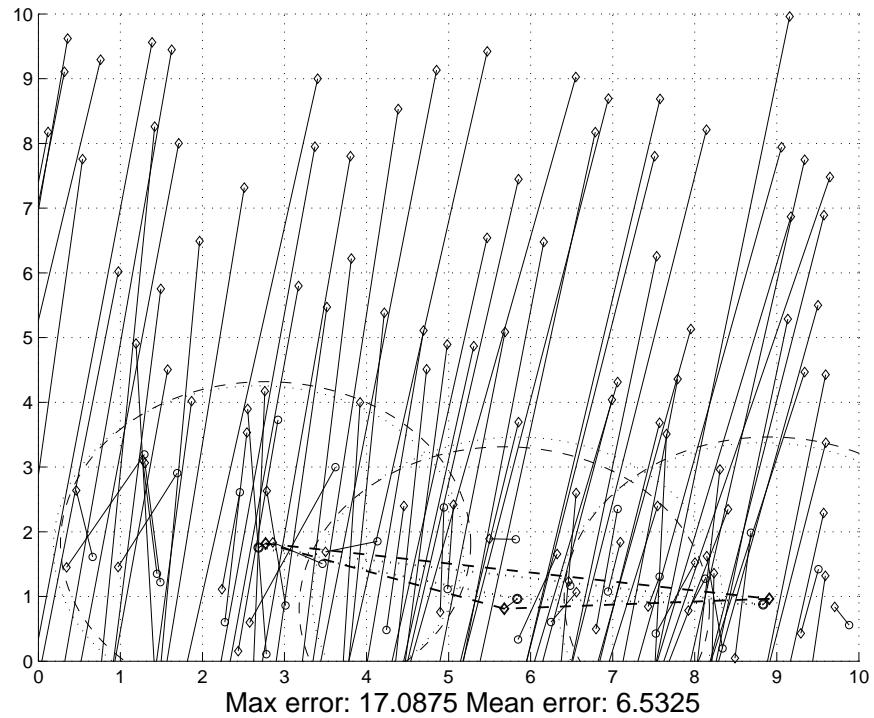


(b) Anchor Set 2

Figure 1.1: Reasonable localization results



(a) Anchor Set 3



(b) Anchor Set 4

Figure 1.2: Poor localization results

error. We find it interesting that there is an incremental increase in error between these four anchor sets resulting in a full order of magnitude difference between good and bad cases, and thus requires the further investigation presented in this thesis.

In practical applications, scientists and engineers need to know what degree of accuracy the localization algorithm provides. This study, therefore, diagnoses the impact of anchor node placement on these errors and, while providing guidelines for anchor node placement, establishes some bounds for the resulting error.

1.2 Thesis Contribution

We establish that by avoiding certain poor anchor node placements, extremely poor location errors can be avoided. Further, we demonstrate that other than these edge cases, the location error falls within a statistically insignificant range, where the range is a function of network connectivity and topology. Specifically, we show that the probability of extremely high location error results from anchor nodes being roughly in a geographically straight line. As the anchor nodes are spread out from a straight line, the probability of high errors decreases, leaving network designers a relatively simple chore when choosing anchor nodes locations.

While some papers have touched upon anchor node placement, we have yet to come across a comprehensive study of the optimal anchor node placement. This paper provides a comprehensive study of possible anchor node placements and their effects on overall network localization accuracy.

While this paper is a comprehensive study of anchor node placements, we focus on a specific set of network design assumptions. First, we assume the network planners want to minimize the number of anchors since they consume more power and cost

more. Therefore, we assume the minimum number of anchors of three to provide two-dimensional coordinates. Further, we assume that range-free algorithms are preferred to avoid dealing with multi-path and other radio effects as well as to limit the required hardware and power requirements on each node in the network.

1.3 Methodology

Many localization protocols and algorithms provide a set of relative coordinates that are then transformed into global coordinates. For the purpose of this research, we chose CCA-MAP [3, 4] as the algorithm to provide simulation results. A Matlab[®] simulation of this algorithm already existed [3], and was modified to provide the necessary output statistics presented here. CCA-MAP is described in more detail in Section 2.2.3.

1.4 Thesis Organization

A brief background of Wireless Sensor Networks and localization protocols in general are presented in Chapter 2. Chapter 3 presents the limited related work on anchor node placement. Chapter 4 contains various proposed anchor node placement methods and summaries of how they perform. The cause of the extreme edge cases comes to light in Chapter 5. Chapter 6 discusses various other factors. Chapter 7 gives some ideas for future work, including other factors that effect localization error and Chapter 8 summarizes the results of this study.

Chapter 2

Overview of Wireless Sensor Networks and Localization

2.1 Wireless Sensor Networks

A wireless sensor network (WSN) consists of a set of nodes tasked with sensing environmental phenomenon at or near each node. Nodes communicate via radios to send their data back to a central acquisition system. Nodes are typically small, cheap devices and are designed with power efficiency in mind to prolong the lifetime of the network's ability to collect data. Nodes are often distributed in the field of interest randomly, sometimes even by dropping them from the air, as on a military battlefield. Other times, they are placed in specific, but unknown *a priori*, locations, as in placing them in bird nests [5]. Or, they may be rolled into a transportation tunnel to give firefighters and emergency crews current information about heat and oxygen levels [6].

A number of issues arise when designing a WSN. Each node must be able to communicate with other nodes and send data to a central collection site. Each node must know what time it is, for purposes of data sampling, and often for routing

protocols as well. Further, each node must know where it is so spatial data can be properly correlated. Location can also be useful for geographic routing protocols. This thesis focuses on determining the location of each node as accurately as possible.

2.2 Localization Protocols

There are two general classes of localization protocols: ranging and range-free. Ranging protocols rely on information from the radio. With this information, a fairly accurate network topology can be built. Ranging techniques can use a variety of metrics to build the network topology. These include Time-of-Arrival (TOA), like GPS [7], Time-Differential-of-Arrival (TDOA) [8], Angle-of-Arrival (AOA) [9], or Received-Signal-Strength-Indicator (RSSI) [10].

However, the special hardware and power requirement to perform these ranging techniques is counter to the goal of low-cost, low-power nodes, and thus we exclude ranging protocols from our study. Regardless, if a ranging protocol does build a relative map, and then does a post-processing step by mapping this relative map to a global map based on a subset of anchor nodes, the results of this thesis apply to ranging as well as range-free protocols.

Range-free protocols do not rely on any specialized hardware for additional information. Rather, they rely solely on network connectivity, specifically knowledge of their direct neighbors. Often, a node will collect information about their direct neighbors' neighbors as well, known as two-hop information. Knowledge of each further node requires more information to be shared and therefore transmitted between nodes, thus requiring more power for radio transmission. It is for this reason that only one-hop or possibly two-hop knowledge is preferred.

2.2.1 Ad Hoc Positioning System

Niculesu, et al. propose a distributed localization algorithm known as Ad Hoc Positioning System (APS) [1]. It is similar to GPS in that it uses triangulation to determine node positions. In APS, each node maintains a table of distances to each anchor. The distance can be represented as a hop count, estimated distance using RSSI, or Euclidean distance. As a distributed algorithm, each node determines its own position based on the distances to the anchor nodes. Thus, APS does not perform well in anisotropic network, that is networks with holes or "C" shapes in the topology, because the communication distance can be far greater than the geometric distance between two nodes.

In its simplest form, APS uses a propagation technique called DV-HOP to determine distances between nodes. DV-HOP is based on classical distance vector exchange from general network protocols like TCP/IP. Each node maintains a table of hop counts between all known nodes. Each node exchanges this table only with its direct neighbors. When an anchor has discovered a hop count to another anchor, the anchor estimates the average distance for each hop since it knows the absolute location of itself and the other anchor. This correction factor is sent to the entire network. DV-HOP thus minimizes the amount of data that must be transmitted in the network.

Further, APS can employ a propagation technique called DV-distance. DV-distance is similar to DV-hop except that it uses RSSI to determine each hop distance and sends this distance instead of hop count. This difference allows DV-distance to effectively detect holes and curves in the network as each anchor can see that the transmission path between them is larger than the Euclidean distance.

2.2.2 MDS-MAP

Shang, et al. attempt to correct the errors introduced by APS and other distributed algorithms through a centralized localization algorithm called MDS-MAP(C) [2], where the C is for centralized. MDS-MAP(C) is divided into three phases. In phase one, shortest path distances or hop counts are exchanged via a distance vector exchange, similar to APS. This provides a rough estimate of the distance between each pair of sensors. In phase two, multi-dimensional scaling (MDS) is applied, resulting in a relative map. MDS is a general data analysis tool originating from psychophysics to transform data from many to few dimensions. In simple terms, MDS takes a set of distances between points and creates a structure that fits those distances. Often, it is used for general data visualization. In this case, the relative map conforms closely to the pair-wise distances provided. In phase three of MDS-MAP(C), the relative map is transformed into a global coordinate system using at least three anchors using the Procrustes algorithm. Procrustes is described in more detail in Section 4.2.

The authors provided a modified, distributed version, MDS-MAP(P) [11]. This variation simply divides the network into smaller, more manageable sections so that the algorithm can be performed locally, with the limited node resources available. Each local map is then patched together, and hence the P for patched. The patching part of the algorithm is not distributed. Local map merging begins at a randomly selected node's local map, and chooses the local map with the most overlapping nodes. The process continues until all the local maps are merged together.

2.2.3 CCA-MAP

Li, et al, propose a similar style algorithm to MDS-MAP called CCA-MAP [3, 4]. It is similar in that it generates relative, local maps of sections of the network and then patches them together into a global coordinate system. CCA-MAP improves

on MDS-MAP in that the algorithm is more efficient. MDS is a non-linear reduction algorithm and has a computational cost of $O(n^3)$. CCA [12] on the other hand, is a self-organized neural network performing quantization and non-linear projection. CCA-MAP has a total computational cost of $O(n^2)$. CCA runs in a series of iterations, where each iteration has a computational cost of $O(n)$.

CCA-MAP has four phases. In the first phase, each node builds a local map of nodes within R hops. For that local map, the shortest distance matrix is accumulated, as in APS and MDS-MAP. The second phase involves performing the CCA algorithm itself on each local map, generating relative coordinates for each node in the local map. In phase three, the local maps are merged together, as in MDS-MAP(P), and finally, in phase four, the relative coordinates are transformed into absolute coordinates based on the known coordinates of the anchor nodes, as described in Section 4.2. Phase four can only be performed with a minimum of three anchors for $2D$ space or four anchors for $3D$ space.

CCA-MAP is flexible as to where computations can be performed. Local map calculations can be performed at the nodes themselves, if computing resources allow, or outsourced to more powerful gateway nodes or a central server. Local map merging can be performed in parallel at selected nodes in the network, or again at a central server. Further, if in any sub-map sufficient anchors are found, then absolute coordinates can be calculated.

Chapter 3

Related Work on Anchor Node Placement

While much attention has been paid to localization accuracy and computational effort, the importance of intelligent anchor node placement is often recognized, but dismissed as future study.

3.1 Empirical Evidence

Often, authors will come across anchor placement by accident and discuss it based on their own empirical evidence. Shang, et al. [2, p. 964] and Li, et al. [3, p. 11] both choose anchors at random within the network. Although, Shang does mention that a co-linear set of anchors chosen in one example "represents a rather unlucky selection", without supporting evidence of why this is unlucky.

Earlier work by Doherty, et al. [13] requires anchor nodes to be placed at the edges, and ideally at the corners of the network. In this case, however, the algorithm is a simple constraint problem. One constraint requires that all the unknown nodes be placed within the convex hull of the anchors, and therefore, better results are obtained when anchors are at the corners.

3.2 Explicit Studies of Anchor Node Placement

While few, there have been some explicit studies of anchor node placement. Hara, et al. [14] propose a method of choosing anchor node locations to achieve a specific accuracy target. The proposal, however, only applies to rectangular network areas and that anchor nodes must be placed at the center of a sub-rectangle of the original rectangle when divided into equal sized rectangles. Further, it assumes simple RSSI-based localization.

Ash, et al. [15] provide analytical proof that placing anchor nodes uniformly around the perimeter of a network provides the best results, in the absence of any other information about the sensor node positions. However, again this assumes a rectangular network, and more importantly a simple localization algorithm like [13] or other multi-lateration techniques. When using all inter-node distances at once, as in MDS-MAP and CCA-MAP, this analysis breaks down.

Karl and Willig dedicate an, albeit short, sub-chapter to the *Impact of anchor placement* in their book [16, p. 247-248]. Referencing [13] and [17], again they defer to perimeter anchor placement as the optimal choice. Unfortunately, the technique proposed involves adaptive deployment, whereby a mobile node with absolute positioning available, like GPS, wanders through the network and attempts to determine the optimal anchor placements as it travels. For the purposes of a priori planning, this technique is not feasible.

Cheng, et all. [18] present a novel technique to handle the effects of adverse anchor placement, specifically in clumps. The algorithm, *HyBloc*, is a hybrid of MDS and proximity-distance map (PDM) [19]. HyBloc combines the two algorithms to draw on the best aspects of each. Namely, PDM is shown to have good performance in anisotropic networks, while MDS performs well even with few anchors. HyBloc, therefore, works by using MDS to add artificial, secondary anchor nodes in specific

isotropic areas of the network, and then uses PDM to complete the overall localization for the entire network.

Another study focuses on the effect of indoor conditions and anchor placement as it relates to RSSI and other radio propagation measurements [20]. The experiments were conducted in a small, enclosed space, and anchor nodes were placed either on the ceiling or the floor of the room. The study concludes that anchor nodes on the ground are better for monitoring moving people in the room, the extension of which is that anchor nodes need to be in the same plane as the nodes they are being used to locate.

3.3 Summary of Related Work

Overall, the previous studies on anchor placement are limited. Specifically, they focus on particular use cases and assumptions that are different from the ones here. The overarching theme of the studies, though, is to place the anchors at the edges of the network. Despite the different assumptions, this idea makes sense from a purely geometric point of view, and is therefore used as the initial basis of hypotheses in this thesis.

Chapter 4

Optimal Anchor Node Placements

4.1 Measuring Location Error

Before searching for the best anchor node placement we must first define what *best* means, in terms of location error. First of all, location error is the distance between the actual and calculated position of each node, measured as a factor of radio radius (or range). Since this study addresses range-free networks, and thus relies solely on network connectivity, the actual units of distance do not matter for general study. What is important to the protocol is how many other nodes in the network fall within the radio radius of a given node. The average number of nodes within range of each node is known as network density.

Every network has its own application requirements, and thus there are many options for what statistics to examine for assessing the quality of locations. The simplest criteria is to look at the mean location error across all nodes in the network. For the most part, this is the basis for the results in this study. However, this assumes that all nodes in the network must be used in the final results. If the network designers know which nodes have poor locations, they may wish to exclude these nodes from the final results. Therefore, if the economics of the network deployment allow, it

may be beneficial to look at the best, for example, 80% of nodes in the network. In practice, the designers do not know which nodes to exclude, so this study also attempts to identify a correlation between a particular node's position relative to the anchor nodes and its localization error. However, no such correlation was determined in our studies.

