Recursive Position Estimation in Sensor Networks

Joe Albowicz, Alvin Chen, Lixia Zhang
UCLA Internet Research Laboratory
{joe, alchemy, lixia}@cs.ucla.edu

Abstract

Recursive hierarchy provides a framework for extending position estimation throughout a sensor network. Given imprecise ranging and inter-node communication, nodes scattered throughout a large volume can estimate their physical locations from a small set of reference nodes using only local information. System coverage increases iteratively, as nodes with newly estimated positions join the reference set, capitalizing on the massive scale of sensor networks. The system frames position estimation as a geometric problem solvable through common nonlinear regression techniques and develops methods for gauging the reliability of position estimates. This provides a flexible framework that can use and enhance a variety of technologies and protocols to produce fine-grained position estimates. A specific model provides a simulation environment showing that over 90% of position estimates are correct to within 3% of the ranging distance with only 5% of the system in the initial reference set.

I. Introduction

Networks of autonomous sensors promise to provide radically new methods for monitoring the physical environment. Though a single sensor node cannot currently replace a properly equipped human observer, the scale, persistence, and low cost of a mesh of many such nodes provides a better solution for a large variety of situations. From large-scale collection of physical data to continuous monitoring of perimeter security, scenarios often lend themselves well to unobtrusive, automated sensor nodes. To be useful in a number of these domains, sensor reports will need to include positional information. Though this often takes the form of a location tag for sensor data, some systems use geographical information for purposes as varied as routing [3, 9, 10] and query expression [8]. An obvious solution could involve equipping each node with Global Positioning System (GPS) [12] receivers to provide absolute positioning, but this currently requires significant increases in size, cost, and energy consumption, multiplied by the scale of the network. More recent approaches often require the highly redundant placement of fixed references [4, 13] or the mass delivery of per-node data to a central unit for processing [5, 14], both of which are unreasonable burdens for many sensor networks. To control system overhead, this work proposes a recursive method for propagating position information throughout a sensor network given a limited number of reference nodes.

The algorithm adopts the Internet design preference for a system of many low-cost nodes over one with fewer, more expensive nodes [2]. While providing absolute positioning to every node would significantly increase per-node cost, in equipment (e.g., GPS receivers) or effort (e.g., manual placement and configuration), the proposed solution provides positioning information to all nodes in a three-dimensional space while restricting heavier investment to a very small subset of the network. Given a few reference nodes with known location and noisy inter-node range estimates, all nodes can derive their positions. Nodes use local information to estimate their positions autonomously, acting as lower-tier references in later iterations, thus improving coverage through a scalable hierarchy. Nodes also attempt to shield themselves from estimation errors and try to limit the propagation of such errors.

Most importantly, this approach allows the reliable designation of new reference nodes, safely expanding positioning coverage across the network. This incorporates the fundamental scale of a sensor network into an implicit hierarchy while allowing operation under flexible assumptions. Reference nodes can be scattered randomly throughout a space, and the system framework can take advantage of a variety of specific hardware and/or software mechanisms. This flexibility adds to the broad applicability of the simple recursive hierarchy.

This paper is organized as follows: Section 2 discusses the design goals of the positioning protocol developed. Section 3 describes the underlying system requirements. Section 4 describes the algorithm, framing the geometric problem posed and discussing its solution through nonlinear regression. Section 5 describes the performance criteria and experimental results obtained via simulation. Section 6 presents an overview of work in related areas, while Section 7 concludes the paper.

II. Design Goals

A. Functionality

Given a sensor network with a few reference points, most nodes with unknown position should eventually derive good estimates of their location. Locations near reference points should be most accurate, but positioning accuracy should decline gracefully for more remote nodes.

B. Scalability

The system must work with arbitrarily large numbers of nodes, often packed into dense configurations. In many situations, sensor instrumentation of an area may require a higher density than needed for communication connectivity. If a given set of reference nodes provides sufficient positioning coverage for an area, the addition of non-reference nodes must not hinder the effectiveness of the system. This emphasizes that information should flow out from the small reference set. Ideally, the system should benefit from the large number of nodes available.

