Relative Accuracy based Location Estimation in Wireless Ad Hoc Sensor Networks

May Wong Demet Aksoy

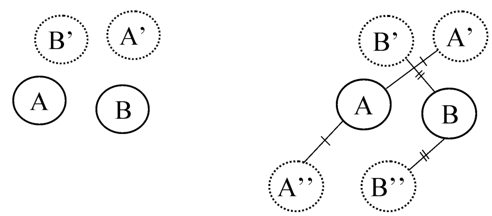
Intel, Inc. University of California, Davis

***Abstract*—In sensor networks, the location of a sensor making an observation is a vital piece of information to allow accurate data analysis. GPS is an established technology to enable precise position information. Yet, resource constraints and size issues prohibit its use in small sensor nodes that are designed to be cost efficient. Instead, most positions are estimated by a number of algorithms. To date, however, the focus was on individual accuracy of each sensor's position estimates in isolation to the complete network. In this paper, we propose a novel positioning algorithm called nQUAD to enable representative topology information based on relative accuracy. nQUAD makes use of relative distances to determine the quadrant it resides in and refines its estimation according to neighbor provided information and reports a level of certainity along with estimates. Our experiment results suggest significant improvements in individual accuracy in comparison to lateration based alternatives. At top of all, drastic improvements are achieved in the overall topology to enable accurate analysis of data.**

I. INTRODUCTION

Wireless sensor nodes are used in a wide range of applications such as scientific research, military, healthcare, and environmental monitoring. Sensor nodes collect information about the environment and communicate their observations to a data collection point from where users can access the collected data without the need to travel to the monitored area. In this regard, every user has to depend on the location information provided by the sensor node that reports an observation. As a result the user's view of the monitored area highly depends on reported locations, and, therefore, it is critical to illustrate a representative picture of observations to users.

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



(a) position estimates, A' and B', for the sensor nodes A and B

(b) comparison with an alternative set of estimates

Figure 1: Representative topology is more important than reducing the individual errors reported in isolation to the network: A and B are the actual positions. A' and B' are the estimates of one localization algorithm. A'' and B'' are the estimates of another algorithm that results in a similar pair-wise error. However, the estimates, A'' and B'', result in a completely misleading overall topology from the end users point of view.

Obviously, errors are inevitable in estimations, and, it is important to understand the impact of errors for a particular application. In Figure 1, we illustrate a simple example with only two sensor nodes. The actual positions of A and B are represented by solid circles in the figure. Recall that, in practical applications, the actual positions of these nodes would not be known, and one will have to depend on the position estimations reported by the nodes. In figure 1(a), we plot localization estimates being reported by these nodes as A' and B'. Now, consider another localization algorithm that produces location estimates of A'' and B'' for the same nodes as demonstrated in figure 1(b). Following the traditional approach in localization studies, we would evaluate these two sets of estimations based on the Euclidean distance between the real and the estimated positions of individual nodes. When considered in isolation as in previous work, this would suggest a similar error in both cases. However, these two sets of estimates have quite different impacts for data management in practical applications! In particular, the relative positions of A' and B' are incorrect in comparison to

This work is supported in part by a research grant from CENIC.

A and B. In consequence, this may result in misleading conclusions during data analysis. For instance, the advection of a particulate pollutant may appear to be in the *reverse* direction than it really is.

In general, the *precise* location of each sensor node is not necessary in most sensor network applications [AKY02]. Yet, accurate overall topologies are vital for accurate identification*, routing, in-network processing as well as overall analysis of observations. Our focus, therefore, is on the overall sensor network topology constructed, rather than on individual estimates as has been the major focus in previous studies, e.g., [NIC04, SAV01, LAZ04, MOO04, NAG03].

Cricket [PRI00], Centroid [BUL00], SeRLoc [LAZ04], Active Badge [WAN92], etc. require a high number of landmarks. Landmarks are super nodes with additional resources, e.g., GPS antennas, additional power and resources to allow precise knowledge of their location. In practice, it is desirable to reduce the number of landmarks or additional hardware at ordinary nodes, as required by DV-Position [NIC04] and AHLoS [SAV01]. While APIT [HET03] requires nodes to move and have accurate signal strength measurements to produce estimates. [WHI06] discusses problems with signal strength based approaches. [NAG03][LAZ04][MOO04] require highly dense networks with lots of neighbor nodes to produce reasonable estimates.

