Refined Statistic-based Localisation for Ad-Hoc Sensor Networks

Tom Parker* Koen Langendoen

Faculty of Electrical Engineering, Mathematics, and Computer Science
Delft University of Technology, The Netherlands
{T.E.V.Parker, K.G.Langendoen}@ewi.tudelft.nl

*Supported by the Dutch Organisation for Applied Scientific Research (TNO), Physics and Electronics Laboratory

***Abstract*— Localisation is required for many ad-hoc sensor network applications. In this paper we look at the limitations of many of the existing proposed localisation techniques with regards to coping with non-uniform anchor distributions and errors in ranging information. We present a refined approach that uses a combination of mobile anchor scenarios for anchor information distribution, along with statistical techniques for performing localisation with inaccurate range data. Simulations with our refined approach have shown significant reductions (in the order of magnitude range) to the required processing for performing statistical localisation over previous attempts, as well as improving the generated location information in situations with non-total anchor information coverage, making possible a wider range of applications.**

I. Introduction

Many possible applications have now been thought of for Wireless Sensor Networks (WSNs), and a significant number of them rely on location information in order to perform their designated function. The main purpose of a WSN is information gathering, and gathered data is only useful if you know what it applies to. For example, the data "the temperature has gone up by 10 degrees" is not very useful, but the information "the temperature has gone up by 10 degrees in room 3C" is a lot more interesting. Location information gives us a context, which allows us to actually use our gathered data. For example, monitoring room temperature can be used to control when to switch air-conditioning systems on and off. When detailed location information is present it might even be possible to personalise working conditions within a shared office (i.e. individual settings per cubicle).

Location information is important in many domains, hence various approaches have been proposed, of which some were even constructed and deployed on a large scale (e.g. GPS). Within the WSN community, specialised localisation algorithms have been developed that address the problems associated with the lack of infrastructure (i.e. GPS satellites) and the limited resources leading to incomplete and inaccurate information. A survey of initial approaches is presented by Hightower and Borriello in [1]; recent work includes [2], [3], [4], [5] and [6].

With WSN localisation, some nodes are referred to as "anchor" nodes, and some are not. The difference is that anchor nodes have a reliable source of location information, and non-anchor nodes do not. Many localisation techniques rely on anchors, but others do not. So called "anchor-free" localisation systems rely on the idea of building a local co-ordinate system based purely on the existing topology of the nodes, which provides the nodes with a location within the local system. In practise however, that location information would require further processing to integrate it with other co-ordinate systems (e.g. latitude/longitude). We are concentrating on anchor-based rather than anchor-free localisation techniques, but there may be future applications for the integration of anchor-free techniques in the event that anchors have not yet been introduced into the network.

A major problem with localisation techniques (both anchored and anchor-free) is acquiring accurate range information between pairs of sensor nodes. This can be done in a variety of ways, ranging from simple techniques like Radio Signal Strength Indication (RSSI), time of flight data for various sensor types (e.g. ultrasound), to more complex ideas like time of flight difference (which measures the difference between two incoming signals travelling at different speeds). In each case, there is generally some error in the ranging information, which localisation algorithms must be aware of and be able to work with.

II. Mobile anchor localisation

In this section we look at how anchor information can be distributed across an ad-hoc sensor network, and how mobile anchor scenarios have several advantages over other methods.

A. Anchor distribution

Most methods for providing location information to a sensor network start with adding additional localisation hardware (e.g. GPS) to a small percentage of the nodes in the target area. These anchor nodes will initially gather accurate location information on their own, and then transmit this information to their neighbouring nodes. This approach has a number of major faults:

- Most localisation algorithms based on "spread anchor" scenarios rely on the anchors being evenly distributed across the sensor network. This is unlikely unless special care is taken to ensure of this. Given a small anchor percentage (as in most proposed applications), there is a high probability that there will be regions of the sensor grid that have insufficient anchors, leading to problems in attempting to localise nodes in those regions.
- Anchor nodes are generally more expensive, which creates a difficult decision regarding the balancing of the application requirements between having improved accuracy (lots of anchors) and reducing the overall cost of the network (few anchors).
- The additional anchor hardware is often only useful during the initial phase of the network setup, and is then mostly surplus to requirements. An anchor may also have a reduced operational lifespan due to the power drains of the localisation hardware.

There have been some attempts to fix these problems (Adaptive Beacon Placement [2] for example), and there are partial fixes, but a better approach is to look at other ways that location information can be distributed rather than the use of static anchor nodes.