C. Robustness

Nodes should prefer to remain uncertain about their position rather than report inaccurate information, especially when providing data to other nodes. The protocol must prevent the propagation of misinformation. A system may recover from lack of a position in a sensor report, perhaps through information gained from report delivery or parallel reporting, but it will have much more difficulty detecting and correcting falsely reported information. This suggests that nodes keep some estimate of position accuracy and reject overly uncertain data.

D. Sensor Autonomy

Nodes are responsible for estimating their own position from locally available information. A centralized solution often involves prohibitive overhead for delivering input to a "position server" and processing collected data, as well as the difficulties of disseminating that data back to the sensors or providing unique mappings for a massive and unreliable sensor array. Instead, nodes should choose their references based on local estimates of reliability, usefulness, etc., keeping distributed computations at the sensor nodes, where their results are needed. This also allows for rapid, ad hoc distribution and auto-configuration.

III. System Parameters

The current approach extends from a given set of system parameters. Most importantly, the protocol assumes that most nodes do not initially know their current position. They may have been scattered from a plane, undergone occasional movement, or been assembled from components with embedded sensors. Even GPS-enabled nodes may be unable to determine their position due to physical obstacles or other levels of interference. Secondly, the system considers the geographic location of sensors to be important but is not able to guarantee that each sensor will have this information available. For other systems, high-level abstractions (e.g., rooms, buildings, printers, as in [11]) or logical relations (e.g., button sensors) may be more appropriate. Nodes should also be able to deliver information to others (e.g., wireless broadcast). Finally, some form of inter-node ranging must be possible (e.g., timing [11, 12, 13, 14], signal strength estimation [1]). Many systems may already include noisy ranging capabilities as part of some other system, such as inter-node wireless communications. In most systems with immobile nodes, positioning only needs to occur once, though a system with low mobility could probably revise position estimates through later iterations.

IV. Position Dissemination Algorithm

Primarily, the current algorithm works through three-dimensional "triangulation," using four or more reference points, though it could take advantage of other position information. Given a high enough level of certainty, a node may then advertise its own position, acting as a reference point for its neighborhood and extending system coverage. There are four phases to the protocol. In the first phase, a node determines its reference points. In the second, the node obtains or uses already collected range estimates for the selected nodes. In the third phase, the node estimates its position. In the final phase, a node may advertise its own position for use as a reference in the next cycle.

A. Reference Selection

In the reference selection phase, a node collects information from nearby reference points. Once it becomes aware of a few references (e.g., by receiving randomly timed broadcasts), it must have a way to rank them. References advertise their residual value, a measure of confidence, and the node chooses those with the lowest residual values for use in the next phase of the algorithm. The residual value for an estimated position (x, y, z) is defined as:

$$residual(x,y,z) = \sum_{i \in reference} \left(\sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} \; - d_i \right)^2$$

where (x_i, y_i, z_i) is the ith reference's position, and d_i is its measured range. In other words, the residual value is the sum of the squared differences between the range from the estimated position and the measured range. For the original reference set, the residual values should be near zero.

B. Distance Measurement

Once the node chooses a set of reference points, it then collects distance estimates to each reference. For some ranging techniques, the node would have obtained this information when it received the original advertisement. For less precise ranging methods, the node can collect a number of samples until the sample variance falls below some threshold. The node would then use the sample mean as the distance between the reference point and itself. The node can also reject a reference point for which it cannot obtain a stable distance estimate.

C. Position Estimation

Once a node estimates its distance to each of its reference points, it can estimate its position. Each reference point monitored by a node yields an equation of the following form:

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} + e_i$$

(ε is an error term). Typically, one estimates a position from such a system of equations through nonlinear regression ([7, 13]), detailed discussion of which can be found in [6, 7]. For the implementation discussed in this paper, the nodes used the linearization, or Taylor series, method. This method approximates the nonlinear system as a linear system by performing a Taylor series expansion about an initial estimate of the parameters and then solving for the parameters using standard regression. Linearization iteratively refines the estimated parameters. Unfortunately, in some cases the method may converge to erroneous values, oscillate wildly, or diverge [6]. Various techniques help minimize the likelihood of these situations. Careful selection of the initial estimated parameters (currently, an average of the reference positions) minimizes incorrect convergence. By only accepting parameters that have reached a quiescent state (i.e., subsequent iterations continue to yield estimates that differ only by a small value), one avoids oscillation and most divergent cases. Finally, a node rejects a position estimate with a high residual (over 0.01 m^2 in our simulations). This further screens out incorrect convergence and divergence.