In this paper, we propose a novel location estimation algorithm, QUAD (Quadrant-based estimation), that aims at calibrating estimates based on relative position information to help assist accurate data management. QUAD assigns levels of certainty to generated estimations since it is of extreme importance to inform the user about the level of uncertainty in location estimates to reflect possibility of errors.

In particular, the relative distances to known landmark positions are evaluated to determine if the node resides East/West/South/North of the landmarks. Once the quadrants are determined, estimates are then generated using neighbor observations. Unlike approaches that require additional hardware (e.g., [NIC04, SAV01]) to allow such directional information, we merely rely on radio communications for location estimations.

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

Our Contributions:

In this study we point out the major disadvantage in existing location estimation algorithms: treating each individual node estimate independently in isolation to the complete network. We propose a novel localization algorithm, QUAD, to address this problem. We focus on the following challenges:

- ***Representative Overall Topology:*** Based on expert user input, the relative positions of the nodes in the network constitute a critical piece of information that should be maintained for accurate data management and accuracy. QUAD enables such representative overall topologies. In addition, each reported location estimate includes a certainty level to help reflect possible errors.
- ***Minimal Specialized Hardware:*** In order to keep the overall network economically feasible additional hardware (e.g., GPS, directional antennas) should be avoided whenever possible. Keeping the individual node size small is another goal. Therefore it is preferred to use existing capabilities (e.g., radio communication), without further investment on additional equipment. In this regard, QUAD relies merely on regular radio communications for making location estimates.
- ***Robustness to Network Density:*** The location estimation should not be dependent on a specific network density, as parts of the network can be sparse while others are dense. QUAD is robust to changes in network densities.

Our performance evaluation results suggest significant improvements in comparison to previous algorithms. At top of all, our proposed algorithm can provide representative topologies for various network settings and densities.

The rest of this paper is organized as follows: In section II, we present our motivation behind QUAD. Section III describes our proposed localization algorithm. Section IV presents results from our performance evaluation using various network topologies. Finally, Section V concludes this paper.

II. Relative Position Accuracy

Our main motivation in this study is the expert user, e.g., environmental engineers', demand for maintaining a representative overall topology with estimated sensor locations. Representative overall topology would not only enable accurate data analysis, but also correct information source selection [AKS05b]. In the following subsections we outline our motivations behind the design of QUAD.

A. *Problems with Lateration*

For basic wireless communications without additional hardware, when a node receives a transmission, it can *at best* estimate the distance from the sender. The sender can be anywhere on a radius of this distance around the receiver. A major technique applied in localization is lateration. Lateration [NIC01][NAG03] refers to finding the intersection point of (at least three) circles around the references. If accurate distances are provided, the solution can be obtained by a set of equations in the form

$$(x_1 - x)^2 + (y_1 - y)^2 = d_1{}^2$$
$$(x_2 - x)^2 + (y_2 - y)^2 = d_2{}^2$$
$$(x_3 - x)^2 + (y_3 - y)^2 = d_3{}^2$$

where (x_i, y_i) are the coordinates of landmark i and d_i is the estimated distance from the landmark. In practice, however, distance d_i can be misleading since the multi-hop distance to a landmark is determined as the sum of the distances along the path between the landmark and the node. Therefore, even if the individual distances between immediate neighbors could be precisely determined, the sum is only an upper bound on the actual distance unless nodes are aligned on a perfect line [KAM06].

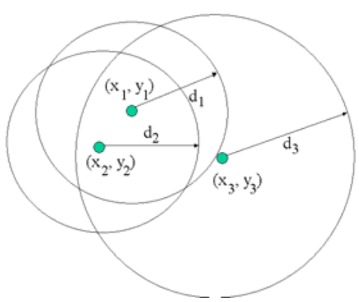


Figure 2: Lateration can only provide a solution if the distances are non-conlicting. In practice, the distances d_1, d_2, and d_3 are only approximated, and as a result the three circles might not even intersect at a common point.