B. Mobile anchor scenarios

Mobile anchor scenarios [6] are an alternate approach, resolving a number of the problems with the spread anchor scenarios. This approach uses a single, large anchor capable of moving along a path. This large anchor could be carried by a car or a person for example. The intention is that this larger anchor will have effectively unlimited power (i.e. can transmit as many messages as needed) because it is intended to be more easily accessible than the individual sensors, and so replacing the anchor node's batteries is less of a problem than replacing batteries in the sensor nodes.

As the mobile anchor moves, it broadcasts its location at regular intervals (either every few seconds, or after it has moved a short distance from its last broadcast location), thus creating a series of "virtual" anchors, as in Figure 1. Each circle represents a position where the mobile anchor has broadcast its current location.

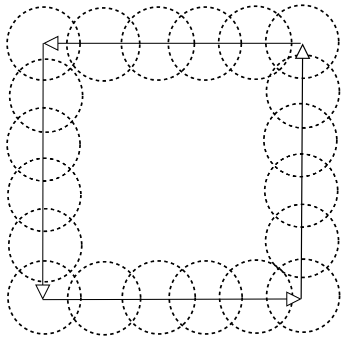

Fig. 1. Example mobile anchor scenario

C. Real-world applications

To see how mobile anchor scenarios map onto various applications, we looked at the structure of these applications, and saw how we could better utilise the already available resources. The main area of interest regards the method for the distribution of the sensors. These methods can be grouped into two categories depending on the distance from the object that is placing the nodes to the location that the nodes are being placed.

The simplest scenarios are when the distance is less than the nodes' radio range (ideally much less). In this case, the placing object itself (be it a person or a car) is the mobile anchor. This can be achieved by combining an anchor node with the placing object (either carried by the person, or attached to the car). It can then broadcast its location information as it places the nodes, thus providing a path that passes near all of the nodes.

More complicated are the situations where the nodes are far away from the placing object, for example when dropping nodes from a plane (especially from a high altitude, or when obstacles are likely to block radio signals from the placing object). One solution to this problem is that the plane could also drop one or more small robots fitted with localisation equipment. These robots could travel along a semi-random path around the sensor grid, providing location information to the sensor nodes as they move around.

D. Advantages

There are several main advantages of mobile anchor scenarios:

- Instead of many anchor nodes (and having to make the trade-offs regarding how many) we have effectively many anchor nodes, but for the cost of only one. All of the sensor nodes should have similar lifetimes, without the additional power drains that would occur if some of them were also anchors for the network.
- In the complicated scenario with the use of mobile anchor robots, the cost of the scenario does go up from what would be possible with more simple scenarios. However, the robots could also be fitted with additional sensors, so that once they have finished providing location information to the network, they can be moved to locations where interesting events are happening to gather more detailed information.
- In the event that the initial anchor path is not sufficient to provide good location information for all of the sensor nodes, we may (depending on the application) be able to do on-the-fly improvements in bad areas.

III. Existing localisation methods

In this section we will have a brief look at other localisation algorithms, with an emphasis on their capabilities regarding the handling of inaccurate range information and their ability to handle non-uniform anchor distributions.

A. Deterministic methods

Langendoen and Reijers [7] studied three localisation algorithms that can handle a low number of anchors (Euclidean, Hop-Terrain, Multilateration), and identified a common three-phase structure. First, information about the anchors is flooded through the network to determine the (multi-hop) distances between anchors and nodes. Second, each node calculates its position using the known positions and estimated ranges of the anchors, for example, by performing a lateration procedure (as with GPS systems). Third, nodes refine their positions by exchanging their position estimates and using the one-hop inter-node ranges. After these three stages, a subset of the nodes have location information that is considered "good".

Euclidean [3] uses basic geometric reasoning (triangles) to progress distance information from the anchors to the nodes in the network, and uses lateration to calculate the position estimates; no refinement is included in the algorithm. Euclidean's basic safety measure against inaccurate range information is to discard "impossible" triangles generated in phase 1. Unfortunately, this happens quite often, leaving many nodes in phase 2 without enough information to calculate their position (distances to at least 3 anchors are required). The end result is that Euclidean is only able to derive an accurate position for a small fraction of the nodes in the network.

Hop-Terrain [3], [4] avoids the range error problem to a large extent by using only topological information in phase 1. The distance to an anchor is determined by counting the number of hops to it, and multiplying that by an average-hop distance. Next, the node positions are estimated by means of a lateration procedure. In the refinement phase, Hop-Terrain switches to using the measured (inaccurate) ranges to neighbouring nodes. To avoid erroneous position estimates affecting neighbours too much, the refinement phase uses confidence values derived from the lateration procedure . Hop-terrain works well for a regular network topology in which nodes are evenly distributed. This however is not the case for the majority of mobile anchor scenarios, resulting in the algorithm becoming increasingly less accurate as the regularity assumption starts to break down.