4.2 Coordinate Transformation

To understand the various hypotheses for the best anchor placement, a basic understanding of how the transformation from local to global coordinate systems is required. After the local networks have been calculated and patched together into a single cohesive network, the anchor nodes are then used. Using Procrustes [21] analysis, a linear transformation of translation, reflection, orthogonal rotation, and scaling is determined for the anchor nodes from the calculated to known coordinates. The transformation is chosen by minimizing the sum of squared errors for the resulting coordinates. Specifically, given actual anchor coordinates, Y , Procrustes gives a transformation as shown in Equation 4.1 that minimizes the difference between the actual coordinates and the calculated coordinates, Z . In the equation, b is a scalar component, T is the rotation/reflection component, and c is the translation component, each determined by the Procrustes analysis. The rotation/reflection component, T , is discussed in more detail in Section 5.1.1.

$$Z = b * Y * T + c \quad (4.1)$$

For example, take a random anchor set, as shown in Figure 4.1. The figure shows the three triangles, which are the local anchor coordinates, the actual global coordinates, and the predicted coordinates after applying the calculated transformation

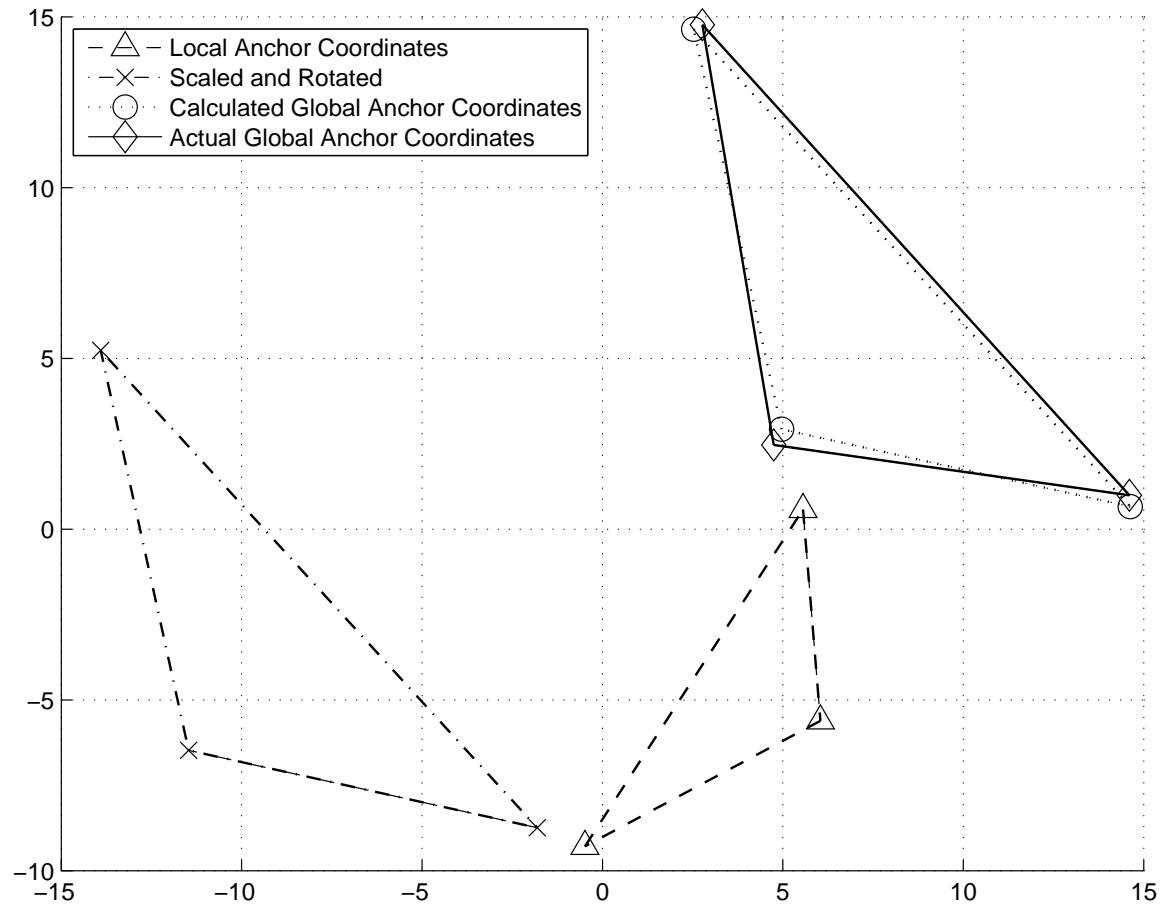


Figure 4.1: A sample anchor triangle Procrustes analysis

from local anchor coordinates to actual anchor coordinates. The transformation equation is shown in Equation 4.2. This demonstrates that the transformation calculated by Procrustes is not perfect, as the actual and calculated coordinates do not align exactly.

$$\begin{bmatrix} 2.5307 & 14.6367 \\ 14.6278 & 0.6594 \\ 4.9662 & 2.9248 \end{bmatrix} = 1.5975 \begin{bmatrix} -0.4861 & -9.2882 \\ 5.5578 & 0.5797 \\ 6.0389 & -5.6138 \end{bmatrix} \begin{bmatrix} -0.3030 & -0.9530 \\ 0.9530 & -0.3030 \end{bmatrix} + \begin{bmatrix} 16.4355 & 9.4010 \\ 16.4355 & 9.4010 \\ 16.4355 & 9.4010 \end{bmatrix} \quad (4.2)$$

The transformation is then applied to all nodes in the network by simply replacing Y in Equation 4.1 with the local coordinates for the entire network. Therefore, how well the anchor nodes transformation captures the overall local network variability dictates how good the final locations will be.

4.3 Methodology

For the purposes of testing each hypothesis, randomly generated networks of varying sizes are used. Unless otherwise specified, all nodes are randomly placed within a square area with an overall density of one node per unit area. For example, a $20x20$ square area will have 400 randomly placed nodes. Anchor sets are chosen by identifying all possible sets via n choose k , where n is the number of nodes in the network and k is the number of nodes per anchor set. For example, in a $20x20$ network of 400 nodes, and three nodes per anchor set, there are a possible 10,586,800 choices. From the total set of possible anchor sets, a random selection is made. For the data in this chapter, 1,000 anchor sets are randomly chosen. Choosing anchor sets from the n choose k population excludes the possibility of choosing the same anchor set more than once. To view all the Matlab© code for simulations and plots see Appendix A.

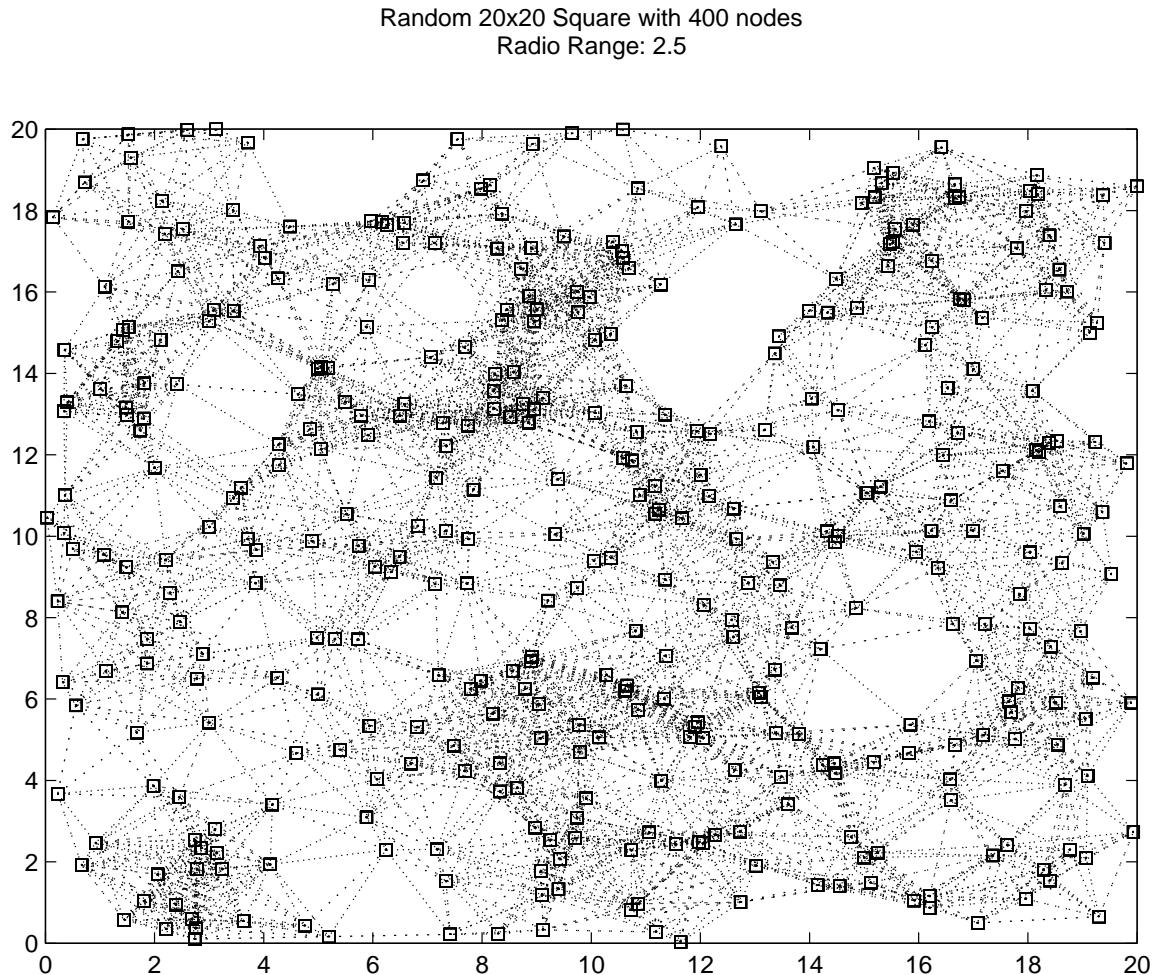


Figure 4.2: The sample network used in this chapter, showing connectivity between nodes

4.4 Anchor Node Placement Metrics

Almost as important as knowing which factors effect localization is knowing which factors to ignore. The following is a summary of hypotheses of anchor node placement factors that show no or little significant correlation to location error. The data in the following graphs is taken from a random selection of 1000 anchor sets of three nodes each for the network shown in Figure 4.2. The node placement is random within the 20 by 20 square unit area, and all nodes have a radio range of 2.5 units, providing a

completely connected network.

4.4.1 Anchor Node Error

The first hypothesis is to choose nodes as anchors that we expect to have high location accuracy. Because we are trying to provide an a priori design technique for network planners, the information chosen to reach this goal must be something the network planner can determine before any localization has been performed. For this purpose we choose the number of one-hop neighbors. On average, increased network density results in higher localization accuracy [2, 3] Since the range-free algorithms depend heavily on network connectivity, the theory is that nodes with more neighbors will have lower location errors. The lower error in the anchors themselves should then translate into a more accurate transformation of the entire network.

Figure 4.3a plots the sum of unique one-hop neighbors for all anchor nodes versus mean localization error. The irregularity of the curve disproves the hypothesis. Further, the correlation coefficient, with 95% confidence intervals, is extremely low, less than 0.1. Based on these statistics, choosing anchors in denser parts of the network does not translate into better network-wide localization accuracy.

Further, we explore if having anchors with lower localization error translates into lower localization error across the network. Figure 4.4a plots the mean error of the anchor nodes themselves versus the mean error for all nodes in the network. This essentially measures how well the transformation generated by Procrustes analysis matches the target. The erratic plot and low correlation coefficient suggests there is no foundation for this hypothesis.

In both plots, it is clear that a small subset of the anchor placements lead to far greater error than the norm. Later, we will explore the cause of these outliers, but for the time being, we exclude them from the analysis as shown in Figure 4.3b

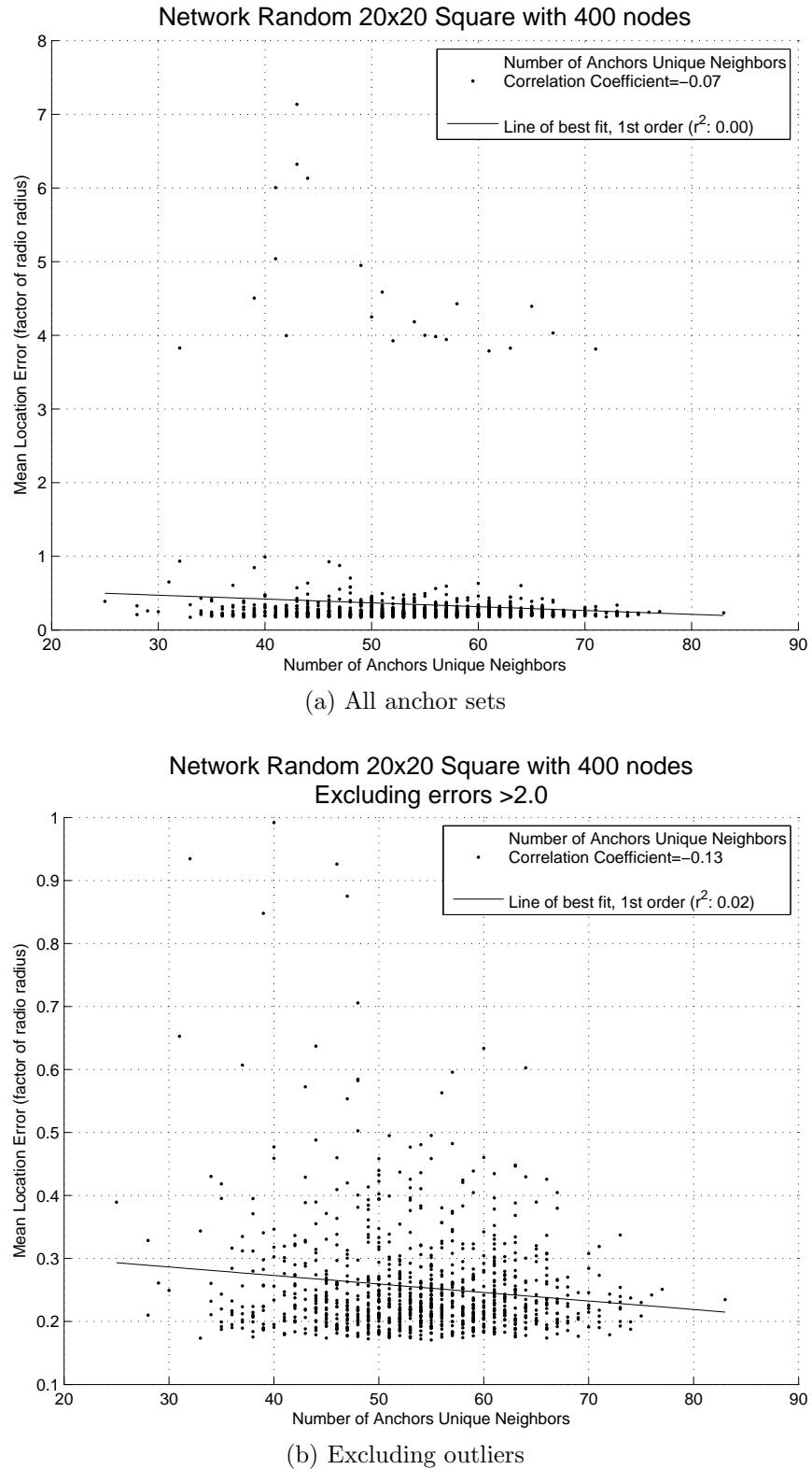


Figure 4.3: Number of anchor neighbors vs. location error

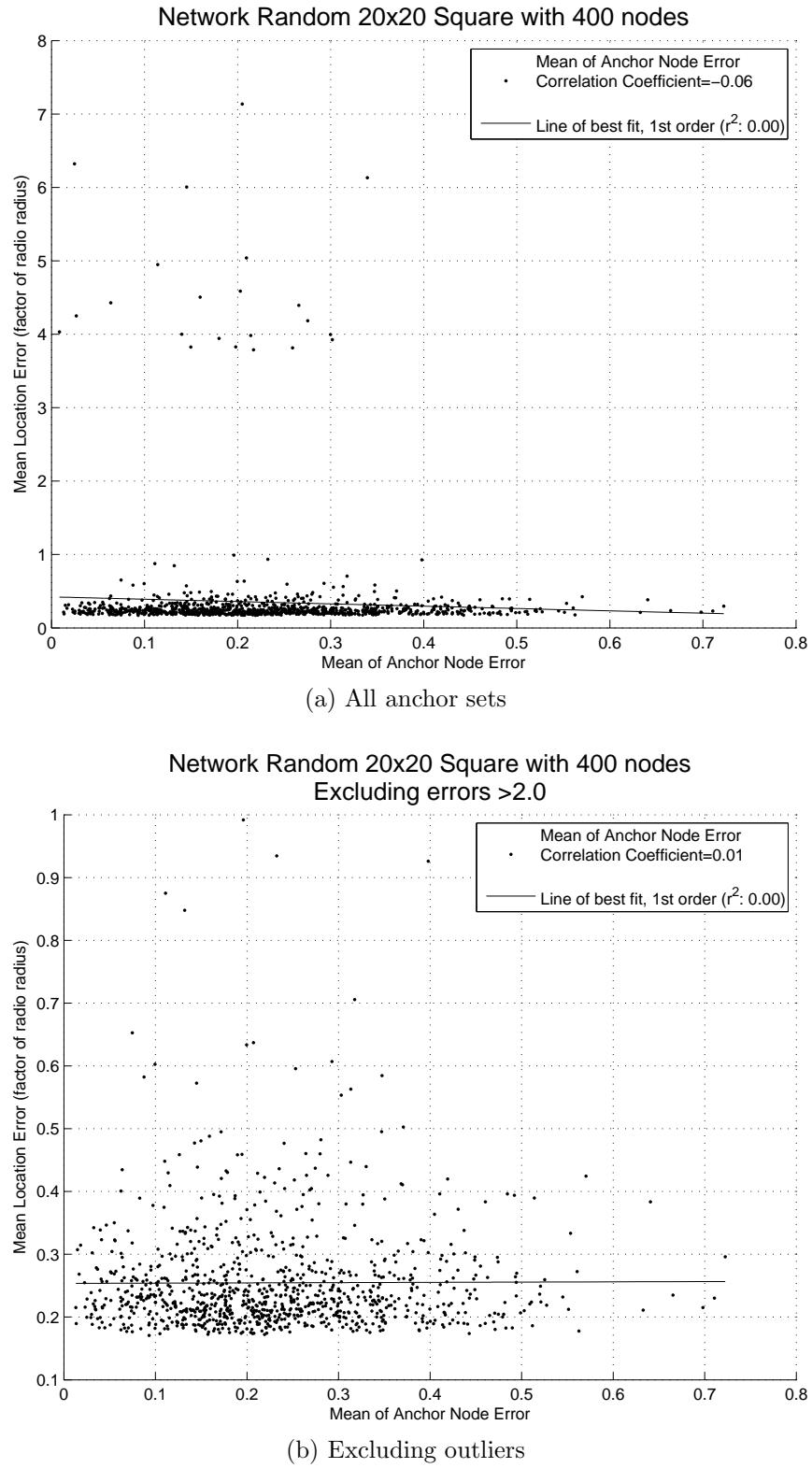


Figure 4.4: Mean of anchor node error vs. location error

and Figure 4.4b. Determining which data points are outliers is quite obvious since the gap between the normal case and the small set of outliers is distinct. Therefore, rather than using a statistical calculation for the exclusion, all anchor sets with mean location error greater than two are excluded.

Even after the outliers are excluded, no pattern emerges and the correlation coefficients remain extremely low. Further, the line of best fit has a low r-squared value of 0.02. Higher order best fits also show no clear correlation.