We demonstrate an illustrative example with a small error in one distance estimate from a landmark in figure 2. In the figure there are three landmarks at locations (x_1, y_1), (x_2, y_2), and (x_3, y_3). A sensor node is estimated to be d_1, d_2, and d_3 apart from these landmarks, respectively. If these distances were precise, we could successfully estimate the node's position. Otherwise, as illustrated by the example, we can end up with conflicting circles around the landmarks that do not necessarily intersect. As demonstrated, it is possible that the error made in distance estimates d_1, d_2, and d_3 can prevent obtaining an estimate for the node's location.

B. *Implications for Data Management*

With the popularity of sensor networks we observe the number of sensors increasing at a significant rate with an increasing demand on data management. For instance, *PLASMA (PLAnetary Scale Monitoring Architecture)* [AKS05a] is an interdisciplinary project that aims at providing an integrated architecture for heterogeneous sensor systems to enable public access for user queries. PLASMA consists of a large number of heterogeneous sensor nodes that report to data collection points. We observed that previous localization algorithms that rely on lateration can yield quite unrealistic views of the actual deployment. For instance, even though some node locations could be estimated with 100% accuracy, strikingly conflicting estimates are made for some other nodes' location even for immediate neighbors. As demonstrated in figures 6-7 the overall topology can look significantly different than what one would expect with the knowledge of the actual topology.

At top of all, for large-scale deployments, it is of extreme importance to inform the user about the level of uncertainty in location estimates to reflect the possible errors being made using the estimates. This is especially important in cases where the estimates are produced with a low confidence level. For the user interface, it is highly desirable to provide some information about the certainty level in location estimations while reporting observations from the field. Based on our observations, we were motivated to develop an alternative approach for localization.

III. QUAD LOCALIZATION ALGORITHM

Our proposed algorithm, QUAD relies on an intuitive comparison between the relative distances to known landmark locations. This comparison allows nodes to figure out the quadrant, i.e., North/South/East, etc. in reference to the landmarks. QUAD can be summarized in three phases as explained below.

1) hop distance dissemination: This first phase is the same in any algorithm that uses hop distance based approach, e.g., DV-Hop [NIC01], Smooth [NAG03]. In particular, each landmark will flood the network with its location. Each node records the minimum hop count to each landmark among the received messages. The hop counts are then converted to actual distances according to the radio range.

2) position vote: In this phase, each node compares its distance to each landmark. If the node is equidistance to all landmarks, it delays its decision until the next phase. Otherwise, it determines its relative position in comparison to the landmarks. Note that we deploy landmarks in clusters, typically of three[*] landmarks. Such a deployment can be easily achieved by tying three landmark nodes with ropes/rods or installing them on a common board.

Each node first categorizes each landmark as the *near*, and *far* landmark nodes based on their relative hop distance. First, the minimum and the maximum hop distances are used to identify the furthest and the nearest landmarks. Then, if a landmark's hop count is closer to the nearest, it is categorized as near; and if it is closer to the furthest, it is categorized as far.[†] For instance, if a node is 1,3, and 4 hops away from the landmark nodes L_1, L_2, and L_3 respectively, then L_1 is put in the near set, and L_3 and L_2 are recorded in the far set. Nodes that can not be categorized in either set are left out.

Then each member in the near set is compared to each member in the far set to produce a negative/positive/middle vote on each x and y-coordinate as plotted in figure 3. The main idea of this categorization is to determine the relative location a node is residing in. For instance, if the node is closer to a landmark that has a smaller x-coordinate, it is concluded that it resides somewhere *East* of the landmarks.

```
vote = None;
for (int i=0; i<# elements in near set; i++)
  for (int j=0; j<# elements in far set;j++) {
    if (landmark[near[i]].X > landmark[far[j]].X) {
      if (vote== East)
        vote = Middle;
      else
        vote = West;
    }
    if (landmark[near[i]].X < landmark[far[j]].X) {
      if (vote== West)
        vote = Middle;
      else
        vote = East;
    }
  }
```

Figure 3: Pseudo code for determining the quadrant that will help estimate the X-coordinate of a sensor node in comparison to the landmarks categorized in the near and far sets.

We apply the simple decision criteria as demonstrated in figure 3 also to Y-coordinates where each occurrence of East is replaced by South and each occurrence of West by North. The combination of X and Y-coordinate votes determine the quadrant a sensor node resides in. For instance, North and East votes suggest the Northeast quadrant in reference to the landmark cluster. These votes help produce estimations as explained in phase 3).

