Simple Position Estimation for Wireless Sensor Networks

Toks Adebutu Lionel Sacks† Ian Marshall‡

† Electrical and Electronic Engineering, University College London, ‡ BTexact Technologies

Abstract: Wireless sensor networks need to have positioning information in order to form a coherent picture of the environment they are sensing. Traditional methods of providing this information, such as GPS, fall outside the cost, power and processing constraints of these networks. This paper presents a solution to this problem based on an exploration of the possibilities that exist given the network constraints and using some of these hidden constraints to an advantage. The method is based on the assumption that inter-node ranging and communication is possible and that there exists devices within the network that know their absolute position via GPS or hard coding. The technique uses "iterative averaging" to enable the network converge to a stable solution. This approach gives fast, approximate localization information, given the constraints of sensor networks.

1. Introduction

Wireless sensor networking presents an alternative to traditional methods of environmental monitoring and surveillance. These sensor networks of small, low cost, low power devices that combine data processing with multiple sensing and the ability to communicate wirelessly will provide cheaper and in many cases more detailed information than current methods in wide usage. Many open research areas exist in this field. One particularly important issue is how a node's location is uniquely determined; an important detail in recreating the environment for analysis after obtaining the results. This is by no means a trivial problem, as the nodes are deployed in an ad-hoc fashion and due to the cost, power and size constraints it is not possible to equip them with a sophisticated localization system such as GPS.

Consequently, it is necessary to develop an approach for locating nodes within a sensor network, using only local information and able to operate within the tight resource constraints of sensor nodes. Additionally, since the nodes are potentially deployed in inhospitable environments with limited accessibility, it is important that the devices operate autonomously and are capable of adaptation to environmental changes. Fault tolerance is another important issue as devices are prone to failure. They are likely to get buried as a result of high winds or strong water currents, or malfunction due to their manufacture by low cost mass engineering. It is feasible that nodes will temporarily lose communication, due to changing weather conditions; storms, high humidity, etc. This paper presents a self-organizing, scalable, fault-tolerant solution that attempts to solve this problem based on the assumption that nodes are able to measure distances between themselves within a bounded error, using some ranging capability.

2. Related Work

There have been numerous attempts to solve the problem of node localization in ad-hoc networks. In [1], Hightower and Borriello provide a general overview of the work in this field. Most of the methods assume that the nodes have some ranging capability [2, 3, 4, 5], which introduces an additional factor that the algorithm needs to cope with. There are also methods [6, 7, 8], that do away with these ranging errors, but have complexities of their own. The commonality between all these techniques is their computational complexity. "Iterative multilateration", as presented in [3] uses an approximation to the Kalman filter. While this considerably reduces the complexity, this is still a fairly computationally intensive process that increases in complexity and takes an increasing amount of time to converge as the number of unknown nodes increases. A similar iterative multilateration approach using linear regression is adopted by Robinson in [4]. Convex optimization, as used in [5], provides a position estimate based on the geometric constraints established by inter-node connectivity. A drawback of this approach is the need for a central node to perform the optimization which is not possible in our system because this will require heavy computation from this one node and significantly shorten its lifetime. Additionally, processing power necessary for this computation is simply not available on an 8-bit processor. These methods all produce accuracies in the range of 10% to 100% radio range or 33% of the separation distance between position aware nodes.

As mentioned above, sensor networks are limited in a number of ways. Any algorithm will need to (a) have a small code footprint; storage is very limited and invariably size is related to complexity and required computation; (b) ranging techniques are mostly very inaccurate and it will be necessary to adapt to this; and (c) devices are very simple and will invariably not have floating point arithmetic capabilities. For these reasons it is necessary that the algorithm is uncomplicated and not limited by the capability of the devices. Algorithms that aim to attain the highest accuracy, within only some of the constraints of the network are not ideal. An alternative approach is to exploit the hidden network constraints, such as finite communications range and symmetry, to develop a robust algorithm that fits into the devices' performance profile. Irrespective of how clever an algorithm is, if the ranging device can only work to an accuracy of centimetres, it is impossible to get a measurement accurate to within millimetres, or whichever units are in use. One method that has been implemented and investigated by myself is *iterative multilateration*, as presented by Robinson in [4], with some modifications.