4.4.2 Network Area Coverage

As suggested in related studies in Chapter 3, an optimal anchor placement is to have the anchors as far apart as possible, around the edges of the network. The rationale behind this hypothesis, in the context of this study, is that if a wider area of the network is covered by the anchor nodes, then the resulting calculated transformation will take into account more network variations.

One way to determine how close the anchors are to the perimeter of the network is to measure how far apart the anchors are from each other. The further apart they are, the closer they are to the edge and the more network area they cover. For statistical purposes, the sum of the distance between each pair of anchors is taken. I describe this as the sum of distances of the anchors instead of the perimeter of the triangle formed because this hypothesis investigates if the further apart the anchors are from each makes a difference. Further, the sum of distances scales for the case of more than three anchor nodes, whereas the meaning of perimeter will change the desired description. Figure 4.5 shows an example for both the three and four anchor node case. For the three node case, it is equal to the perimeter of the triangle. More generically, for more nodes, as shown here for four, it is the sum of all pairwise distances.

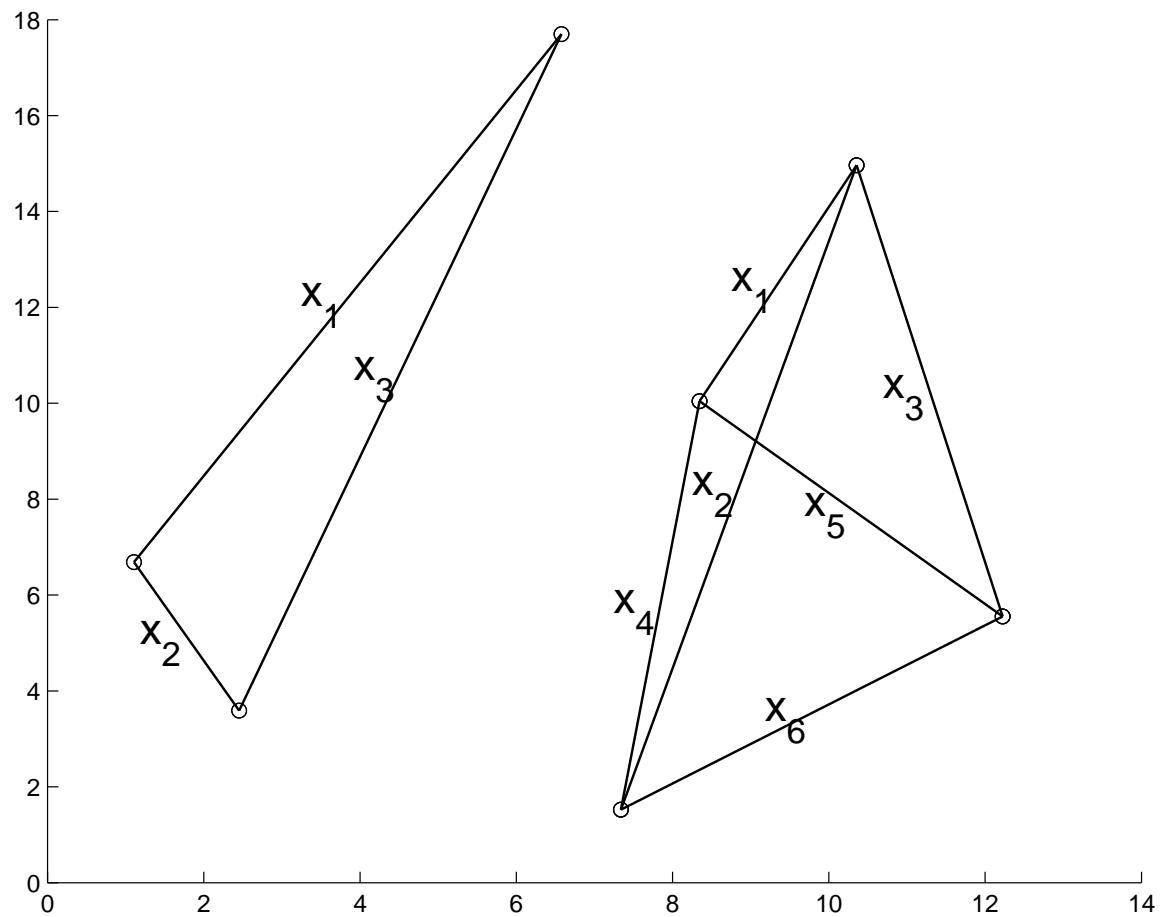


Figure 4.5: Sum of Distances = $\sum_{i=0}^n \|x_i\|$

Figure 4.6a shows the sum of distances for each anchor set versus mean localization error for that anchor set. The plot shows that again there is low correlation between the distance between anchors and the location calculation. Further, the outliers are spread relatively evenly regardless of distance between anchor nodes, and thus this is not a good indicator of an outlier.

When the outliers are excluded, as shown in Figure 4.6b, a moderate level of correlation is seen, with a much clearer line of best fit. The only moderate, 0.48, correlation coefficient can be explained by the lower bound for location error. There is a virtually straight line across the bottom of the data indicating that, at any distance between anchors, the lower bound of mean location error can be reached. However, as the distance increases between the anchor nodes, the probability of getting a high location error decreases.

Even if the sum of distances of the anchors is high, it is possible for two anchors to be very close together and far from the third. Therefore, the minimum distance between a pair of anchors is shown in Figure 4.7a. Again, the correlation coefficient is low at 0.20. However, it does appear that the outliers are slightly more likely to appear when the minimum distance between a pair of anchors is low. Further, when outliers are excluded, as shown in Figure 4.7b, the coefficient is 0.32, slightly lower than that of the sum of the distances, implying that sum of the anchor distances is a better indicator of localization performance than minimum anchor distance.

4.4.3 Anchor Node Triangle

Continuing the trend of trying to show some increased correlation between the anchor coverage and location error and based on speculative statements by Shang in [2, p. 964] presented in Section 3.1, this section examines two additional metrics attempting to measure collinearity of the anchor nodes: the area and height of the anchor node

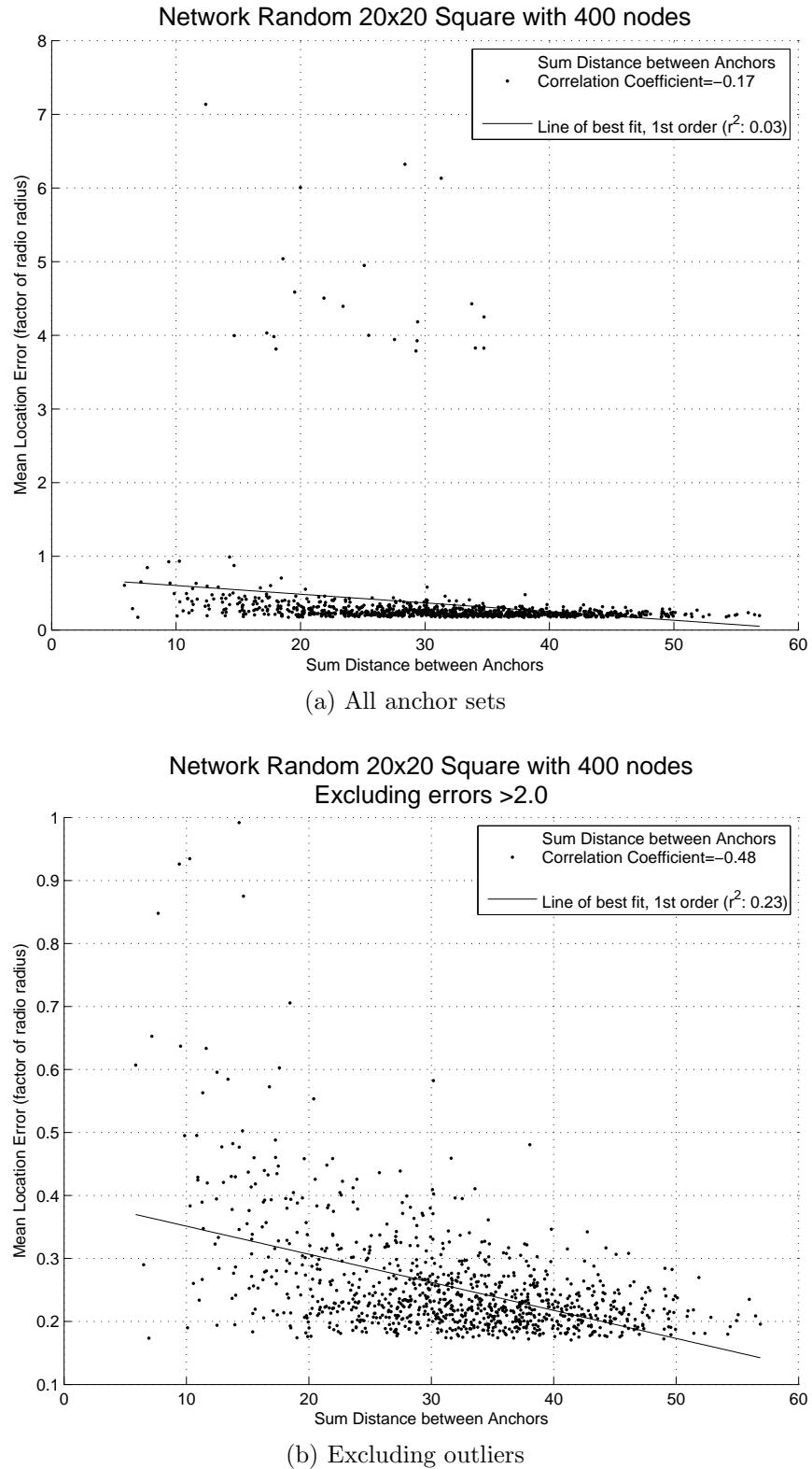


Figure 4.6: Sum of distance between anchors vs. location error

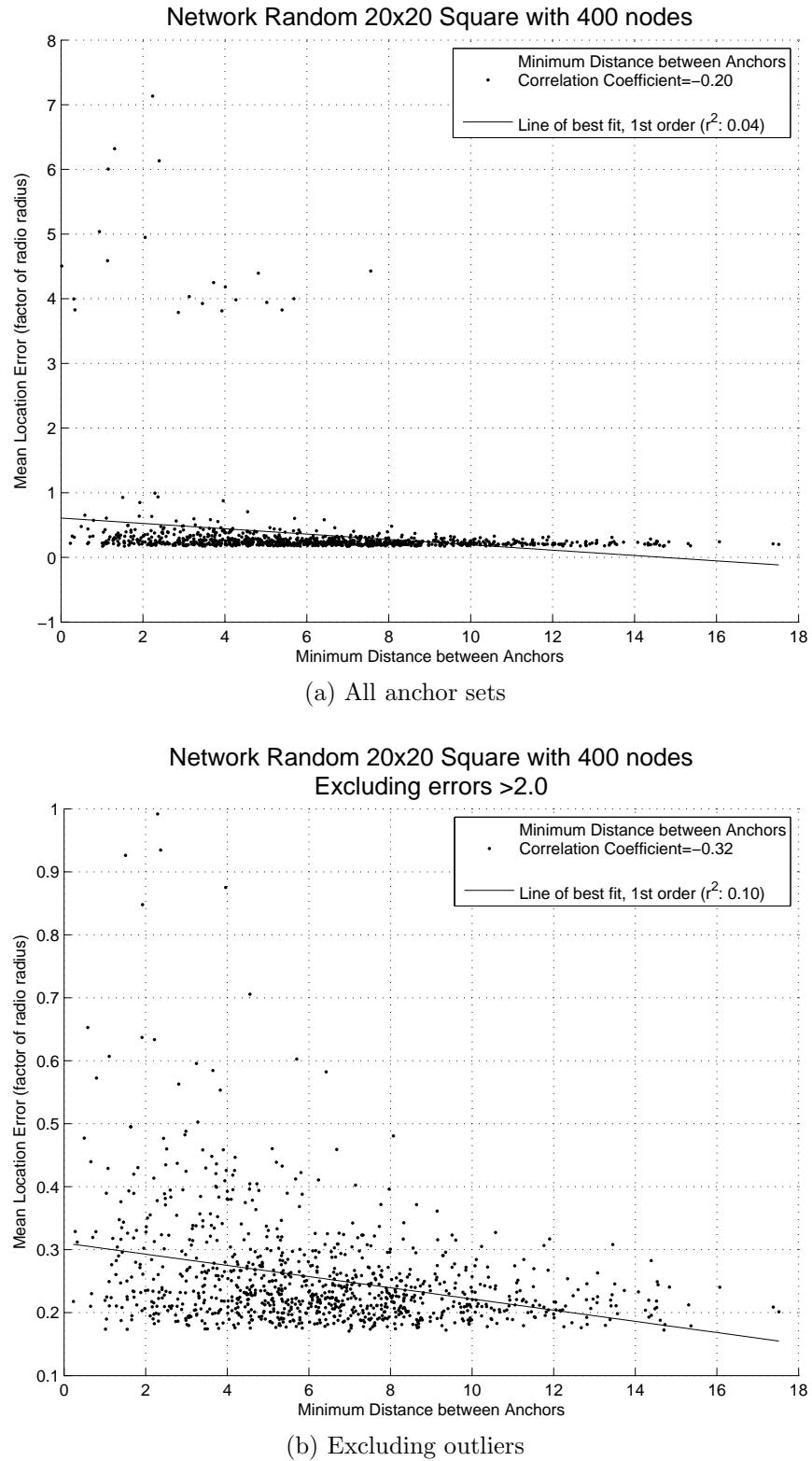


Figure 4.7: Minimum distance between anchors vs. location error

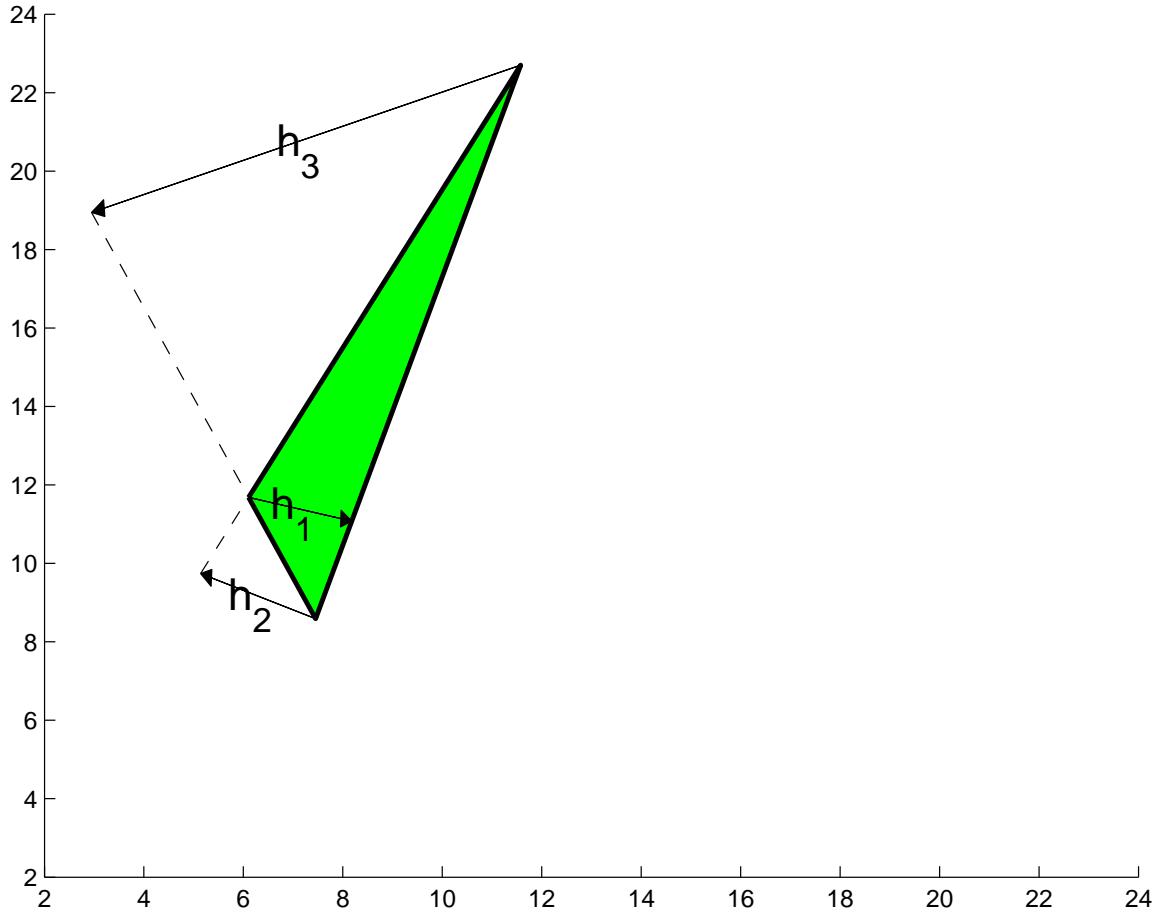


Figure 4.8: Area and height of a triangle
 $\text{Minimum height} = \min(\|h_i\|)$

triangle. For height, we calculate the height in each direction of the triangle, and use the shortest value. Both metrics can be extended to more anchor nodes. Figure 4.8 shows a simple example of calculating the area and minimum height of a triangle.