Figure 4 plots an example topology with a cluster of three landmarks at (50,50), (49,49) and (50,48), respectively. The landmark nodes are represented by dark circles, and the ordinary nodes are represented with light colored circles. In this example the radio range is 1 unit such that each node can communicate with direct east, west, north, or south neighbors. Note that the perfect grid structure helps demonstration of our approach. In practice, additional errors are due to irregularities of deployment in the field.

In the example, node A categorizes the landmark at (50,50) as the near and those at (49,49) and (50,48) as the far landmark nodes. The X-coordinate of the nearest landmark (50) is larger than or equal to each of the nearest landmarks (located at 49 and 50). Therefore node A will have a *West* vote for the X-coordinate and a *North* vote for the Y-coordinate. A is 3 hops away from (50,50) so, intuitively, it can not have an X-coordinate greater than 53 or a y-coordinate greater than 53. At this point, node A can be concluded to be somewhere in the approximated *NorthWest* region shown in the northeast quadrant.

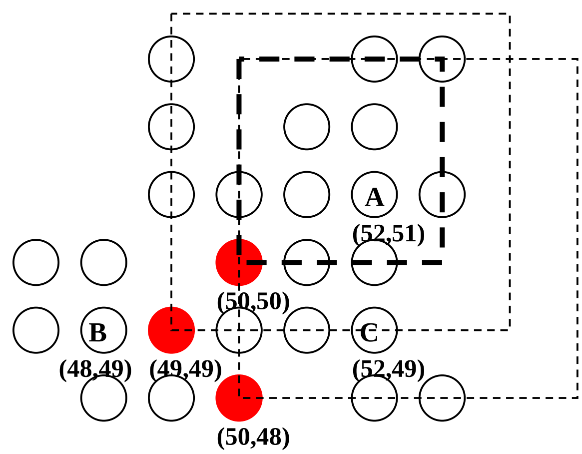


Figure 4: Example topology with a radio range of 1 unit. Node A is 3 hops away from the landmark node at (50, 50), and the marked region plots possible estimations for node A.

3) location estimation: After the nodes determine their position votes, they announce these votes (if any) to their immediate neighbors. The estimation process is as follows. If a node has produced a *Middle* vote for a particular coordinate, it is estimated to be in between those coordinates of the landmarks. For

[*] In theory two landmarks would be sufficient for a cluster with an appropriate alignment. However, due to irregularities in the field we require the additional landmark to ensure comparable X and Y coordinates.

[†] The main objective is to single out landmarks that will help in estimating the relative location.

instance, in the previous example node B would have a *Middle* vote for the Y-coordinate, its coordinate is estimated to be the midpoint of the highest and the lowest Y-coordinates of the landmarks, namely 49. Node B is 1 hop away from the landmark at (49,49). The distances in the X and Y direction should add up to 1. As a result node B will produce a coordinate estimate of (48,49).

All nodes apply similar approximations to determine the lower and upper bounds of possible locations. Nodes that hear conflicting votes on a particular coordinate adjust their estimates accordingly. For instance, in the previous example node C will hear from both *South* and *North* neighbors which indicates that it is close to the borderline between south and north for the y-coordinate. This information is exploited to determine the other coordinate similar to the above case.

For pure quadrant votes, we have a wider range of possibilities based on the size of the approximated box. Therefore, the certainty level of the estimate is set to be inversely proportional to the distance from the landmarks. As a result, nodes closer to the landmarks have higher certainty levels in comparison to those further away. For boundary cases, e.g., *Middle* vote or conflicting votes, the certainty level is set higher since the range of possibilities is smaller. Such higher certainty levels help neighbors improve their estimations.

At the end of this phase nodes adjust their own estimates using weighted average of neighbor estimates with equal or higher levels of certainty. Note that it is possible to continue refining the estimates to improve estimates further.

IV. PERFORMANCE EVALUATION

To evaluate the performance of QUAD, we have implemented a simulator using C++. We compared QUAD with well-known localization algorithms, DV-Hop, Min/Max , and Smooth for a wide range of scenarios. In each setting we feed the simulator with an arbitrary topology and obtain the position estimates of each sensor node based on the localization algorithm.