D. Next-Level Advertisement

If the node obtains a reasonable position estimate, it may participate in the protocol as a reference point. The current implementation requires the node to pass a more restrictive residual test before advertising itself as a reference point. Therefore, while most nodes should manage to get good estimates of their positions, only the most accurate should extend system coverage, in order to avoid increasingly erroneous values. The new reference points increase both the number of nodes that can find their position and the accuracy of their estimates. We refer to nodes that decide to act as references during the ith iteration as level i references (the initial references are level 0). When applied recursively, this process allows the system to turn the challenge of massive scale into an advantage for the sensor network.

V. Simulation and Analysis

A. Accuracy For Single Nodes

Before investigating system performance in a full network, we first discuss accuracy for a single unknown node with at least the minimum number (four) of reference points. We explored single-point estimation through sets of 10,000 independent simulations. Each run randomly generated a given number of references within a limited ranging distance (10 m) around the unknown point (i.e., uniform distribution within a sphere). To simulate noise, each range sample was disturbed by a normal random variant with a mean of 0.1% of the range and a standard deviation of 1% of the range. The experiment did not add ground interference for estimates between altitudes. Figure 1 shows the cumulative density function for position error, (i.e., the probability that an estimate's error is less than that shown on the logarithmic x-axis). Figure 1 shows that the quality of the position estimate increases dramatically with the addition of reference points, but even with only four references, over 90% of the nodes suffer less than 0.12 meters of error. Table 1 shows the mean and worst errors, along with the residual value for the worst case. During normal operation, a residual test avoids large errors by rejecting estimates with residuals of over 0.01-m^2. While this avoids most position errors, the worst-case run with four references shows that a degenerate set of references can still produce misleadingly small residual values. That is, if the four references are roughly coplanar, a small skew in the distance measured to one of the reference points can result in a large position error, because coplanar references generate two possible solutions to the positioning equation. Figure 2 shows the two-dimensional version of this problem: collinear references (A, B, and C) generate two solutions (D and E). To avoid this problem, one can wait for additional reference points if the current set is nearly collinear (i.e., the perpendicular distance of a reference point to the line between the furthest separated references is below some threshold). In three dimensions, one would use an analogous planarity test.