Figure 4.9 plots the area of the triangle formed by the three anchor nodes versus mean location error. This metric can also be extended to more anchor nodes, by using the area of the polygon formed by the anchor nodes. Figure 4.10 does the same for the shortest triangle height. The first observation from the two plots is that the outlier cases are more tightly correlated to anchor area and height than any of the other metrics. The outlier points are all closer to the y-axis rather than being spread across

the plot. This is explored further in Section 5.2. Secondly, after removing outliers, the anchor area has a slightly higher correlation coefficient than anchor height, indicating that anchor area is a slightly better predictor than anchor height.

4.5 Best Anchor Node Placement

Analyzing all the metrics discussed above, the best indicator of a good anchor placement is the sum of the distance between anchor nodes, once the outliers are excluded. Avoiding the outlier case is discussed in Chapter 5. Practically speaking, this means that network designers can choose between proposed anchor placements and use the measured distance between them to provide guidance about which placement will result in the least amount of error.

Figure 4.11 uses data from 20 networks, each occupying an area of 20x20 units with a radio range of 2.5 units. The networks have a density of one node per unit squared, but all nodes are randomly placed in each network. 5,000 anchor sets of three nodes each are chosen from the randomly placed nodes. The outliers are then removed from the data and the sum of distance between the anchors is plotted against the mean location error. The data is then grouped into two-unit intervals and a mean is calculated for each interval. A horizontal bar through the mean of each interval displays the width of that interval. Further, a 95% confidence interval is shown for each mean value.

This plot shows us that as the sum of distances between the anchors approaches ten times the radio range, the differences in location error are statistically insignificant. Therefore, network designers can choose anchors nodes locations at their convenience, as long as they meet this criteria. The actual asymptote value depends on other factors, like network density and will be discussed further in Chapter 7.

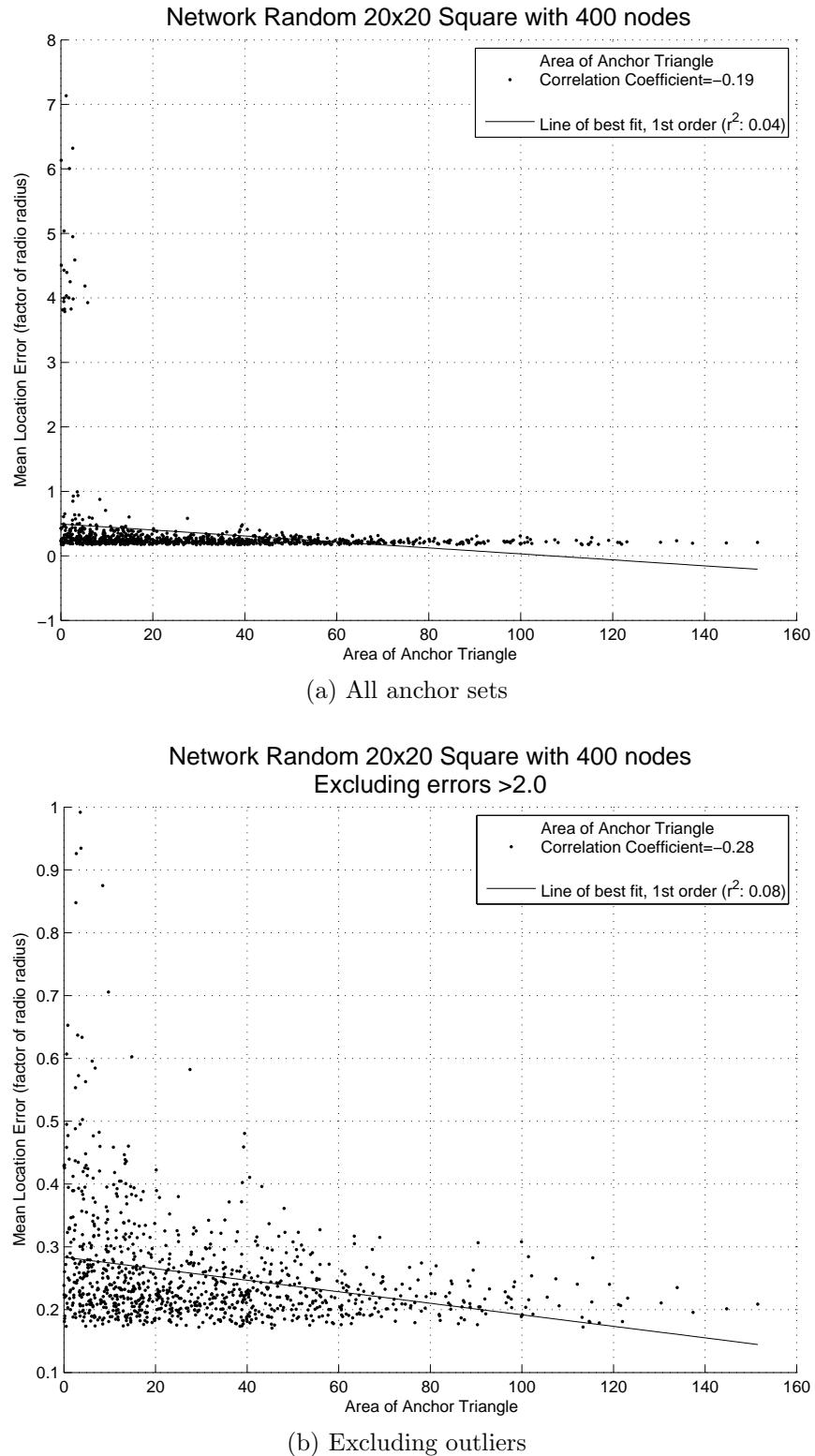


Figure 4.9: Area of anchor triangle vs. location error

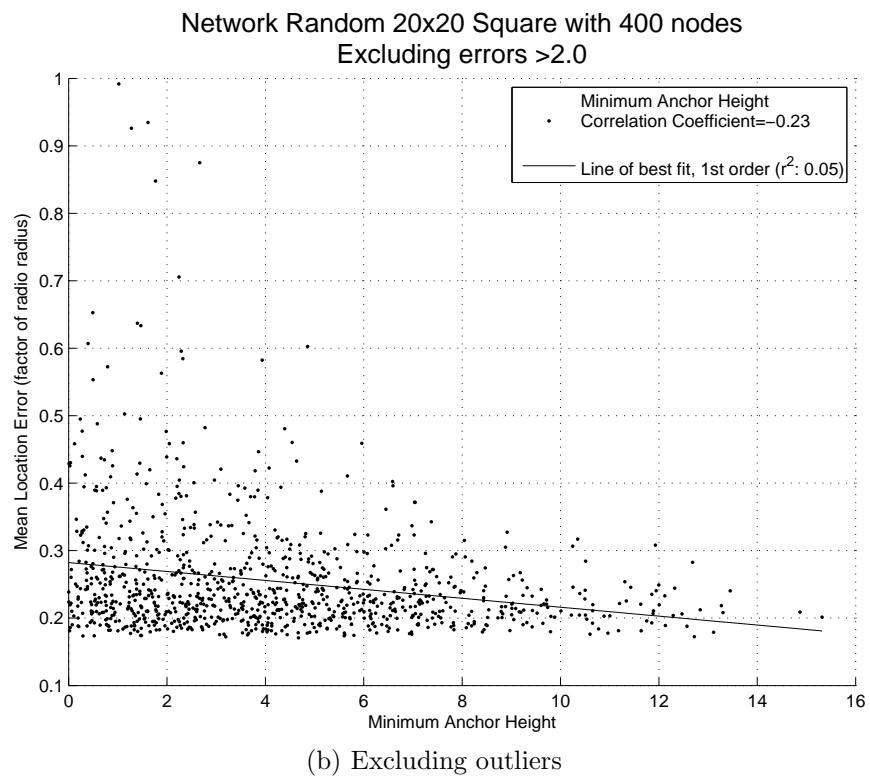
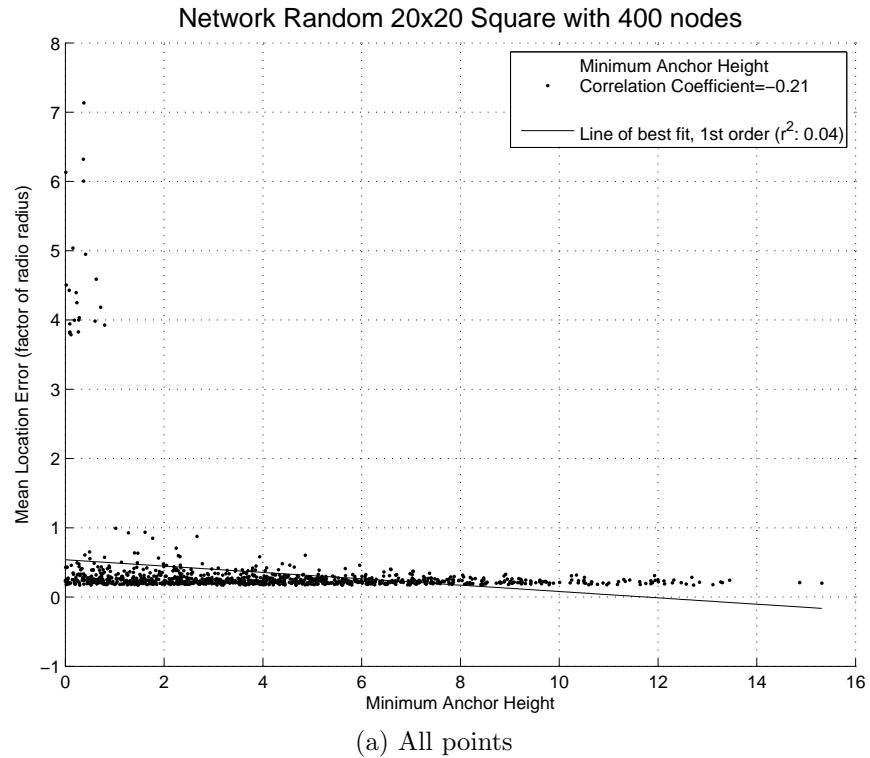


Figure 4.10: Minimum height of anchor triangle vs. location error

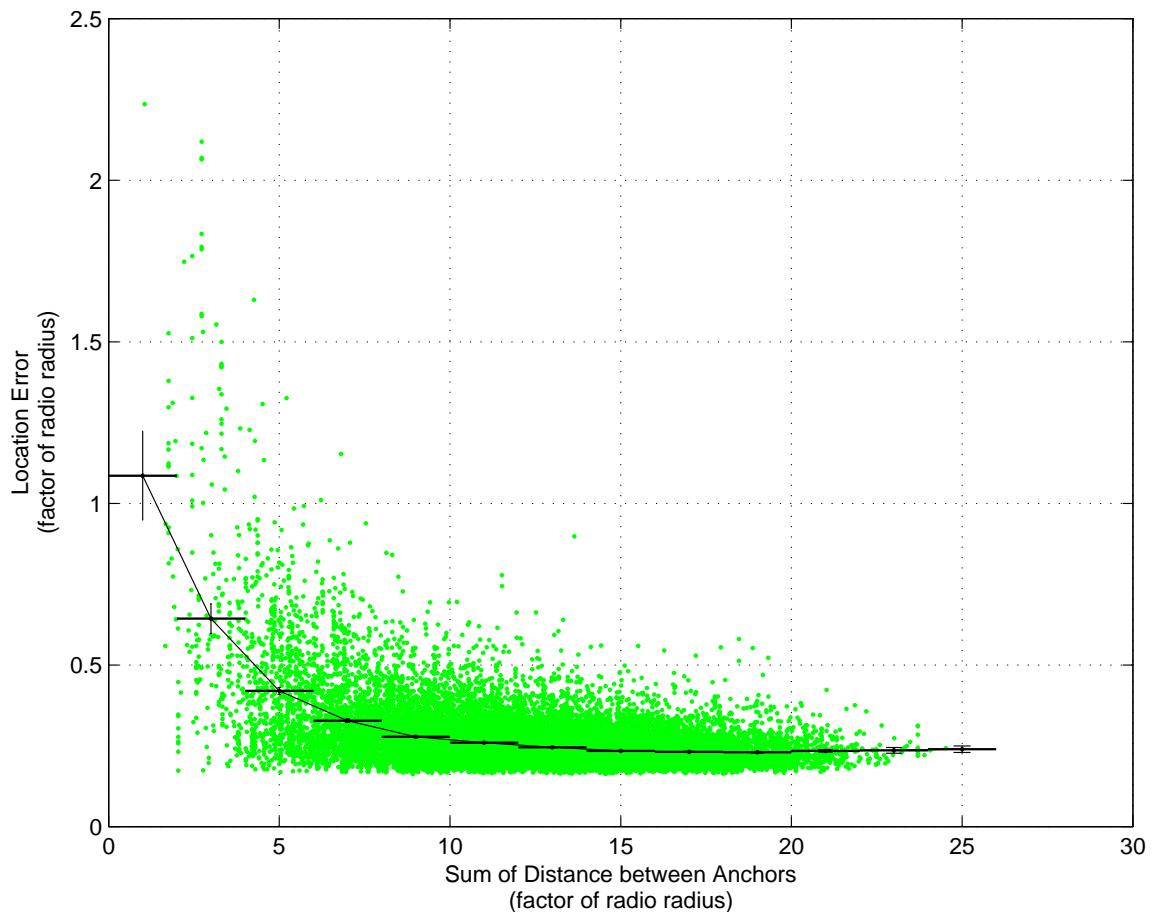


Figure 4.11: Sum of distance between anchors vs. location error for 20 random square 20-by-20 unit networks with 400 nodes, 5,000 anchor sets each, and a radio range of 2.5 units, excluding outliers

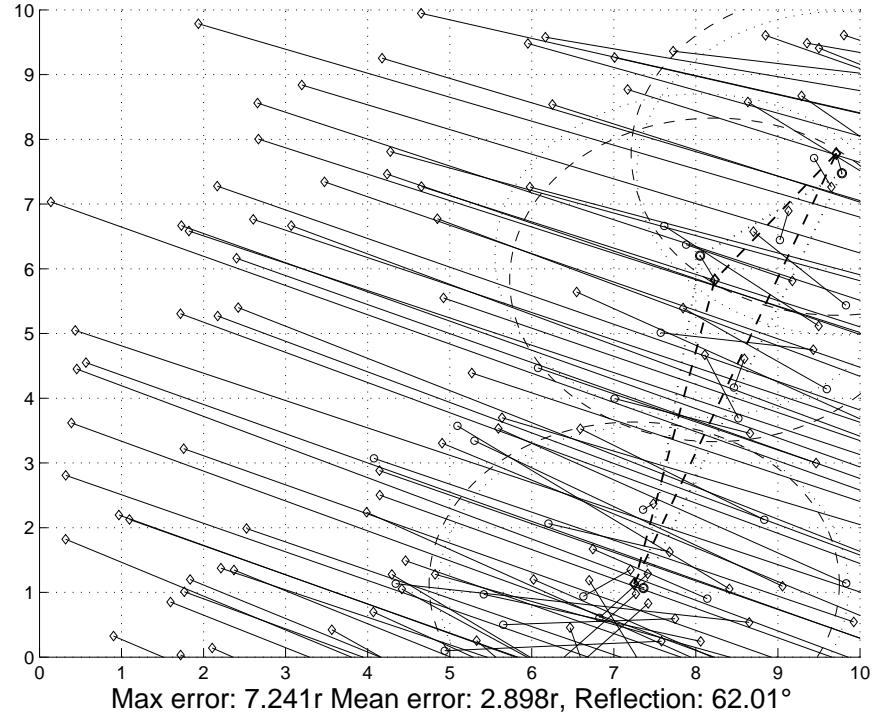
Chapter 5

Avoiding the Worst Anchor Node Placements

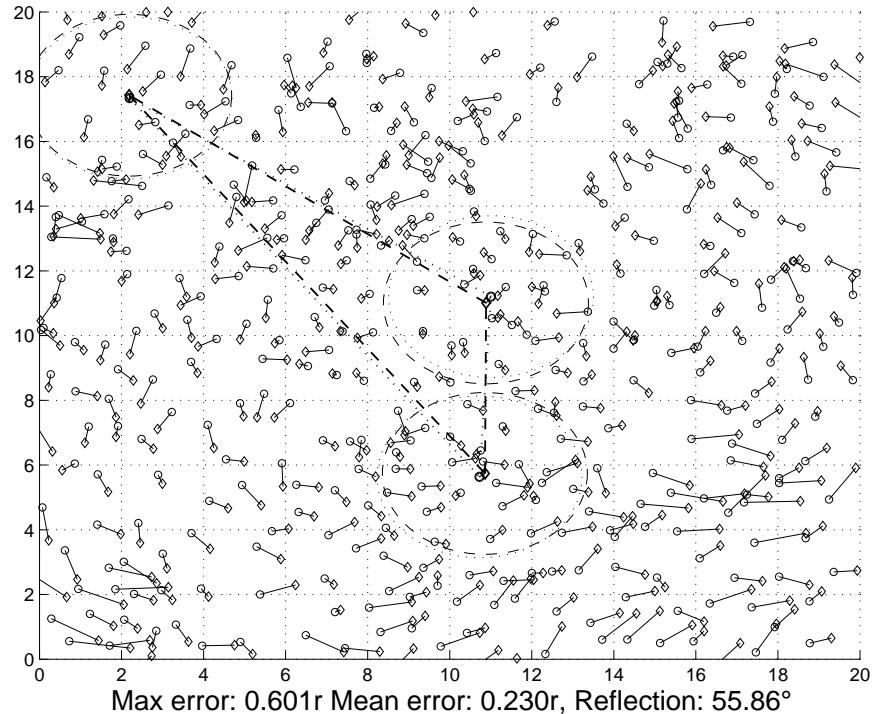
As seen in the previous chapter, there are some anchor node placements that are significantly worse than the average case. In this chapter, we explore in more detail the cause of these outliers and more importantly, is this condition detectable a priori and hence preventable in the real world.