DV-Hop is a practical implementation of DV-Distance [NIC01] that addresses the problems raised in [KAM06]. For DV-Hop implementation, we record the minimum hop distances between the landmarks and the nodes. The distances are then improved based on topology information before lateration. Min/Max [SAV03] is an approximation to DV-Hop to reduce the complexity of lateration operations. In this case, instead of circles around landmarks approximated squares are assumed. The intersection of these squares help determine the location estimation of each node. Finally, Smooth [NAG03] allows neighbors exchange their distance estimates to improve their estimates before applying lateration. This adjustment highly improves the accuracy of the distance estimates for highly dense deployments.

Error in the estimates is traditionally measured as the Euclidean distance between the real coordinate and the estimated coordinate of a sensor node as

$$\text{Euclidean Error} = \sqrt{(x_e - x_r)^2 + (y_e - y_r)^2}$$

where (x_e,y_e) is the estimated position of a node and its real position is (x_r,y_r). Note that this metric would not differentiate between the two sets of estimates shown in Figure 1. Therefore we also present some visual representation of data to provide better insight.

In our experiments, we used a default radio range of 5 units. We then studied the sensitivity to varying communication ranges. We repeated the experiments with different random number generator seeds such that we end up with different topologies each time. In the experiments, we simplify the communication model assuming no message loss or corruption. In practice the communication model should be enhanced using organized scheduling [BAL07].

A. *Traditional Metric: Estimation Accuracy*

In the first experiment, we exploit three landmarks at (50,50), (49,49) and (50,48) in a 100x100 grid. At 100% density, we deploy a sensor at each integer coordinate in the grid such that all neighbours are equidistance to each other. This highly unrealistic perfect grid topology is used as a comparison basis. We then randomly deleted sensors in this grid. This creates more diverse and representative topologies to reflect problems that can occur in actual deployments. As we use lower density deployments we face unisotropic deployments [LIM05] to stress-test our algorithm. At each setting we apply the algorithm from scratch and evaluate the performance according to the final estimates.

We plot the average Euclidean error of different topologies in Figure 5 as the number of nodes in the network decreases. Recall that the first data point in figure 5 presents an unrealistically dense and uniform topology. At this highly unlikely setting, we have a sensor node at *every* point in the grid. Beyond this unrealistic point, we observe QUAD to provide a significantly better performance in comparison to the other algorithms. As we move to more realistic non-uniform topologies with lower densities, Smooth is out-performed by more and more other algorithms.

This is due to the fact that Smooth depends on the density in the network to improve the hop distance before applying lateration. In low densities, the neighbors are not uniformly distributed around a node, and the targeted improvement is, in fact, detrimental to the performance. When nodes do not have a sufficient number of neighbors Smooth performs worse than DV-Hop.

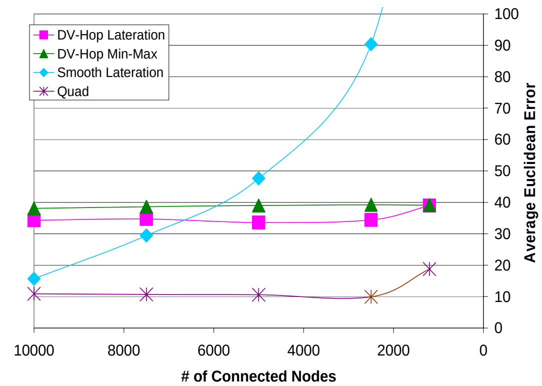


Figure 5: QUAD's individual error performance is the best except for the unrealistic case of a perfect grid.

QUAD, on the other hand, is quite robust to changes in the topology because it relies on relative positioning information. In between the lowest and the highest density settings, for instance, at 50% density DV-Hop has 330% higher error than QUAD.

We were not satisfied with these promising results obtained by the traditional performance metric since our main motivation is to capture a representative overall topology. For this reason, we take a closer look at the estimates in the following.

B. Need for a New Accuracy Metric

In the previous section, we have represented the performance of alternative approaches on a complex overall topology using a simple numeric metric. To provide a better understanding of the performance of the algorithms, we analyzed the estimates made for each node in more detail. For this purpose, a simple illustration of the network with each algorithm's estimation on a 2-dimensional area was not sufficient. In other words, when we plot the location estimation of each sensor node, we loose the information about the relationship to their actual coordinates. For instance, it is not possible to evaluate a case where two estimates at, say, (120,50) and (35,99) are actually immediate neighbors in the deployment topology!