Multilateration [5] proceeds by summing the distances along each multi-hop path in phase 1. To account for the accumulated inaccuracies it does not perform a lateration procedure, but instead uses each distance to specify a bounding box centred around the associated anchor, in which the node may be located. In phase 2, these bounding boxes are simply intersected and the position estimate is set to the centre of the intersection box, followed by a refinement procedure in phase 3. Multilateration's effectiveness with varying errors in range measurements will depend on the exact nature of the errors. If many of the measured distances are larger than the true distances, then Multilateration should be able to cope with the problem (as the true distance will still fall within the bounding box). In the case of underestimated ranges, however, Multilateration is forced to discard possible location information due to contradictions between ranges with varying errors.

B. Statistic-based Localisation

One technique that attempts to do more with inaccurate ranging information is Statistic-based Localisation. The initial work on this was performed by Sichitiu and Ramadurai [6]. It assumes that while the incoming range data has errors, these errors can be modelled with a probability distribution based on the incoming data (either from a sensor or the Radio Signal Strength Information (RSSI) from the radio [6]). This model can be worked out either from manufacturer-supplied data for the sensor providing range data, or from experimental data [8].

Given that a node has a series of ranges to anchors, and that for each range you have an error model, these models can be combined to calculate a "map" of the most likely locations for this node by calculating probabilities for each location at discrete intervals across the sensor grid. Figure 2 shows a visualisation of an example map, and Algorithm 1 has more details about how the maps are generated.

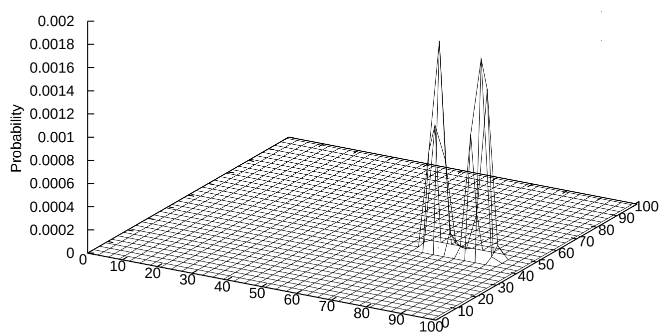

Fig. 2. Visualisation of a local probability map

Statistic-based localisation has two main problems however. Firstly, the large amount of computation required to create the maps and secondly, the requirement for the nodes to be at one-hop distances from the anchors (achieved in [6] by using a mobile anchor with a very dense path). This is required because [6] does not have a method for distributing anchor information received by one node to another, and so only anchor nodes can send their distance information.

IV. Refined Statistic-based Localisation

We made a number of changes to Statistic-based localisation. These make significant improvements to the basic algorithm, making it a more useful algorithm for applications with limited resources (i.e. most proposed sensor node applications). In this section, and in Algorithm 2, we detail the improved algorithm that we used in our experiments.

A. Bounding boxes

If a node has received a position estimate from an anchor then it knows it is in radio contact with that anchor, and so therefore it must be within radio range of that anchor. So, we can limit the space of possible locations for that node to a circle centred on the anchor's location with radius equal to the radio range. For practical purposes (significant speed improvements) we use a bounding box rather than a circle, with each side equal to 2*radio range, and the anchor in the centre (Figure 3a). (The basic concept of bounding boxes has previously been analysed in [9], but not in combination with statistic-based localisation.) This results in a larger region, but we still have the guarantee that all feasible locations for the node are located within the box, while keeping the box size to a minimum. This currently assumes a circular radio model, but for radios with non-circular transmission spaces, we can calculate the minimum box that contains the entire possible transmission space, and so be still able to use this methodology.

When a node receives location information from an additional anchor, it knows that it must be within the bounding boxes for

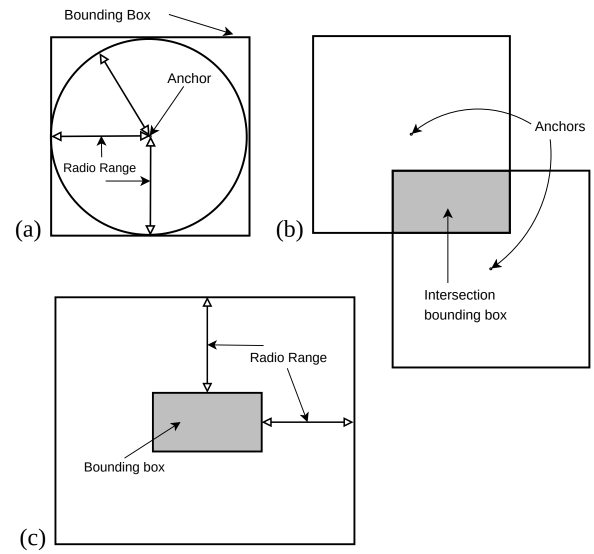

Fig. 3. Bounding Boxes

both anchors. Therefore, we can reduce the bounding box for the node to the intersection of both of these boxes (Figure 3b, and Algorithm 2, step 2). A bounding box is defined by two points, its Top-Left and Bottom-Right corners. Note that the probability visualisation in Figure 4 only shows a partial grid (as opposed to Figure 2 which shows basic Statistic-based localisation, and uses a complete grid). This partial grid is the section of the complete sensor grid corresponding to the bounding box for this particular node.