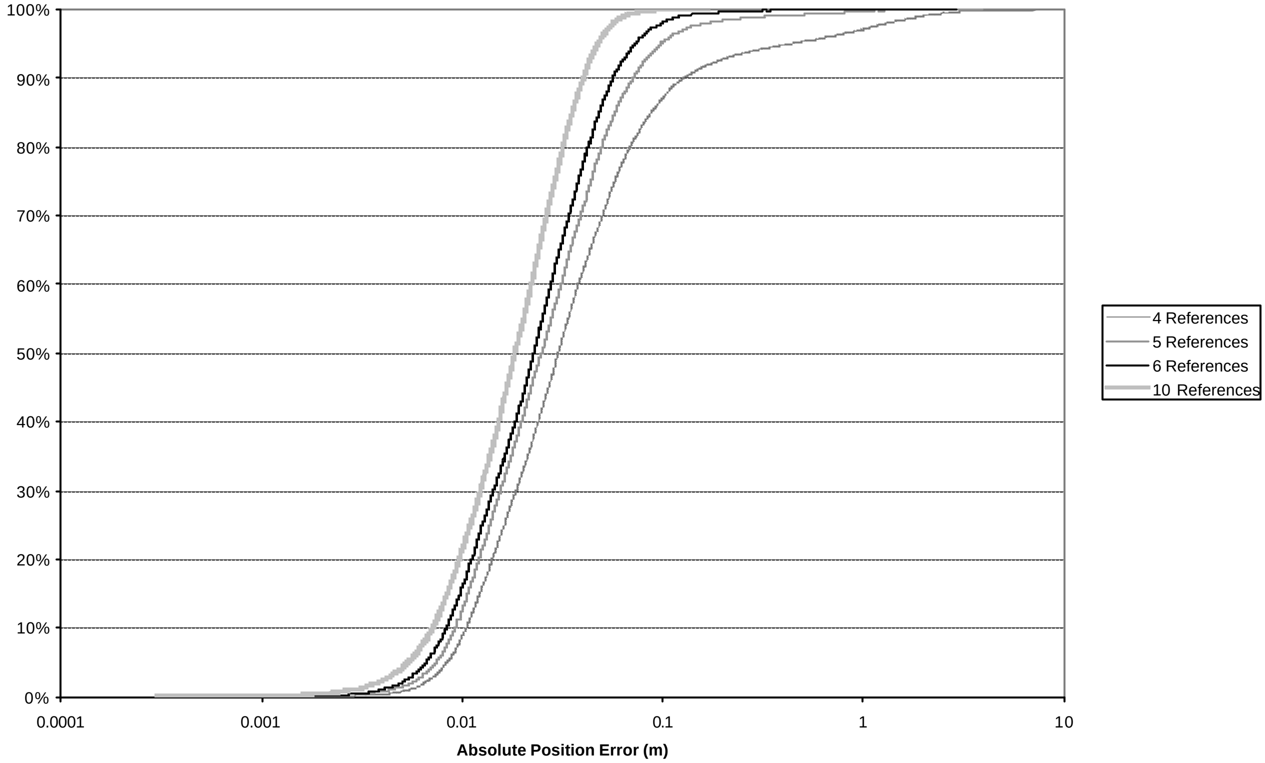

Figure 1. Cumulative distribution of single-node position estimation error

Number	Mean	Max	Residual
4	0.114	11.7764	0.0003
5	0.043	3.9118	0.1604
6	0.030	2.8896	0.0334
10	0.021	0.1691	0.0006

Table 1. Single-node position estimation error with residuals for worst cases

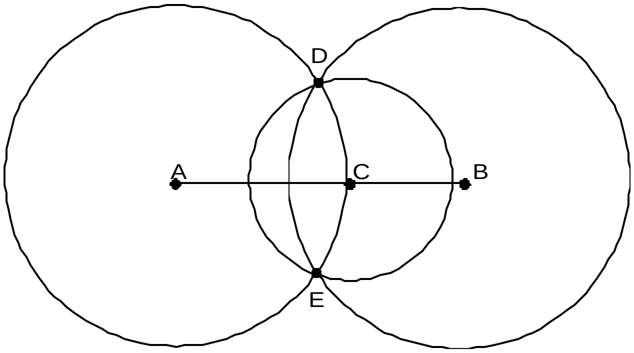

Figure 2. Degenerate case (collinear references)

B. Obstructed Path Errors for Single Nodes with Four References

For some ranging methods, distance estimates to reference points may experience disruptive phenomena. Errors may occur when an obstacle blocks the direct path but allows range estimation along an indirect path. For example, the acoustic signals used by some ranging systems easily reflect from surfaces. We refer to these types of errors as obstructed path errors. The effects of different skews were investigated in single-node simulations with four references, one of which suffers from an obstructed path. Therefore, the simulations conducted were similar to those used to produce the four-reference curve above, with the addition of a large skew. The skew varied from 10 cm to 5 m. Table 2 displays the percentage of estimates that pass the residual threshold test, as well as the mean and maximum errors for accepted position estimates. Clearly, the percentage of nodes accepting an estimated position drops quickly with the magnitude of the obstructed path error, allowing most nodes to seek better reference points, but skews exacerbate problems with planar ambiguity.

Obstruction	%	Mean	Max
0.1	99.46	0.510	14.3914
0.5	31.19	1.764	13.9475
1	15.60	2.680	12.4098
5	1.89	7.374	14.9973

Table 2. Single-node position estimation acceptance with obstructed paths

The above results show that given four or more reference points, a node can determine a very accurate position estimate, even in the face of conflicting noise. Additionally, the method reliably shields itself against environmental conditions that might skew distance estimates. This allows for the use of a variety of ranging methods, some of which may suffer from significant estimation errors. Full implementations can also take a number of distance samples, allowing a node to average out time-varying noise or ignore temporary noisy phenomena.