3. Iterative multilateration

The algorithm works on the premise that there are two types of sensor nodes; *position-aware* (PA) and *position-determining* (PD) nodes. Aside from knowing their exact location, the position-aware nodes are identical to the position-determining nodes. Each node measures the distance (± the intrinsic error of the ranging equipment) between itself and all nodes within communication range. The next step of the algorithm is for all nodes to make an initial estimate of their position. Once all nodes have done this, each node uses this information to compute a *perceived error* for its position estimate. This perceived error is then used to correct the current location guess so as to reduce this error. The perceived error ?E, is considered to be a vector that moves the position estimate of a node to a point where all distance data is satisfied.

If every node moves to this point, the positioning error will either oscillate wildly or increase exponentially. To avoid this, position estimates are modified by only a small amount; (*%mov*). The optimum value of *%mov* was determined experimentally, and it was found that moving between 2 – 4% along the perceived error was optimal. Each iteration shifts a node's position estimate to a point of lower perceived error and the nodes' position estimates are gradually moved from the estimated (grey) positions to the real (black) positions. Figure 1 shows the reduction in positioning error as the algorithm converges for a run of the simulation. The introduction of a reset clause allows the algorithm to converge to a point of zero perceived error [4]. This takes anything up to 2000 iterations. There are many applications that require a high level of accuracy and have the resources to achieve it, but there are just as many, for which the power, processing and latency costs are too high. For these applications, "*iterative averaging*" is presented.

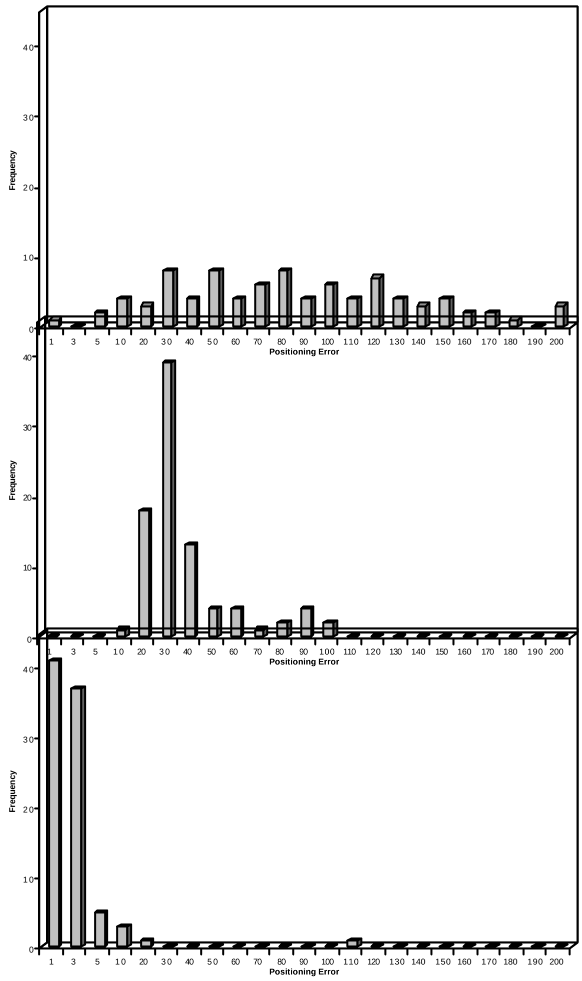

Figure 1: Initial position estimate for iterative multilateration.

4. Iterative averaging

This method proposes all PDs set their position estimates to the average of the position estimates of all the nodes within communication range that have made at least one position estimate. This process is initiated by the PAs sending out their positions to all neighbouring nodes. These then send their position estimates out to their neighbouring nodes. The process continues iteratively until the network reaches a stable state and position guesses are no longer changing.