In our study we had a pressing need to plot the estimates as a function of their actual location. For ease of illustration we plot the X and Y-coordinate estimates separately to avoid an unreadable 4-dimensional representation. We have observed a similar trend for both X and Y-coordinates, and therefore, focus on a single coordinate for demonstration purposes.

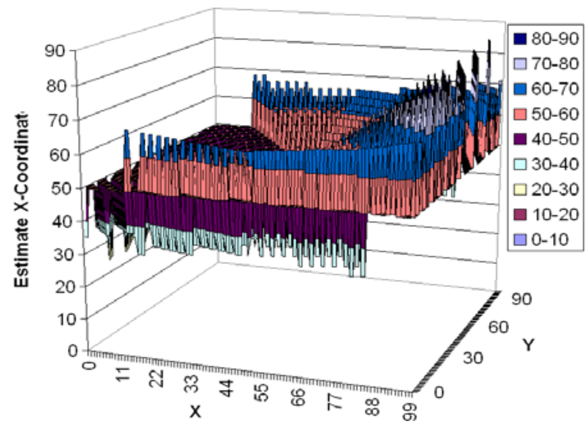


Figure 6: X Coordinate Estimates using DV-Hop: Estimates are all over, either overestimated or underestimated, resulting in highly conflicting estimates even for immediate neighbors.

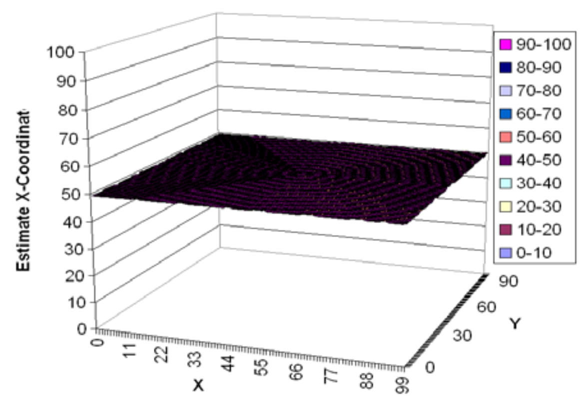


Figure 7: X-Coordinate Estimates using Min-Max: The overall topology is completely misleading.

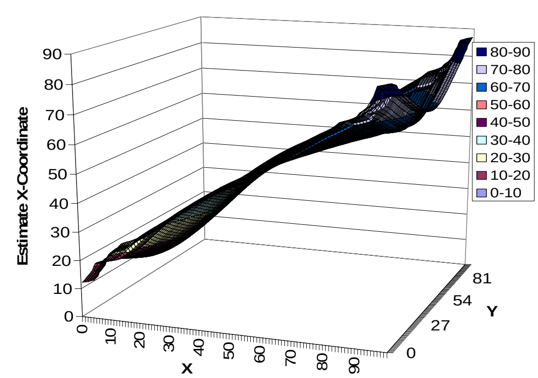


Figure 8: X Coordinate Estimates using QUAD: Refinements based on certainty levels in estimates help improve the overall topology significantly in comparison to alternatives.

Figure 6 depicts the X-coordinate estimates of DV-Hop. we plot the X-coordinate estimate of the node located at (0,0) at position (0,0). In the ideal (no error) case, all nodes would have the exact X-coordinate and the 3-dimensional illustration would appear to be a smooth surface with a 45˚ inclination (similar to figure 8). In figure 6, however, the estimates of DV-Hop are far from being representative as the values are either overestimated or underestimated without a specific pattern. Even immediate neighbors can have highly conflicting estimates in this case. In addition, DV-Hop is likely to produce the same estimates for symmetric locations in a quadrant such that it is not possible to differentiate between the relative positions of sensors.

Figure 7 plots the estimates made by Min-Max. In this setting, Min-Max estimates all nodes to be more or less at the same coordinate, in particular around the average within the whole field. Recall that in Figure 5 Min-Max appears to have a reasonable performance using the traditional metric.

This result suggests that the Euclidean distance metric, as has been used in previous studies, is not a representative metric to evaluate location estimation algorithms. The main problem with this metric is that it does not reflect the relative positioning errors that are very critical for data management.

Using Min-Max it is not possible to differentiate between, say, even the two furthest corner nodes at southeast and northwest, when they report an observation. In figure 8, on the other hand, we plot the X-coordinate estimates generated by QUAD. After refinements applied using neighbour estimates QUAD's topology gets much closer to the actual plane.