5.1 Effects of Procrustes Analysis

Figure 5.1a shows a network map of an example outlier anchor node placement for a random network. A line is drawn between the real location and the calculated location for each individual node. The length of this line is the location error. For purposes of comparison, Figure 5.1b is a randomly chosen non-outlier case, where the node difference lines are short, representing low localization error. However, in the outlier case, the node difference lines criss-cross the network along a clear, single angle. This visual representation suggests the reflection component of the final linear transformation is to blame for the extremely poor, outlier results.



(a) An outlier network difference



(b) A normal network difference

Figure 5.1: Network map of localization performance at each node

5.1.1 Transformation Reflection and Rotation

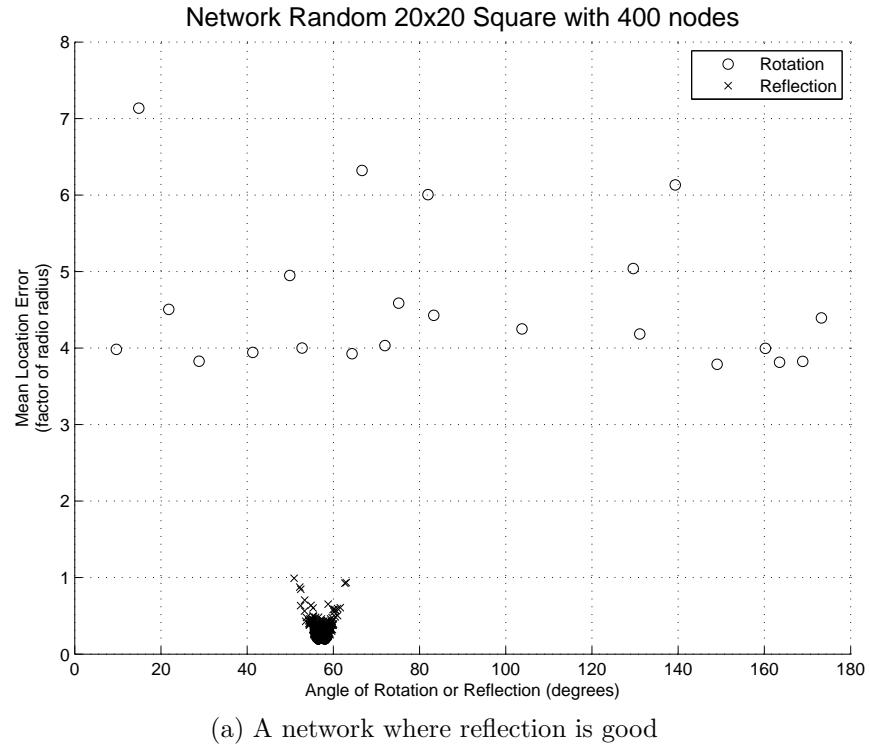
Unfortunately, simply disabling the reflection component of the Procrustes transformation algorithm does not solve the problem. The output of the Procrustes algorithm is a linear transformation which includes a rotation or reflection matrix, as discussed in Section 4.2. If the determinant of that matrix is +1, then the resulting transformation has a rotation, with an angle as in Equation 5.1.

$$\det(T) = +1 \Rightarrow \text{Rotation with } T = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (5.1)$$

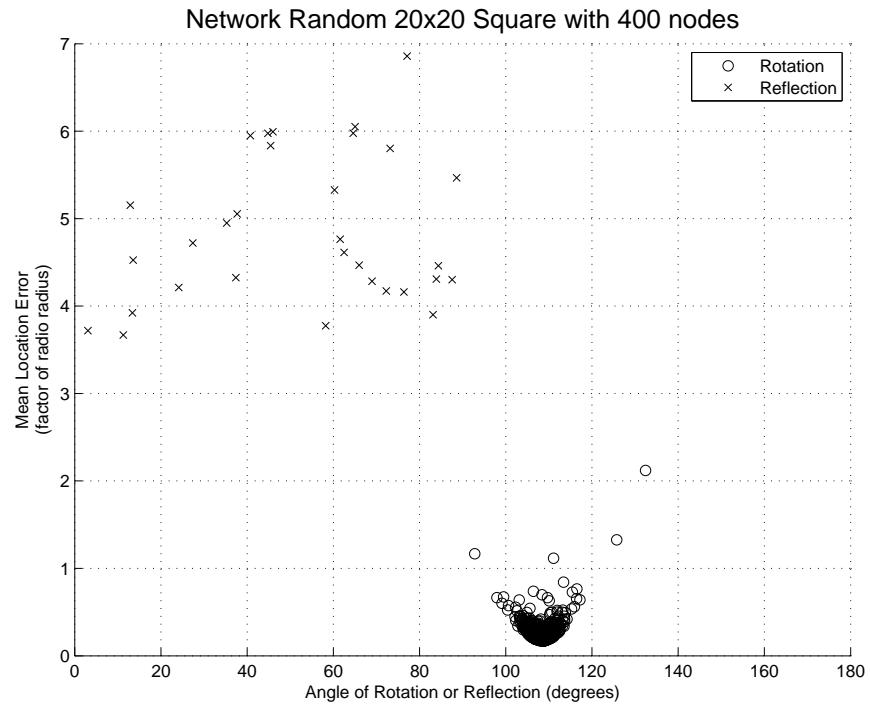
If the determinant is -1, then the resulting transformation has a reflection component across a line at an angle as shown in Equation 5.2.

$$\det(T) = -1 \Rightarrow \text{Reflection with } T = \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix} \quad (5.2)$$

Figure 5.2 shows the rotation and reflection distributions of two different networks, for a random set of anchor sets for each network. In both networks, and with consistency across others, the bulk of the data points have the same angle of either rotation or reflection, while the outliers have the opposite property with a wide variance in the angle. However, in different networks, it is not consistently either rotation or reflection that leads to poor localization results. Therefore, relying on the determinant of the transformation is not a sufficient indicator for a network designer knowing that an outlier case has been detected and that the localization results are essentially useless.



(a) A network where reflection is good



(b) A network where rotation is good

Figure 5.2: Rotation and reflection versus location error

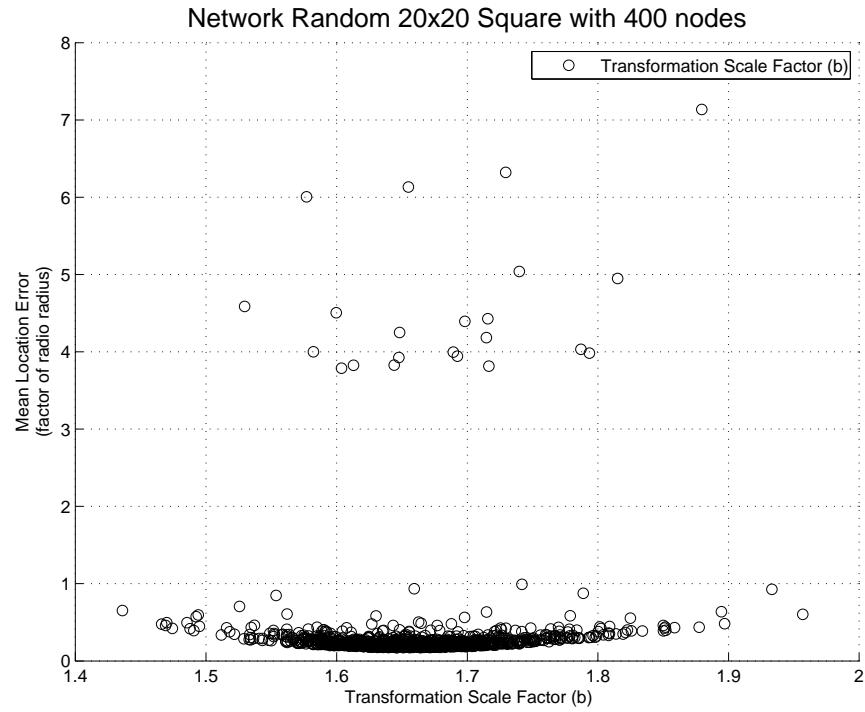
5.1.2 Transformation Scaling and Translation

For completeness, we explore the possibility that the scaling or translation component of the linear transformation is causing the issue. In Figure 5.3, the scale component, b , of the Procrustes analysis is plotted against the mean location error for the same two networks as shown above for reflection and rotation in Figure 5.2. While there is an apex of the data around a particular scalar value where the minimum localization error is seen, the outliers can also have this same scalar value. Therefore, the scaling component is not a good indicator of extremely poor localization performance.

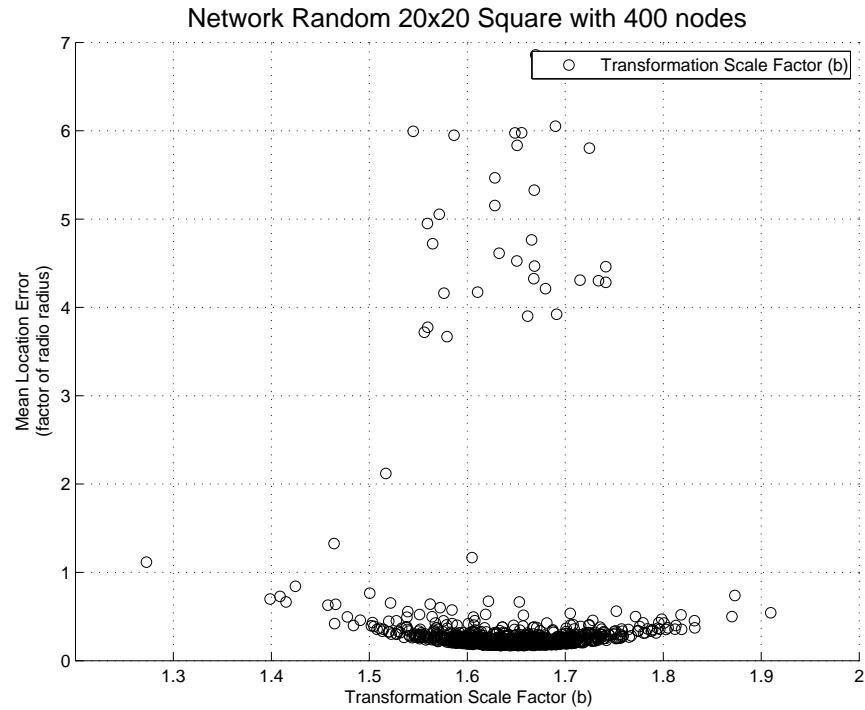
Likewise, Figure 5.4 plots the horizontal, x , and vertical, y , translation components, c , against the mean location error. As with the other transformation components, there is concentration around a *natural* value in x and y that gives best localization performance. The fact this value x and y overlap in Figure 5.4a is a coincidence. However, much like for the scaling factor, while outlier localization errors are the only values that occur away from this natural value, the outliers also occur around that value as well. Therefore, it is not possible to classify an anchor set as one of the outlier cases based on the translation component of the linear transformation it produces.

5.1.3 Procrustes Dissimilarity

The Procrustes algorithm itself provides a measure of the dissimilarity. Specifically, it is the minimized value of sum of squared errors [21]. As shown in Figure 5.5, it is not a good indicator of the transformation as it pertains to the entire network. This is because the Procrustes algorithm is only performed on the anchor nodes themselves and for those nodes themselves, the transformation is good.