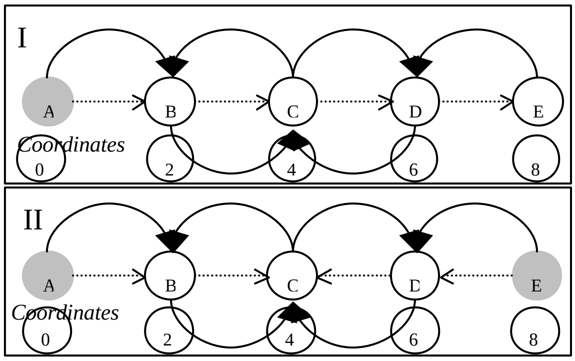

Figure 2: Information flow between nodes in iterative averaging.

For this to yield useful results there needs to be at least three position aware nodes in a fully connected two dimensional network. This can be illustrated by a simple one-dimensional example. Assume a network with 5 collinear nodes. One node, *A* is a position determining node, as depicted in Figure 2I. At iteration 0, the position determining node sends its coordinate to node *B* – as shown in Figure 2. As none of *B*'s neighbours have previously made a guess, its position estimate is set to 0. *B* sends this new information to *C* and the same happens iteratively until *E* ends up with a position estimate of 0. Given that all the nodes have the same coordinates if they continue sending their estimates to each other and taking the averages, they will still all have estimates of 0.

This obviously has no use in any situation. In order to overcome this, more position aware nodes are needed in the network. With two position aware nodes, at the first iteration, both *A* and *E* send position estimates to *B* and *D* respectively. *B* and *D* then set their coordinates to 0 and 8 respectively. They are then both in a position to send their estimates to *C*, which then receives this information and sets its estimate to 4; the average of both values. *C* now sends this value back to both *B* and *D*. *B* computes the average of the values it receives from *A* and *C*, arriving at a coordinate of 2. *D* does the same for *C* and *E*, arriving at 6. Even with further information exchange none of the nodes will change their position estimates further. This shows that algorithm has converged to a stable solution. Examining the coordinates, we see that they are correct.

Intuitively we can see that if the nodes send out their positioning information to all other nodes within communication range, this model extends to both two and three dimensional problems and satisfactorily determines the positions. The example presented here, however, is a trivial problem since the separation distance between nodes is uniform. In practise, this is not realistic, as separation distances vary by a great deal and this will affect the accuracy of the algorithm as it stands. To tackle this, a refinement phase is introduced, once the seeded algorithm has converged to a stable set of position estimates. This will factor in a weight to the above averaging process such that correlation between numbers of neighbours, average distance of neighbours, distance to nearest PD and positioning error is exploited. Evidence of this correlation is presented in the next section.

5. Results

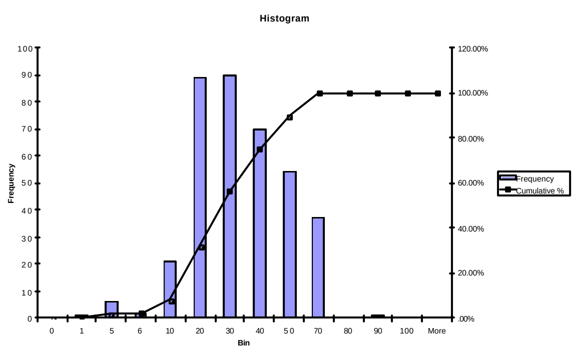

Figure 3: Histogram of positioning error.

This section presents the preliminary results obtained for the algorithm. All experiments were carried out on a 400×400 grid, with 16 evenly distributed position aware nodes having no communication between each other. All nodes had an effective communication range of 50. The algorithm converged on average in ten iterations, which is significantly faster than most current methods. However, 85% of nodes were located within one radio range, which is also half the separation distance between position determining nodes. These claims are supported by Figure 3.

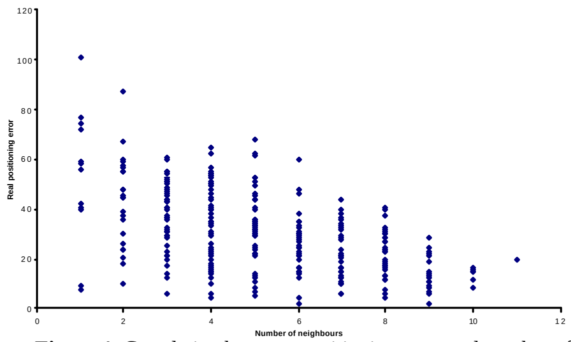

Figure 4: Correlation between positioning error and number of neighbours.