C. Accuracy and Coverage for Sensor Networks

The system can obtain greater coverage by having nodes advertise themselves as reference points once they have estimated their own positions. These results were averaged over 10 topologies wherein 2000 sensors were randomly distributed throughout a 100-by-100-by-2.5 meter volume. The simulations fixed the ranging radius of each node at 10 meters, and they designated 5% of the nodes as level 0 (i.e., original reference points). Figures 3 and 4 summarize the results. Figure 3 shows that the increased availability of reference nodes decreases positioning error significantly, even though many of the new reference points advertise derived estimates of their own positions. In fact, over 90% of the nodes have less than 0.28 meters of error. Random distribution also decreases the effect of degenerate reference sets, as only 0.1% of the estimates have errors over 5 meters. Figure 4 shows that the recursive referencing increases system coverage by over 140%, allowing over 90% of the nodes to determine their positions.

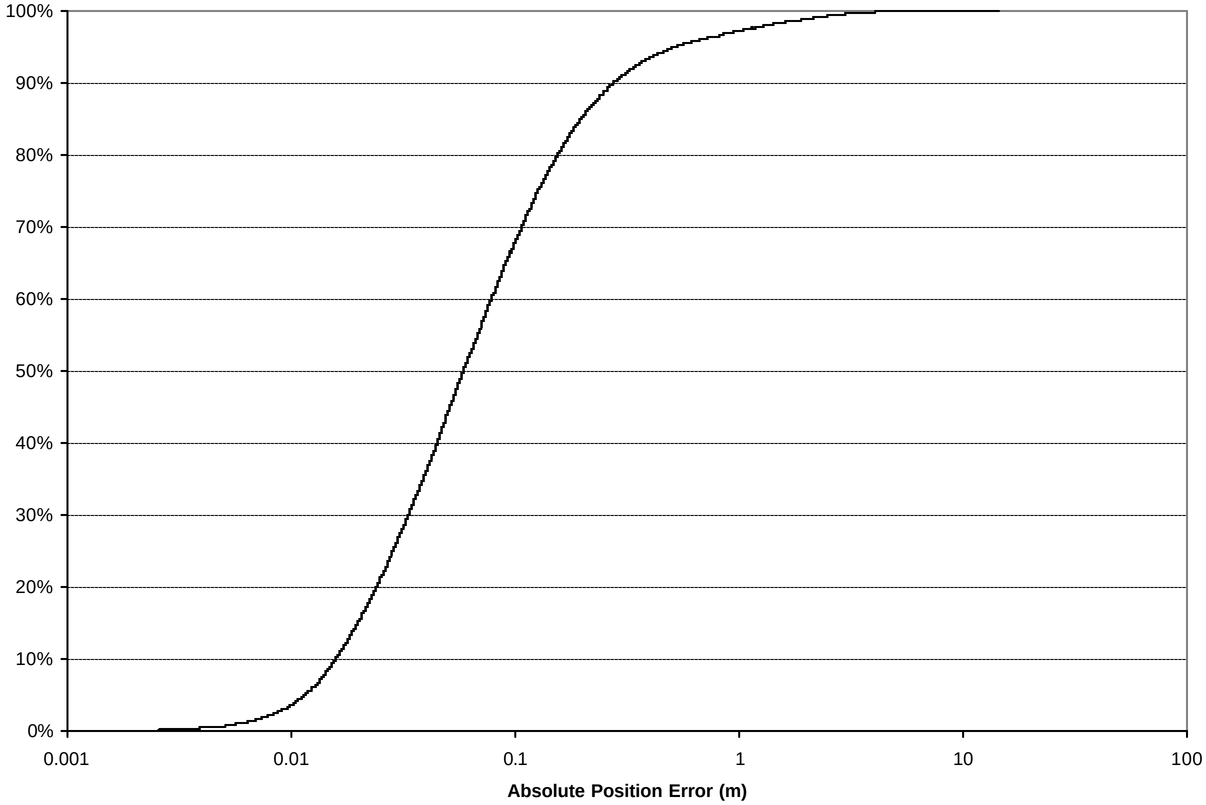

Figure 3. Cumulative distribution of position estimation error

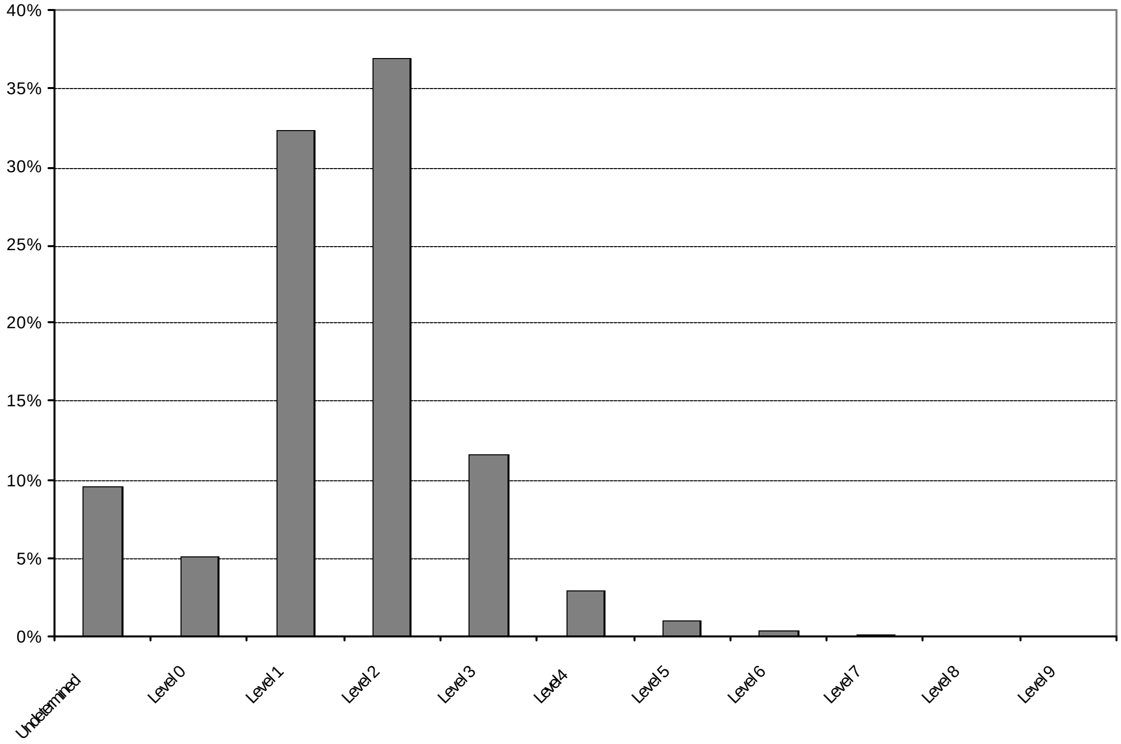

Figure 4. Node level distribution (recursion depth)

D. Positioning Time

The time involved in positioning depends strongly on the time it takes a node to obtain its ranging estimates. While ranging via an ultrasonic beacon might be done with the initial reference broadcast, noisy signal strength estimations might benefit from a series of samples. This defines the fundamental length of an iteration, which adds one level to the system hierarchy. Figure 4 shows that the bulk of all position estimates finish in the first two iterations, while system coverage shows diminishing expansion until the sixth iteration. While nodes could incorporate new reference broadcasts at any time, most immobile sensor networks would only need to undergo positioning during a short initialization phase.

VI. Related Work

A popular approach to positioning is the Global Positioning System. A GPS receiver estimates its distance from a number of satellites by timing the satellites' signal propagation. The receiver estimates an offset between its local clock and satellite system time to perform the position estimation. The clocks require precise synchronization, and receivers must avoid overhead cover. These drawbacks make full GPS unrealistic for most large sensor networks.

Other research in positioning often takes a centralized approach. In [13], the "Active Office" a) places a matrix of receivers in the ceiling of an office, b) attaches a wireless transmitter to each object, c) polls the receivers from a control unit for range information, and then d) estimates each object's position. The regular placement of receivers and the polling of transmitters automatically restrict the applicability of this approach. A similar system was designed for "tags" that need not be aware of their position, instead letting a central inventory system record that information [14]. For some inventory tracking systems, this is a practical approach, but a sensor network often requires that each node have this information available. The potential scale and unreliability of a sensor network also defies conventional centralized control.