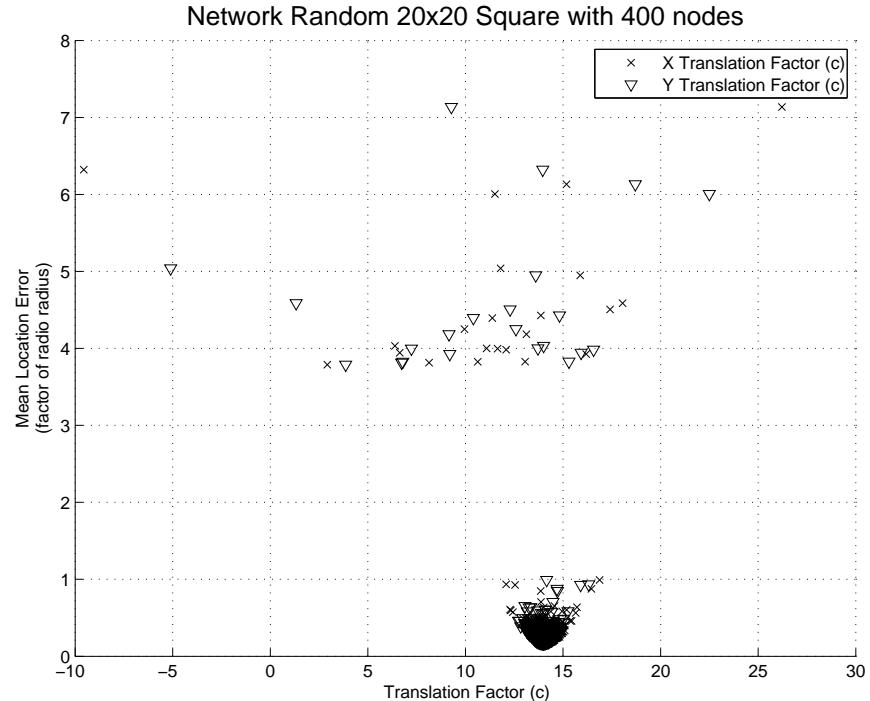


(a) Corresponding to a network where reflection is good

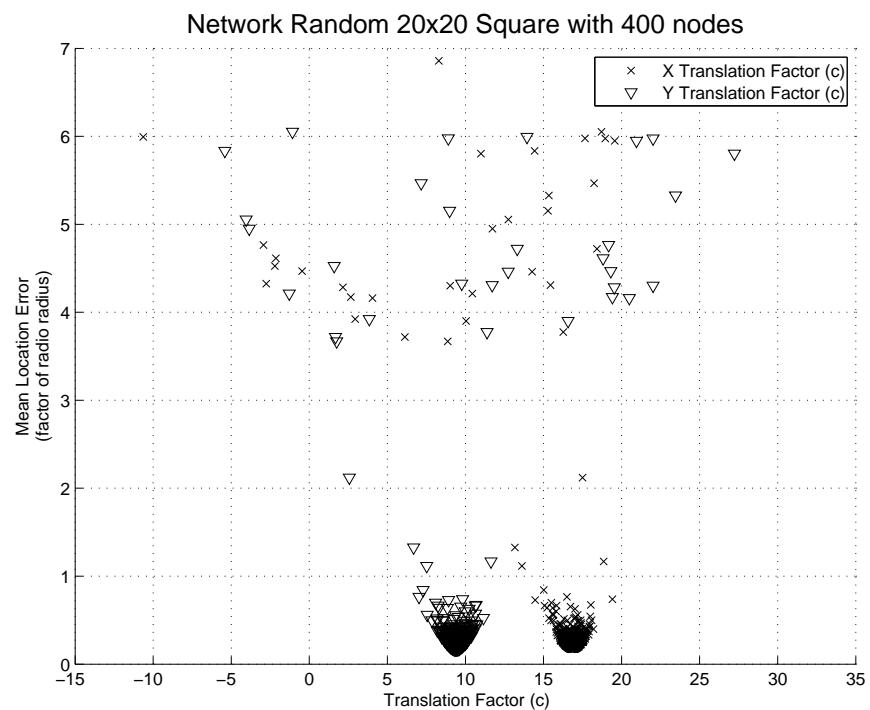


(b) Corresponding to a network where rotation is good

Figure 5.3: Transformation scalar component versus location error



(a) Corresponding to a network where reflection is good



(b) Corresponding to a network where rotation is good

Figure 5.4: Transformation translation component versus location error

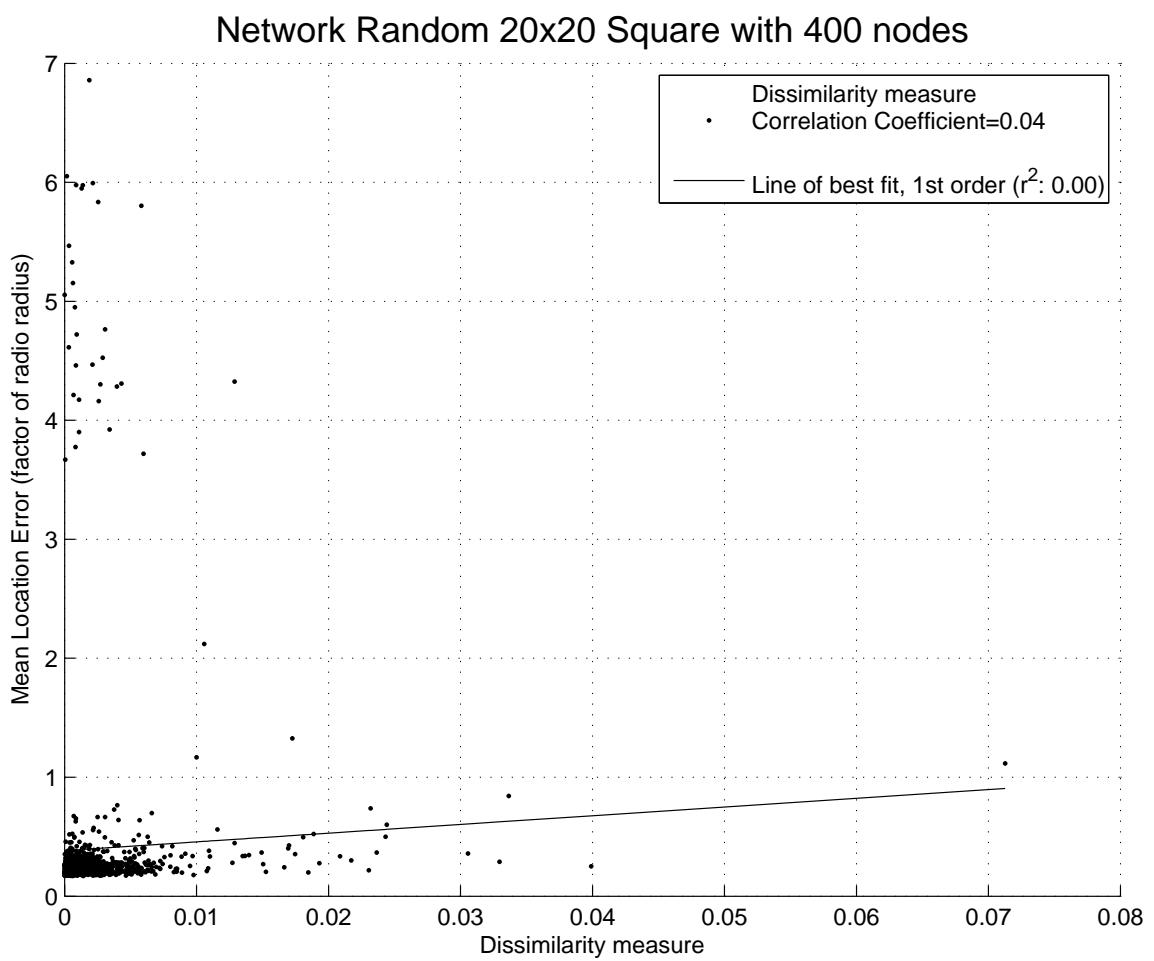


Figure 5.5: Procrustes dissimilarity measure

5.2 Outlier Indicators

As shown above, there unfortunately is no direct indicator that a particular anchor set generates a transformation with an incorrect angle. Therefore, network designers must avoid all anchor sets that could potentially generate such an angle. As demonstrated in Section 4.4.3, the area or height of the triangle formed by the anchor nodes is a good indicator of the possibility of a poor transformation angle. These two metrics are explored in Figures 5.7 and 5.6, respectively. This time, the x-axis is a log scale to better show the outlier cases. Also, the area and height itself has been adjusted as a factor of radius. Further, 95% confidence intervals are shown for the mean of each 0.1r increment of both area and height. A solid horizontal line is shown through each mean, indicating the width of the increment, since it can be difficult to visualize on a log scale. As the area and height increase, the mean expected area and height decrease, as does the confidence interval of that mean. The horizontal dashed line indicates the cutoff for what are considered outliers, based on the large gap in data points. The vertical dashed line indicates the first interval in which there are no outliers.

Upon comparing the two plots, it is clear that the minimum height metric is far more stable when it comes to predicting outliers than the area metric. This is based on the observation that the mean localization error monotonically decreases with increased minimum height, while the mean localization error relative to area is far more erratic. Further, there are some outlier cases after the first interval of triangle area that has no outliers. A triangle with small area might provide a false-positive indication of an outlier. After careful thought, this result is not unexpected since height of a triangle geometrically exposes collinearity better than simply area of a triangle.

Based on the minimum height statistics, we can now give network designers a

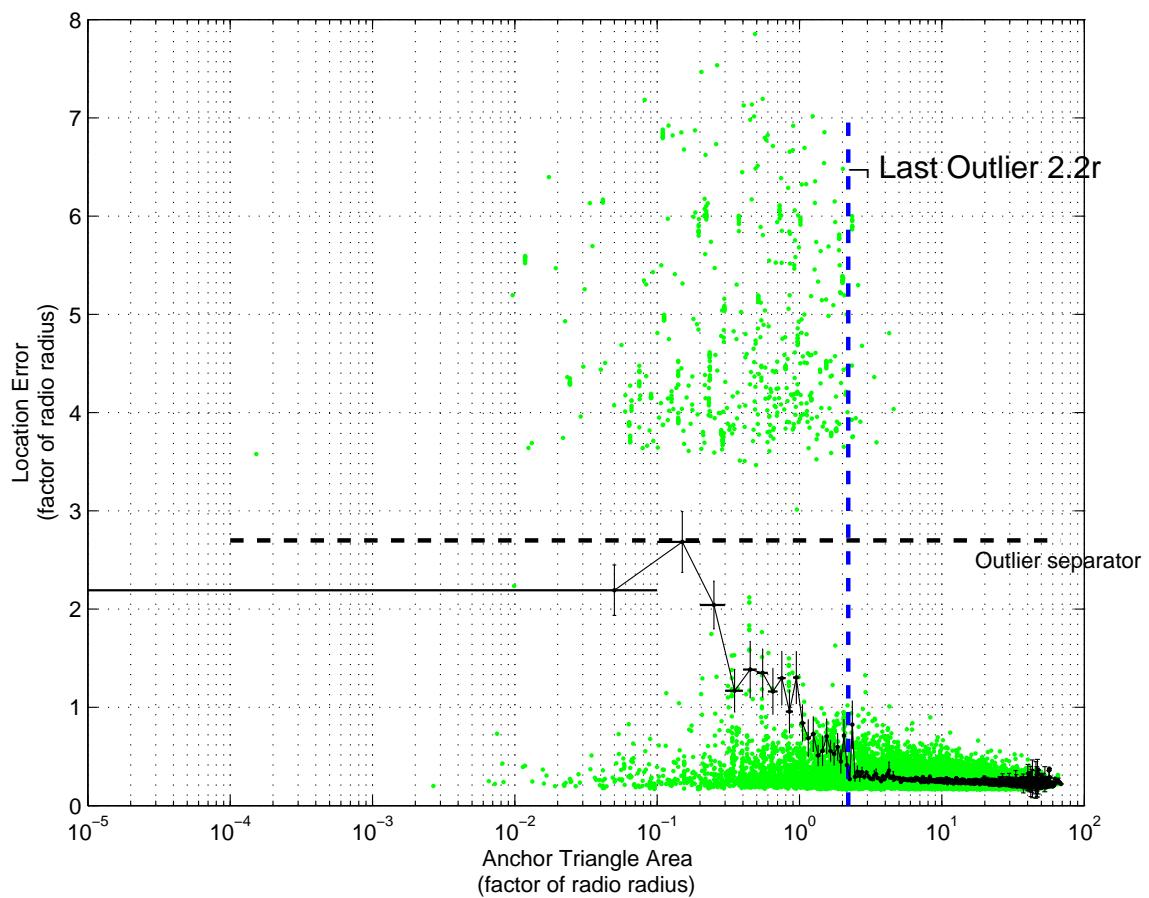


Figure 5.6: Anchor triangle areas, with confidence intervals grouped in intervals of $0.1r$

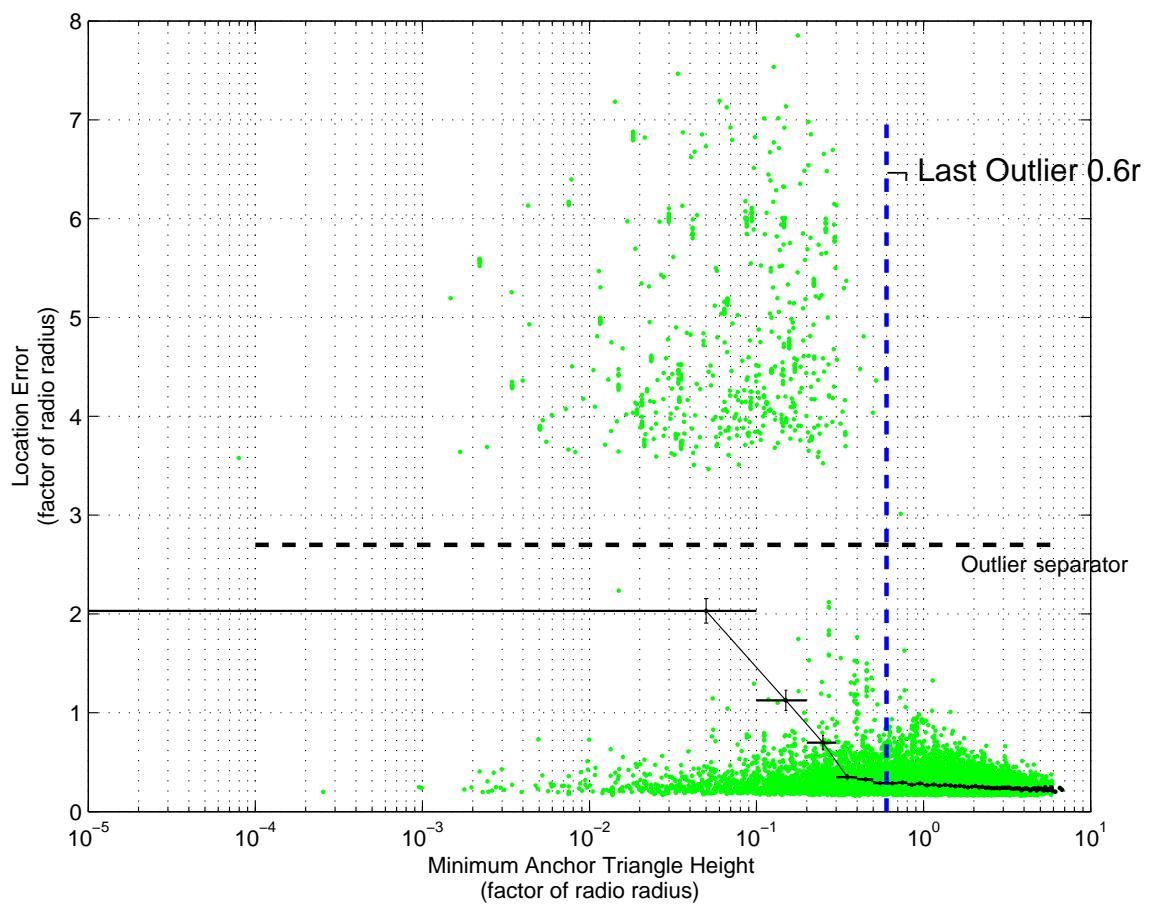


Figure 5.7: Minimum anchor triangle heights, with confidence intervals grouped in intervals of 0.1r

metric of how collinear is too collinear for a set of anchor nodes. The raw statistics indicate that if the triangle formed by the anchor nodes has a minimum height greater than 0.6 times the radio range, then there is very low probability of getting an extreme outlier case where the calculated locations are actually reflected or rotated across the network. However, also in the data shown, there are some near-outlier cases beyond this point. While these cases are not of the extreme nature discussed, it is worthwhile to give a margin of error here to avoid these cases as well. Therefore, we assert that the triangle formed by the anchor nodes should have a minimum height equal to the radio range of the network.

Chapter 6

Effects of Network Topology

6.1 Node Distance from the Anchors

Of great assistance to network designers would be to know in which regions of the network to expect poor localization performance. With this information, and if the anchor placement is constrained, they could either avoid placing nodes in the expected poor area, or take into account the higher expected localization errors in the analysis of the data.

Figure 6.1 shows a different view of in-network errors. Instead of showing a line representing the localization errors as in other figures present in this study, the error at each node is used to interpolate a grid of location errors throughout the area of the network. A contour plot, based on that interpolated grid, is then shown in the figure. The goal in viewing the network map in this way is to discover a pattern in terms of the relationship of a region of the network to the location of the anchor nodes. Unfortunately, no geographic correlation between location error and anchor node placement can be ascertained. It may be possible, with future study, to remove the effect of other factors to reveal a pattern based on anchor placement.

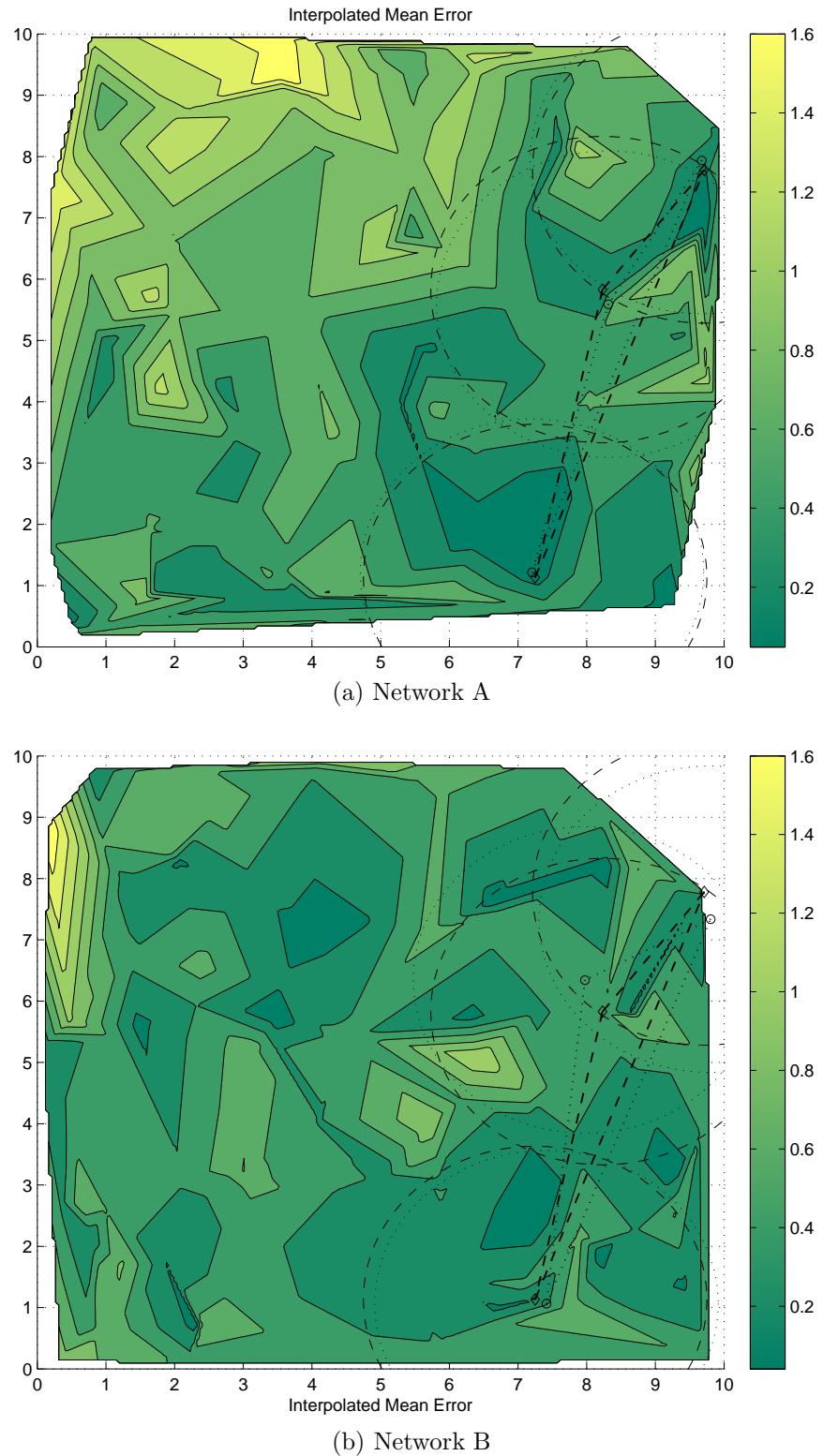


Figure 6.1: Localization error as a contour plot

6.2 Applicability to Varying Network Topologies

So far, we have examined anchor node placement for a continuous square network, with randomly placed nodes. However, in the real world, networks are not always so simple. There are often regions in the network where it is not possible to put nodes, due to physical barriers like lake, buildings or access to property. In this section, we look at how the results thus far apply to C-shaped and long, narrow pipeline network topologies.

6.2.1 C-Shape Network Topology

A C-shape network consists of a relatively square region with an empty area on one side, as shown in Figure 6.2. In terms of anchor placement requirements, we do not expect any difference between the recommendations presented for square networks.

To show that there is no difference between C-shape and square network topologies, 10 random C-shape networks with 5,000 anchor sets each was simulated in the same manner as the square networks. Figure 6.3a shows the location error relative to the sum of the distances between the anchors, as in Section 4.5. As expected, the mean location error flattens out as the sum of distances between anchor nodes reaches about 10 times the radio range as was the case with the square network. There is a slight increase in the floor of the mean location error as compared with the square network, but this has to do with the performance of the CCA algorithm in the presence of the empty region of nodes in the network. The increase in mean location error at the end of the plot, and specifically the increase in the confidence interval, is caused by the small sample size in the random selection of anchor set with a very high distance between nodes.