V. CONCLUSIONS AND FUTURE WORK

Previous work in localization algorithms focus on individual accuracy of sensor position estimates without considering the relative positioning of nodes in comparison to each other. Yet, such relative positioning defines the overall topology of the sensor network for accurate data management. In this paper, we proposed a novel localization algorithm, QUAD, to produce a representative overall topology based on the relative positions of landmarks to sensor nodes. QUAD only depends on already existing radio communications to produce representative estimates. Our performance evaluation results suggest 2-fold improvements in the traditional Euclidean distance based error metric and orders of magnitude improvement when we consider the overall topology. We are currently developing an alternative accuracy metric that compares the estimated locations of all sensors according to their actual positioning in the field. In terms of future work, we plan to use the certainty levels in estimates for the visualization interface to enable accurate data management for PLASMA queries.

REFERENCES

[AKS05a] Aksoy D., PLASMA: A PLAnetary Scale Monitoring Architecture, Proc. ACM MULTIMEDIA, Singapore, 2005.

[AKS05b] Aksoy D., Information Source Selection for Resource Constrained Environments, ACM SIGMOD Record, *Vol 34, Num 4,* Dec 2005.

[AKY02] Akyildiz I.F., Su W., Sankarasubramaniam Y., Cayirci E. A Survey on Sensor Networks. IEEE Communications Magazine, August 2002, pp.102-114.

[BAL07] Balasubramanian S. and Aksoy D., "Adaptive Energy-Efficient Registration and Online Scheduling for Wireless Sensor Networks," to appear in Elsevier Computer Networks, 2007.

[BUL00] Bulusu N., Heidemann J., Estrin D. (2000) GPS-less Low-Cost Outdoor Localization for Very Small Devices. IEEE Personal Communications, vol. 7, no. 5, Oct 2000.

[HET03] He T., Huang C., Blum B.M., Stankovic J.A., Abdelzaher T. Range-Free Localization Schemes for Large Scale Sensor Networks. Proc. of the 9th Int. Conf. on Mobile Computing and Networking, Sept 2003, pp.81-93.

[LAZ04] Lazos L., Poovendran R. SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks. Proc. of ACM Workshop on Wireless Security, 2004.

[LIM05] Lim H.., Hou J. Localization for Anisotropic Sensor Networks, in Proc. IEEE INFOCOM 2005.

[LIZ05] Li Z., Trappe W. Zhang Y. Nath B. Robust Statistical Methods for Securing Wireless Localization in Sensor Networks, in Proc. IPSN, 2005.

[MOO04] Moore D., Leonard J., Rus D., Teller S. Robust Distributed Network Localization with Noisy Range Measurements. Proc. of SenSys, Baltimore, MD, 2004.

[NAG03] Nagpal R., Shrobe H., Bachrach J. Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network. Proc. of the 2nd Int. Workshop on Information Processing in Sensor Networks, April, 2003.

[NIC01] Niculescu D., Nath B., Ad Hoc Positioning System (APS). Proc. of IEEE Globecom, Nov, 2001.

[NIC04] Niculescu D., Nath B. Error Characteristics of Ad Hoc Positioning Systems (APS),. Proc. ACM Int Symp on Mobile Ad Hoc Networking and Computing, 2004.

[PRI00] Priyantha, N.B., Chakraborty A., Balakrishnan H. The Cricket Location-Support System,. Proc. ACM Int. Conf. on Mobile Computing and Networking, August, 2000.

[SAV03] Savvides A., Park H., Srivastava M.B. The N-Hop Multilateration Primitive for Node Localization Problems," Mobile Networks and Applications, 2003

[WAN92] Want R., Hopper A., Falcao V., Gibbons J. The Active Badge Location System. ACM Transactions on Information System, vol 10, issue 1, Jan 1992, pp. 91-102

[WHI06] Whitehouse K., Karlof C., Culler D. A Practical Evaluation of Radio Strength for Ranging-based Localization. Proc. ACM Mobile Computing and Communications Review (MC2R), 2006.

[ZHO04] Zhou G., He T., Krishnamurthy S., Stankovic J.A., Impact of Radio Irregularity on Wireless Sensor Networks. Proc. MobiSys, 2004.