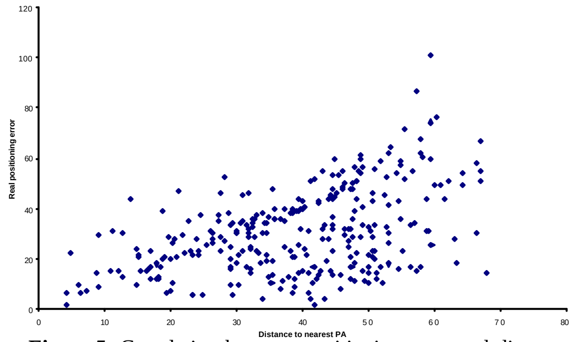

Figure 5: Correlation between positioning error and distance to nearest PA.

The position estimates need a significant improvement in accuracy and precision, as there are very few applications that can cope with the current levels. Further analysis of the data showed a negative correlation between the number of neighbours and the positioning error; Figure 4 and a positive correlation between the distance to the nearest position aware node and the positioning error; Figure 5. As a result, the introduction of a *refinement* phase once the initial *seeding* process is complete is proposed. This *refinement* will introduce weights that factor in the number of neighbours, distance to the nearest beacon and the distance to the neighbours, which as yet, has not been used. Up till now, the algorithm has functioned independent of range and has therefore not been influenced by ranging errors.

6. Conclusion

Whilst significantly better results can be obtained by a number of existing methods, the results attained here suggest the possibility that a first principles approach can be almost as effective these methods. Work that has been carried out to develop a similarly complex algorithm to those discussed here shows how quickly trigonometric, quadratic and even higher order terms are introduced, which need intensive processing to compute. Sensor networks are implemented for a wide range of applications, with widely varying operational requirements. These requirements are invariably conflicting; recording all environmental changes, while conserving battery power. To record all environmental changes, it is necessary to sample frequently, but this in turn requires considerable power, which conflicts with the power conservation requirement. Unless device technology advances further, sensor networking will continue to be characterised by large trade-offs. The trade-off in this algorithm is accuracy for cost, speed and power. Various applications will require differing trade-offs and a one-size fits all approach is currently unworkable.

References

[1] J. Hightower & G. Borriello, "*A Survey and Taxonomy of Location Systems for Ubiquitous Computing*," IEEE Computer, 34(8):57-66, August 2001

[2] C. Savarese, J. Rabaey & K. Langendoen, "*Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks*," USENIX Technical Annual Conference, Monterey, California, June 2002

[3] A. Savvides, H. Park & M. B. Srivastava, "*The Bit and Flops of the N-hop Multilateration Primitive for Node Localization*," Proceedings of the First International Workshop on Wireless Networks and Applications held in conjunction with Mobicom 2002, Atlanta, Georgia , September 2002

[4] D. P. Robinson & I. W. Marshall , "An *Iterative Approach to Locating Simple Devices in an ad-hoc Network* ," London Communications Symposium 2002, London, September 2002

[5] L. Doherty, L. Ghaoui & K. S. J. Pister, "*Convex Position Estimation in Wireless Sensor Networks*," Infocom 2001, Anchorage, April 2001

[6] N. Bulusu, J. Heidemann & D. Estrin, "*GPS-less Low Cost Outdoor Localization for Very Small Devices*," Technical report 00-729, Computer Science department, University of Southern California, April 2000

[7] T. He, C. Huang, B M. Blum, John A. Stankovic & Tarek Abdelzaher, "*Range-Free Localization Schemes for Large Scale Sensor Networks*," to appear in the 9th Annual International Conference on Mobile Computing and Networking (MobiCom 2003), San Diego, California, September 2003

[8] D. Niculescu & B. Nath, "*Localized Positioning in Ad Hoc Networks*," in Elsevier journal of AdHoc Networks and in IEEE SNPA 2003, Anchorage, Alaska, May 2003