Similarly, the same criteria for eliminating the outlier localization results also

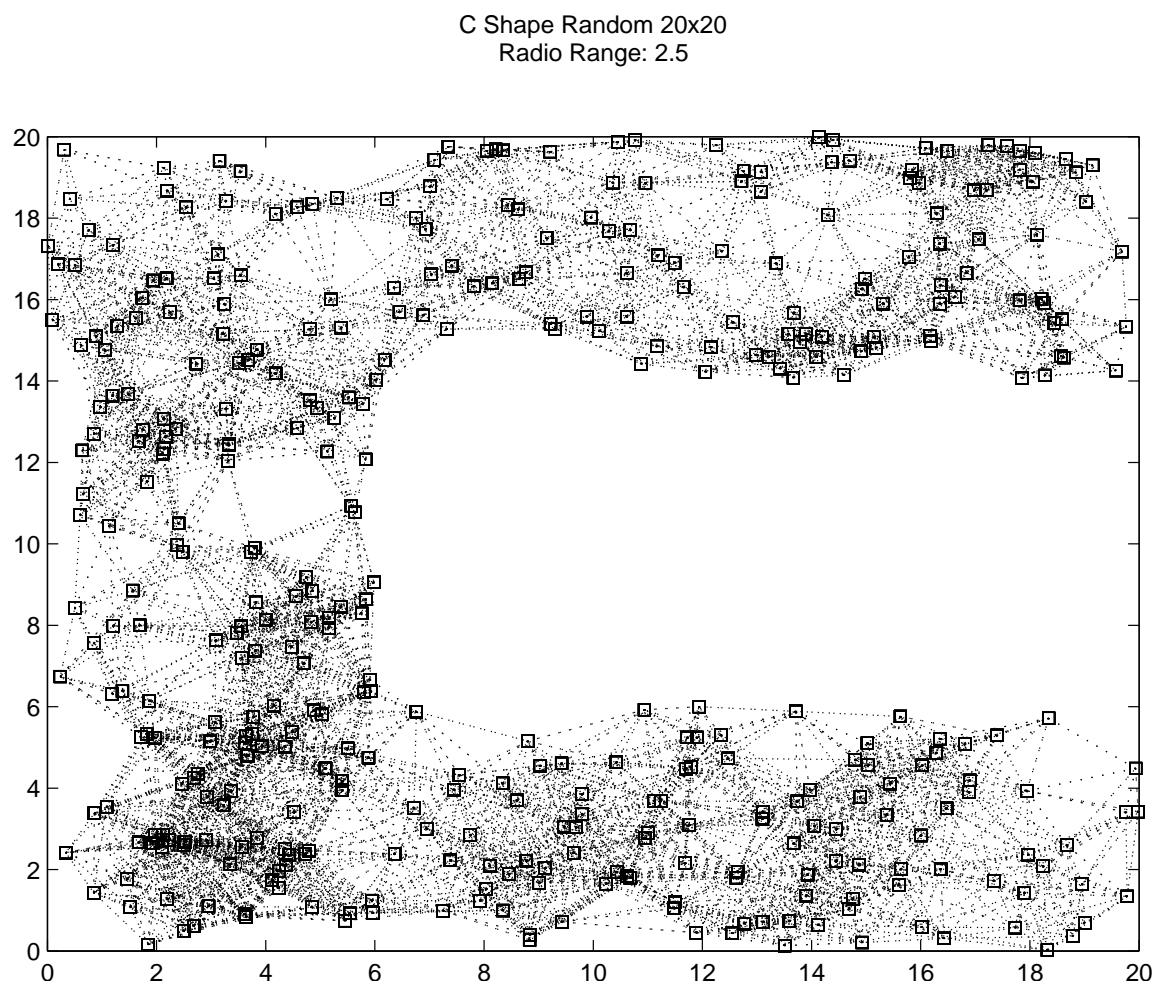


Figure 6.2: Example of C-shape network topology

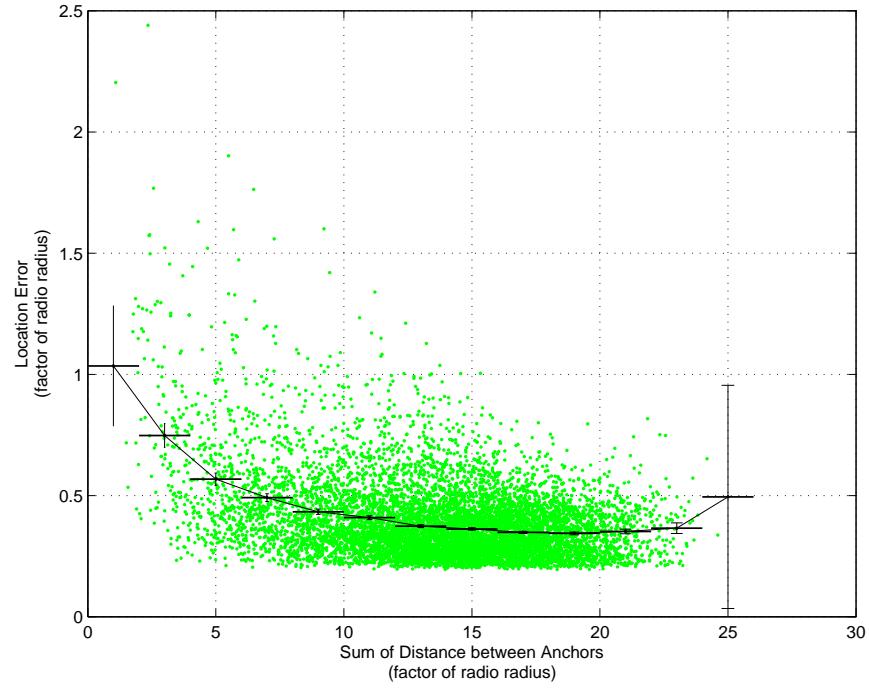
holds for C-shape topology as it did for square in Section 5.2. Figure 6.3b shows that as long as the minimum height of the triangle formed by the anchors nodes is greater than one times the radio range, then the outlier case can be avoided, as is the case for a square network.

6.2.2 Pipeline Network Topology

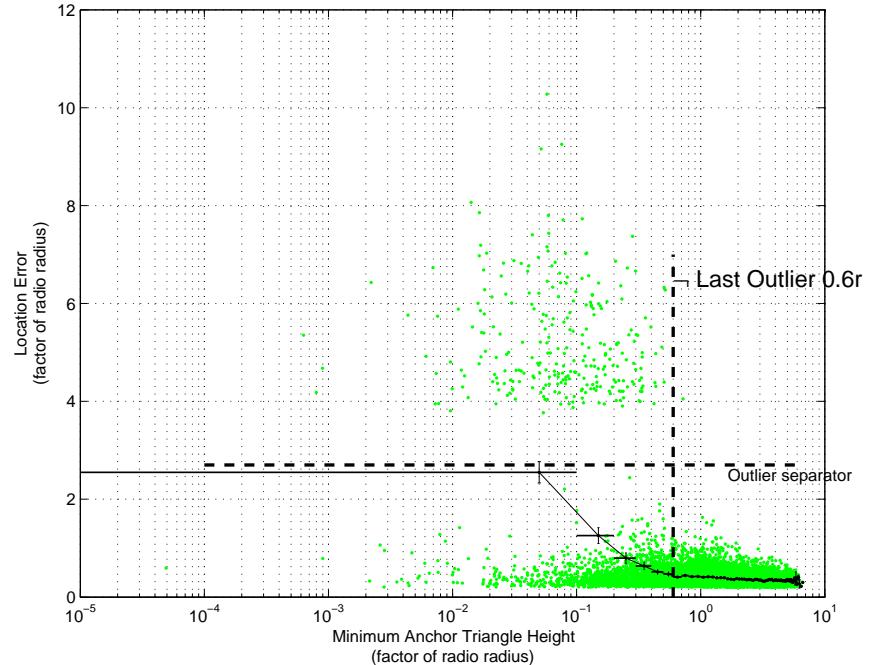
In some applications, that network region has very little depth to it, such as gas pipeline or along a highway or railroad line. The extreme case is where there is a single node placed along a straight axis. As we have shown, this is the worst possible scenario, as it is the most likely was to cause the outlier condition for localization. In that case, it is worthwhile to explore the possibility of other localization techniques, such as GPS at each node, or recording the location as the nodes are placed.

However, for randomly nodes, when there is a bit of depth to the network, as shown in Figure 6.4 (unequal x and y scales), there is still the possibility of good localization results. For pipeline networks with randomly placed nodes, it is much more difficult to meet the network connectivity requirements. Therefore, the simulations are run for just six networks with 400 nodes in a 200 by 2 unit area. Note that the width of the pipeline, 2 units, is less than the radio range, 2.5 units. Also recall that the recommendation to avoid outliers is that the minimum height of the triangle must be at least equal to the radio range, which even the width of the network does not meet.

Even worse than expected, all of the 30,000 anchor sets across the six networks are outlier cases. Recall that in a square network, some anchor sets that did not meet the minimum height recommendation are still able to produce usable localization results. However, here none of the anchors do. This tells us that something about the two-dimensional nature of the square area is able to sometimes compensate for the collinearity whereas in a pipeline, the one-dimensional nature is not able



(a) Sum of distance between anchors vs. mean location error, excluding outliers



(b) Minimum height of anchor triangle vs. mean location error

Figure 6.3: Anchor metrics for 10 random C-shape 20-by-20 unit networks with 400 nodes, 5,000 anchor sets each, and a radio range of 2.5 units, excluding outliers

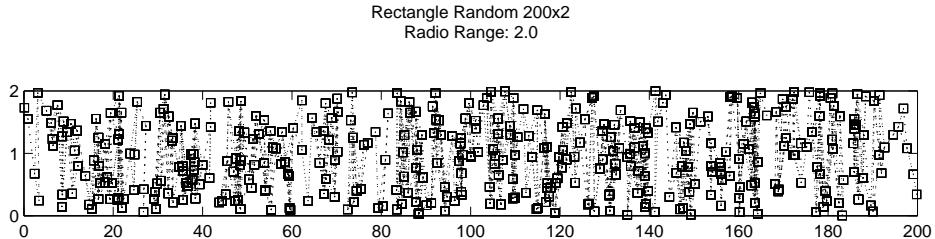


Figure 6.4: Example of pipeline network topology

to compensate.

Figure 6.5 shows that the mean location error is about 20 times the radio range which is well beyond any reasonable value. Interestingly, a relatively small number of anchor sets cause the location error to increase even further when the sum of the distance between anchors is roughly equal to the length of the pipeline. However, the mean of the mean location error actually dips, although only slightly, while the maximum errors peak. The peak in maximum mean location errors may be explained by noticing that in order to have a sum of distance between anchors equal to the length of the pipeline, there are two basic scenarios. First, the three anchor nodes are spread across the entire distance of the network. This scenario likely falls in the normal range of errors. The second scenario is when two anchors are roughly co-located, and the third anchor is half the length of the network away. We suspect that this case causes the worst localization results since effectively this is equivalent to having just two anchor nodes.

Surprisingly, when the same simulation is run on a 200 by 4 unit area pipeline network where the radio range is 2.5 units, all the anchor sets still provide extremely high location errors. Figure 6.6 shows the sum of the distance between anchor nodes versus the mean location error of the network. The minimum location error in this case is roughly five times the radio range. When comparing with the pipeline network where the width of the network is less than the radio range, two different patterns

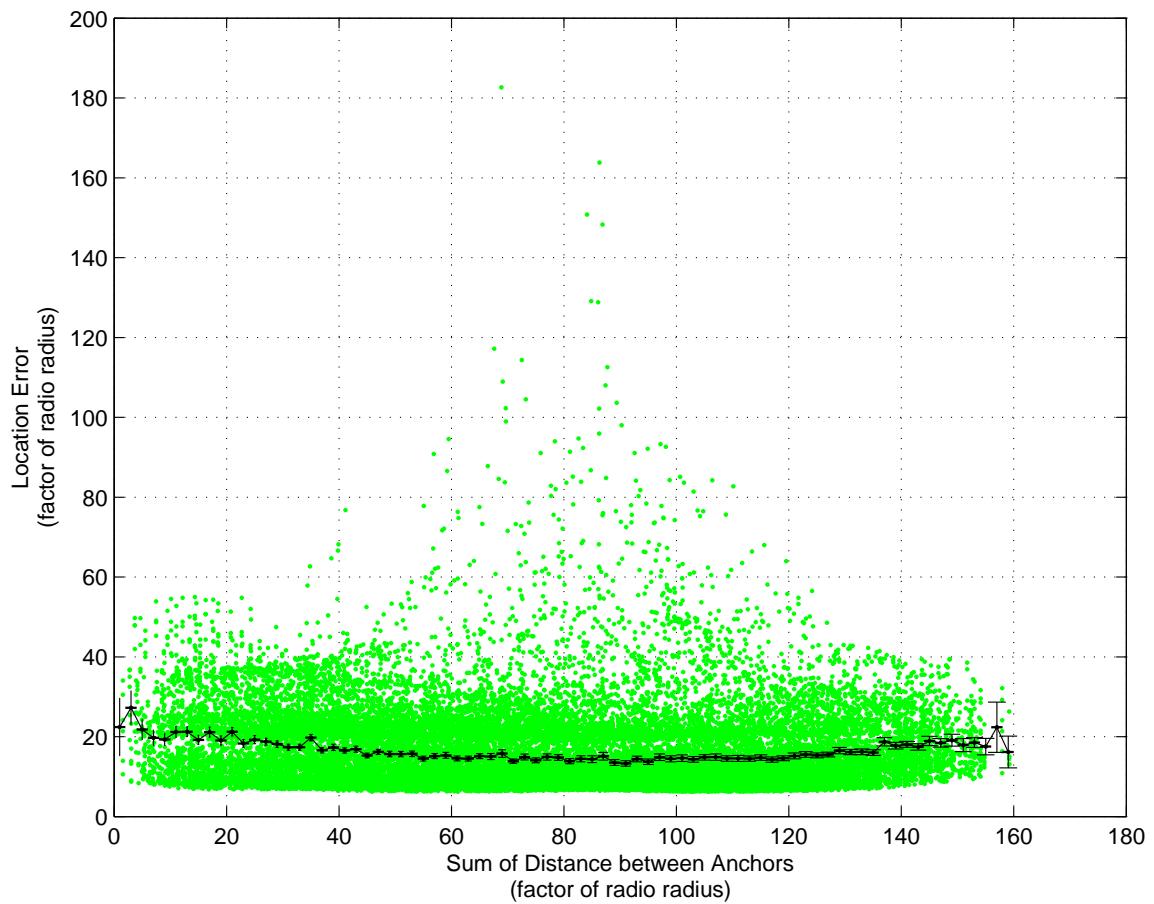


Figure 6.5: Sum of distances between anchor nodes vs. location error for 6 random pipeline 200-by-2 unit networks with 400 nodes, 5,000 anchor sets each, and a radio range of 2.5 units

emerge. The overall data distribution has an inverse relationship between the sum of the distance between anchors and location error. However, there is a similar relationship when comparing to the mean of each interval of the data.

The mean location error is lowest when the sum of the distance between anchor nodes is roughly one length of the pipeline. This can be explained by inferring that when the sum of the distance between anchors is low, the anchors must be clumped together in the pipeline. In this scenario, only a small portion of the pipeline is covered and therefore leads to poor localization performances. As the sum of the distance between anchors approaches twice the length of the pipeline, two anchors must be at one end of the pipeline and the third anchor at the other end. Since two of the anchors are therefore more or less co-located, the algorithm again behaves as if there are only two anchor nodes, causing poor localization performance.

In the end, even though the results are a bit better when the width of the pipeline is more than the radio range, the localization results are poor using this method for pipeline network topologies. Therefore, it is recommended to use a different class of localization algorithms that does not rely on transforming local coordinates into global coordinates with anchor nodes.

[TODO Is there an overall conclusion then? How comes your results are so poor when the published results for CCA in the case of a pipeline network are much better? When does a pipeline network become "square enough" to allow using CCA-MAP or MDS-MAP?]