Still, a number of recent sensor network positioning methods take similar approaches. [4] uses a grid of reference points, like that of [13], but reverses the flow of information. Given higher communication overlap or, equivalently, higher reference point density, the averaged positions of in-range reference points give better position estimates. This approach might benefit from recursive referencing as presented in this paper, since the increased reference density would improve the accuracy of later estimates. Unfortunately, their initial estimates suffer from relatively large errors, and nodes have no gauge of confidence level, allowing misinformation to accumulate within the system. Along the other direction, [5] again requires that nodes deliver connectivity information to a central processor, here for solution as a convex optimization problem. Very few well-placed reference points, or many more randomly distributed ones, can provide well-bounded position estimates. Even allowing for centralized solution, the constraints in the problem are susceptible to errors, since lack of connectivity implies great separating distance, though nearby nodes may actually just be blocked by an obstacle or intermittent noise. Inaccuracies might cause solutions to oscillate or, worse yet, make the problem infeasible.

VII. Conclusion

This paper presents a basic framework for extending positioning coverage across a sensor network. The algorithm correctly estimates three-dimensional position given noisy range estimates in a local volume, and it rejects large errors that may arise in the ranging process. This produces an implicit, scalable hierarchy through simple recursion.

Future work will address the refinement of position estimates and development for specific scenarios. Planarity tests could block rare degenerate cases, while groups of connected nodes should be able to identify members with large position errors (at the cost of node autonomy). Nodes might also try to select the best references as they become available. In GPS, receivers typically try all combinations of four reference points to reduce position error. At the scale of sensor networks, this combinatorial approach is intractable. Here, the nodes simply made an estimated ranking of the available references by residual, but heuristics that take reference distance, recursion level (since earlier iterations have less opportunity for accumulated position error), and/or other knowledge into account might yield better results. Similarly, experiments should explore the impact of reference distribution. While this paper uses the general case of random distribution, some situations might allow a user to place reference nodes in a specific pattern, or they might restrict the reference area (e.g., overhead cover blocking access to GPS satellites). Finally, the incorporation of other methods, like those mentioned in the related work section, into the framework of the system could provide more accurate position estimates and/or extend the applicable domain of the system.

VIII. Acknowledgments

We would like to thank Fan Ye for his comments and contributions during the development of this work. Thanks also go to Tony Givargis for providing matrix code for the simulations. Finally, we would also like to acknowledge the comments of Songwu Lu and the anonymous reviewers during the finalization of this paper.

IX. References

[1] P. Bahl and V. Padmanabhan, “RADAR: An In-Building RF-based User Location and Tracking System,” *Proc. IEEE Infocom 2000*, March 2000.

[2] P. Baran, “On Distributed Communications Networks”, *IEEE Transactions on Communications*, March 1964.

[3] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, “A Distance Routing Effect Algorithm for Mobility (DREAM),” *Proc. ACM/IEEE MobiCom’98*, October 1998.

[4] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less Low Cost Outdoor Localization For Very Small Devices,” *IEEE Personal Communications*, Vol. 7, No. 5, October 2000.

[5] L. Doherty, K. Pister, and L. El Ghaoui, “Convex Position Estimation in Wireless Sensor Networks,” *Proc. IEEE Infocom 2001*, April 2001.

[6] N. Draper and H. Smith, *Applied Regression Analysis*, John Wiley & Sons, 1966.

[7] J. Farrell and M. Barth, *The Global Positioning System and Inertial Navigation*, McGraw Hill, 1999.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks,” *Proc. ACM/IEEE MobiCom 2000*, August 2000.

[9] B. Karp and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” *Proc. ACM/IEEE MobiCom 2000*, August 2000.

[10] Y. Ko and N. Vaidya, “Location-Aided Routing (LAR) in Mobile Ad Hoc Networks,” *Proc. ACM/IEEE MobiCom’98*, October 1998.

[11] N. Priyantha, A. Chakraborty, and H. Balakrishnan, “The Cricket Location-Support System,” *Proc. ACM/IEEE MobiCom 2000*, August 2000.

[12] US Naval Observatory (USNO) GPS Operations, http://tycho.usno.navy.mil/gps.html, April 2001.

[13] A. Ward, A. Jones, and A. Hopper, “A New Location Technique for the Active Office”, *IEEE Personal Communications*, pp. 42-47, October 1997.

[14] J. Werb and Colin Lanzl, “Designing a positioning system for finding things and people indoors”, *IEEE Spectrum*, September 1998.