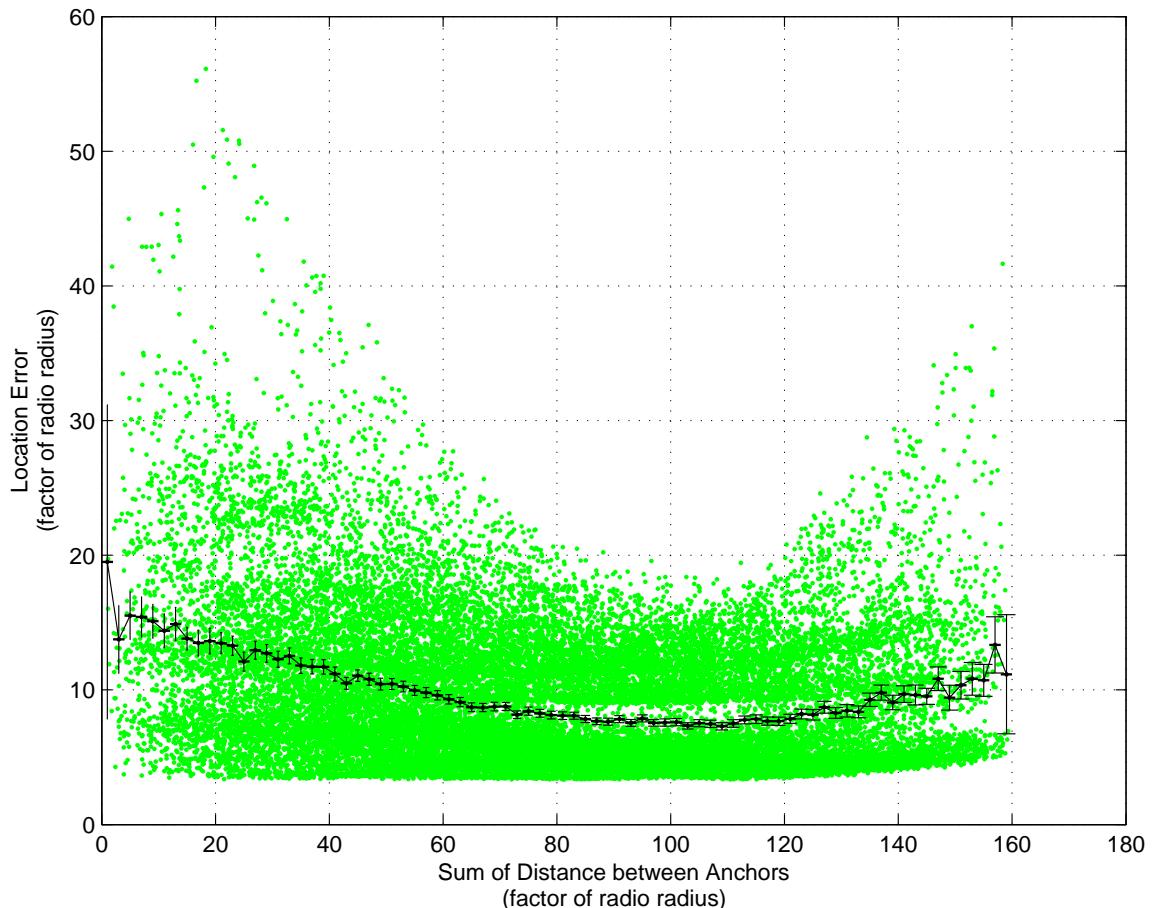


Figure 6.6: Sum of distances between anchor nodes vs. location error for 6 random pipeline 200-by-4 unit networks with 800 nodes, 5,000 anchor sets each, and a radio range of 2.5 units

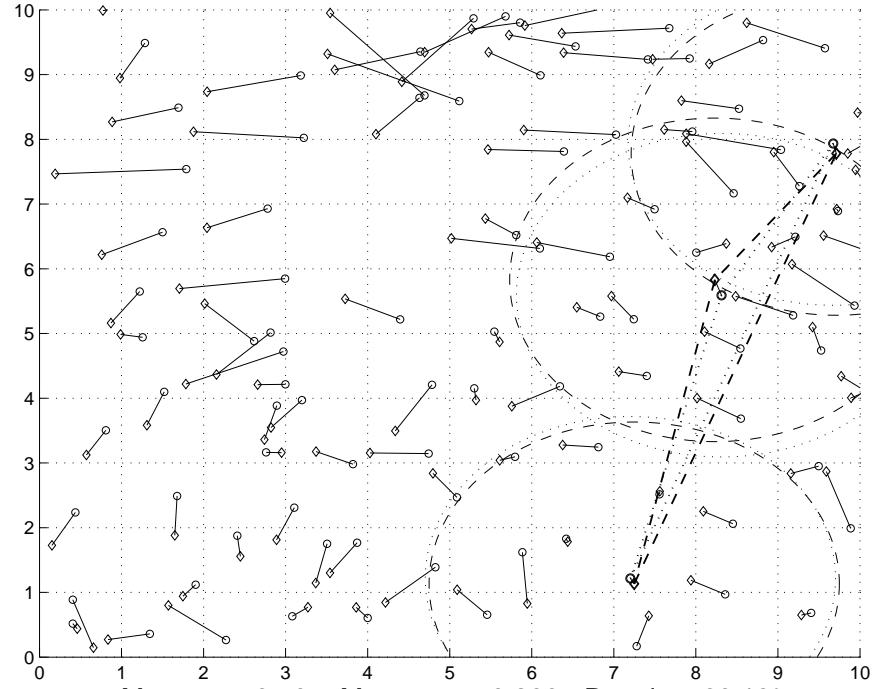
Chapter 7

Other Factors and Future Work

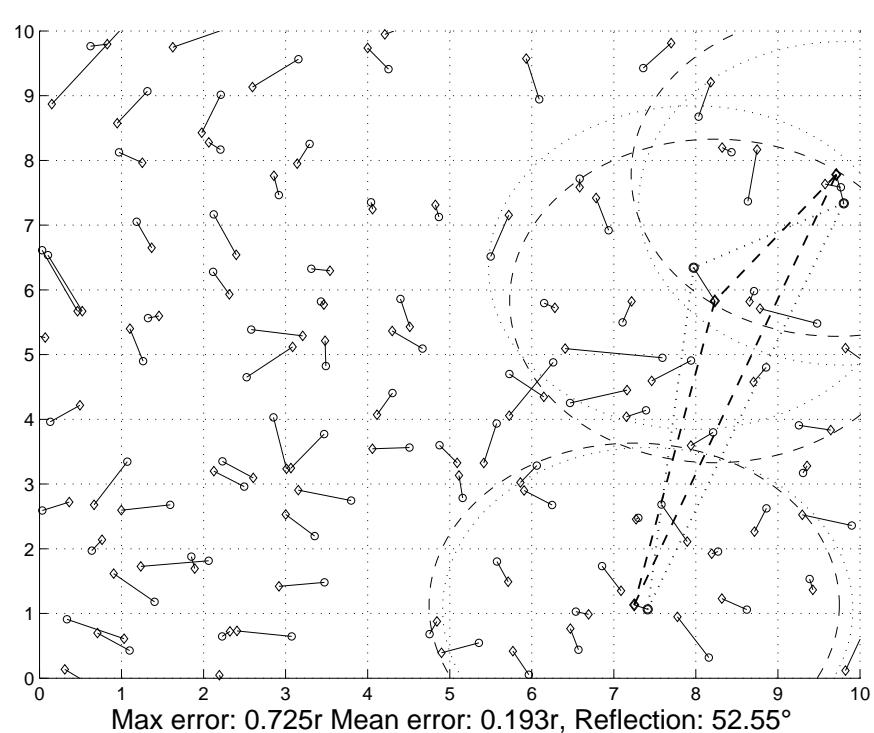
So far, we have examined anchor placement as the sole source of localization error. We set out here to show that there are indeed other factors that play a significant role in determining localization performance.

In the following example, the same anchor set is used in four different random networks. Figure 7.1 shows two non-outlier cases. The mean error for each network, despite using the same anchor set, is different. What is surprising is the degree to which they are different: 37%. Figure 7.2 shows two more networks with the same anchor set. This time, both of the plots reveal outlier cases. Despite using the same anchor set, the localization performance differs significantly. While we have shown that anchor placement does play a role in the localization performance, these simple examples clearly show that there are significant other factors affecting the localization error.

These results show that there is significant variation in localization results across networks, when the anchor set is geographically fixed. In other words, the absolute anchor positions are not enough to predict localization errors. Therefore, it is clear there are other factors besides anchor placement effecting the resulting location errors. A key piece of future work is to isolate and analyze these other factors. However,

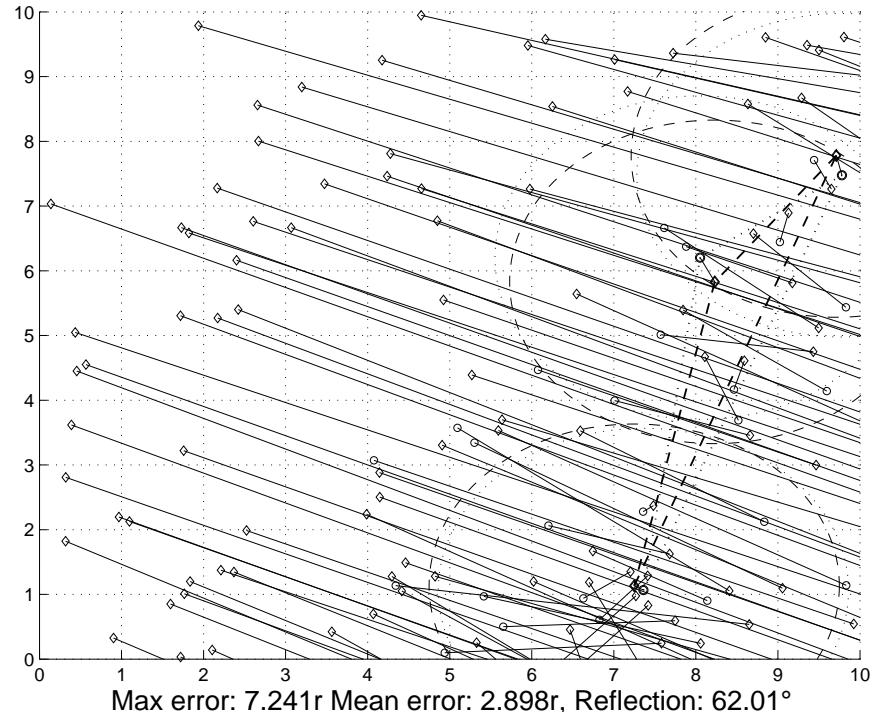


(a) Network A

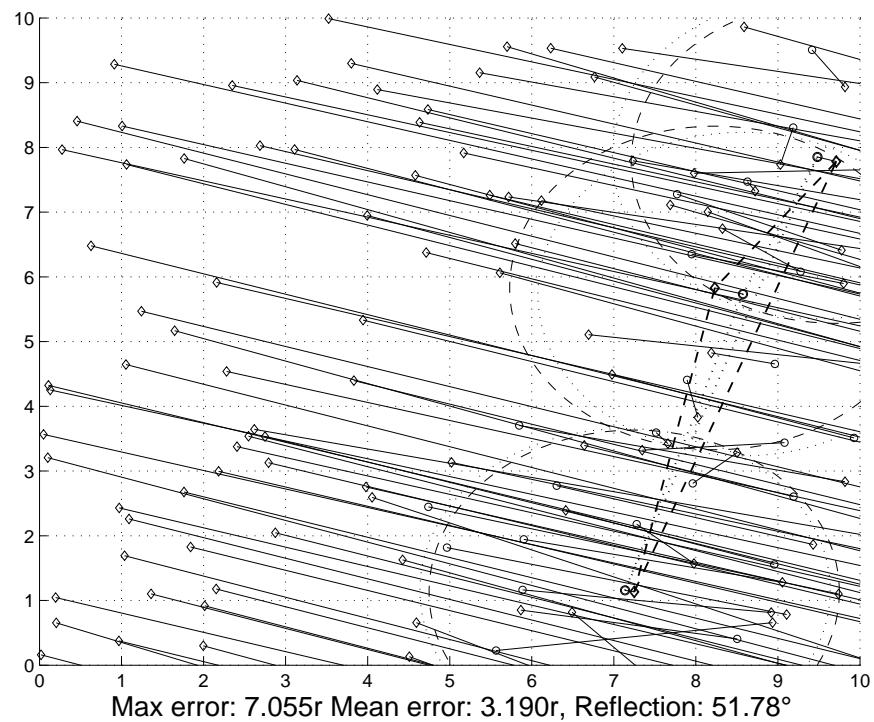


(b) Network B

Figure 7.1: Good localization performance with same anchor set in two different networks



(a) Network C



(b) Network D

Figure 7.2: Poor localization performance with same anchor set in two different networks

since the outlier case itself is undetectable in the absence of extensive ground truth data, the best we can hope for is a way to minimize the probability beyond analyzing the height of the anchor node triangle by studying what these other factors might be.

Some of these factors have been previously explored. Network connectivity level is highlighted in [?]. Connectivity level is a factor of both node density in combination with the radio range. During this work, we discovered the node chosen to start the local map patching also has an effect on the end result, although not always significant.

Chapter 8

Conclusion

8.1 Future Work

While we have provided some recommendations for anchor placement, there are a few key areas where future study could enhance the results. First, an examination of the other factors affecting localization performance would be beneficial. Specifically, further analysis of the results presented here, as they are affected by network connectivity levels. Second, expanding these results to three-dimensions may benefit some network designers. For example, a sensor network may be deployed through a high-rise building or on a bridge. While we expect that results to be similar, it is worthwhile to confirm. Outside the scope of sensor networks and the physical world, these results could also apply to any dimension of data. This would effectively be an exhaustive study of Procrustes analysis where a set of data in one coordinate system is transformed into another coordinate system based on a subset of known points.

8.2 Summary

While the simulations in this study use the CCA-MAP algorithm, the results apply to any protocol in which a local coordinate system is transformed into a global coordinate system using a set of anchor nodes, like MDS-MAP. Further, the scope of the study extends beyond wireless sensor network protocols and can be applied to any transformation problem where a small subset of points is used as the basis to transform a set of points.

This study provides two key recommendations for network designers when placing anchor nodes. Namely, make sure that the sum of the distance between anchor nodes is at least ten times the radio range and that the minimum height of the triangle formed by the anchor nodes is at least equal to the radio range. Further, the larger these two metrics are, the lower the mean location error of the network will be and the lower the probability of using an anchor set that will cause extremely poor localization performance. We have further shown that these criteria apply to network topologies where the overall network area is two-dimensional, but fails when the network topology is extremely narrow compared with the radio range, as in pipelines or roads.

List of References

- [1] D. Niculescu and B. Nath, “Ad Hoc Positioning System (APS),” in *Global Telecommunications Conference (GLOBECOM 2001)*, vol. 5, (San Antonio, Texas), pp. 2926–2931, November 2001.
- [2] Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz, “Localization from connectivity in sensor networks,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 15, pp. 961 – 974, nov. 2004.
- [3] L. Li and T. Kunz, “Localization applying an efficient neural network mapping,” in *Proceedings of the 1st international conference on Autonomic computing and communication systems (Autonomics 2007)*, vol. 302, (Rome, Italy), 2007.
- [4] L. Li and T. Kunz, “Cooperative node localization using nonlinear data projection,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, February 2009.
- [5] J. Kumagai, “The secret life of birds,” *IEEE Spectrum*, vol. 41, pp. 42–48, April 2004.
- [6] “RUNES (Reconfigurable Ubiquitous Networked Embedded Systems) IST Project.” <http://www.ist-runes.org>, 2004-2006.
- [7] B. Wellenhoff, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*. Berlin, Germany: Springer, 4th ed., 1997.
- [8] A. Savvides, C. Han, and M. Strivastava, “Dynamic finegrained localization in ad-hoc networks of sensors,” in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom’01)*, (Rome, Italy), pp. 166–169, July 2001.
- [9] D. Niculescu and B. Nath, “Ad hoc positioning system (APS) using AOA,” in

Twenty-Second Annual Joint Conference of the IEEE Computer and Communications (INFOCOM 2003), vol. 3, (San Francisco, California), pp. 1734–1743, April 2003.

- [10] N. Patwari, A. Hero, M. Perkins, N. Correal, and R. O’Dea, “Relative location estimation in wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, 2003.
- [11] Y. Shang and W. Ruml, “Improved mds-based localization,” in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, pp. 2640 – 2651 vol.4, 7-11 2004.
- [12] P. Demartines and J. Herault, “Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets,” *Neural Networks, IEEE Transactions on*, vol. 8, pp. 148 –154, jan 1997.
- [13] L. Doherty, K. Pister, and L. El Ghaoui, “Convex position estimation in wireless sensor networks,” in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1655 –1663 vol.3, 2001.
- [14] S. Hara and T. Fukumura, “Determination of the placement of anchor nodes satisfying a required localization accuracy,” in *Wireless Communication Systems. 2008. ISWCS ’08. IEEE International Symposium on*, pp. 128 –132, 21-24 2008.
- [15] J. Ash and R. Moses, “On optimal anchor node placement in sensor localization by optimization of subspace principal angles,” in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 2289 –2292, March 2008.
- [16] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, Ltd, 2005.
- [17] C. Savarese, J. M. Rabaey, and K. Langendoen, “Robust positioning algorithms for distributed ad-hoc wireless sensor networks,” in *ATEC ’02: Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference*, (Berkeley, CA, USA), pp. 317–327, USENIX Association, 2002.
- [18] K.-Y. Cheng, K.-S. Lui, and V. Tam, “Hybloc: Localization in sensor networks with adverse anchor placement,” *Sensors*, vol. 9, no. 1, pp. 253–280, 2009.

- [19] H. Lim and J. Hou, “Localization for anisotropic sensor networks,” vol. 1, pp. 138 – 149 vol. 1, mar. 2005.
- [20] R. Zemek, M. Takashima, S. Hara, K. Yanagihara, K. Fukui, S. Fukunaga, and K. Kitayama, “An effect of anchor nodes placement on a target location estimation performance,” in *TENCON 2006. 2006 IEEE Region 10 Conference*, pp. 1 –4, 14-17 2006.
- [21] “Procrustes analysis.” <http://www.mathworks.com/help/toolbox/stats/procrustes.html>, 2010.

Appendix A

Matlab©Simulation Code

All source code used to simulate CCA-MAP and generate resulting data can be found in Google©Code. The code was originally written by Li Li and modified for the purposes of readability and different output data and plots.

The primary files are:

- ccaconfig.m: Property file to setup the network shape, number of anchor sets, anchor placement, radio range and other simulation parameters.
- simcca.m: Generate and simulate a network over the configured number of anchor sets and output a suite of statistical plots.
- multisimcca.m: Generate and simulate a configured number of networks, with the same exact anchor sets and output a suite of statistical plots.
- indicator.m: Outputs a suite of statistical plots for a set of previously run output directories from simcca.m. This allows combining multiple networks and their anchor sets into a single analysis.

See <http://code.google.com/p/sim4j/source/browse/#svn/trunk/thesis/matlab> for details.