Experimental results for testing the reduction in the size of the calculated sensor grid, show an average reduction in the number of required calculations by a factor of 8 when we use bounding boxes. Also, with the additional optimisation of not doing calculations for the nodes with the largest bounding boxes, we could improve this result further. For example, by not performing any calculation for nodes with bounding boxes where $(width * height) > (3 * RadioRange)$, we reduce the overall calculation load by an additional factor of 3.

B. Limited broadcast

To get around the problem of needing anchors within one-hop of the sensor nodes, we perform a limited broadcast of calculated node location information - limited by only broadcasting if we exceed a minimum probability threshold for the quality of our location information (currently set in our implementation to 0.003). The node effectively acts as an additional "pseudo" anchor, but with two changes from normal anchors.

Firstly, location information is broadcast with a confidence value (gained from the local probability map), and the error model used by nodes receiving this information will be scaled accordingly, as shown in Algorithm 2, step 3a with the use of $Confidence_{anchor}$ in the generation of PDF_{rssi}. This confidence value is a weighting value for use in the statistical models i.e. a node with confidence 1.0 (an anchor) will have twice the effect of a node with confidence 0.5.

Secondly, with pseudo anchors, the bounding box is broadcast as well, and the box used by receiving nodes is not just a square centred on the node (as for anchors), but a rectangle equal to the bounding box size, plus radio range in each direction (Figure 3c). This is because the bounding box contains all locations the pseudo anchor could possibly be in, and so increasing it by the radio range creates a box in which nodes that can hear this pseudo anchor could possibly be located.

Algorithm 1 Statistic-based localisation [6]

1) Initially, the local probability "map" is set to a constant value across the entire sensor grid, as all locations are considered to be equally likely at the start of the algorithm.
   $PosEst(x,y) = c \quad \forall(x,y) \in [(x_{min}, x_{max}) \times (y_{min}, y_{max})]$
2) Incoming anchor information is processed as follows:
   a) The incoming anchor location is used to create a "constraint" function on the possible locations of the node
      $PDF_{rssi} = N \sim (EstimatedDistance_{anchor}, RadioRangingVariance)$
      $Constraint(x,y) = PDF_{rssi}(distance((x,y),(x_{anchor}, y_{anchor}))) \quad \forall(x,y) \in [(x_{min}, x_{max}) \times (y_{min}, y_{max})]$
   b) The node applies Bayesian inference to its current map to generate an improved map
      $NewPosEst(x,y) = \frac{OldPosEst(x,y) \times Constraint(x,y)}{\sum_{x_{min}}^{x_{max}} \sum_{y_{min}}^{y_{max}} OldPosEst(x,y) \times Constraint(x,y)} \quad \forall(x,y) \in [(x_{min}, x_{max}) \times (y_{min}, y_{max})]$
3) Finally, the weighted average of all of the data in the map is used to calculate the estimated position of this node
   $(\hat{x}, \hat{y}) = (\sum_{x_{min}}^{x_{max}} \sum_{y_{min}}^{y_{max}} x \times PosEst(x,y), \sum_{x_{min}}^{x_{max}} \sum_{y_{min}}^{y_{max}} y \times PosEst(x,y))$

Algorithm 2 Refined Statistic-based localisation

Abbreviations used here: TL = Top-Left corner of a bounding box, BR = Bottom-Right corner, R = Radio Range of the nodes

1) Initially, the bounding box for a node is set to $[(-\infty, \infty) \times (-\infty, \infty)]$.
2) As (pseudo-)anchor information comes in, the bounding box for this node is intersected with the existing bounding box (see Figure 3 for examples of bounding boxes, including a diagram of this step in Figure 3b)
   $NewBox(TL, BR) = [(Max(Anchor_{TL_x} - R, OldBox_{TL_x}),\ Max(Anchor_{TL_y} - R, OldBox_{TL_y})) \times$
   $(Min(Anchor_{BR_x} + R, OldBox_{BR_x}),\ Min(Anchor_{BR_y} + R, OldBox_{BR_y}))]$
3) Once information from at least two (pseudo-)anchors have been received, and the minimum waiting period since the last incoming anchor has passed, then we initialise the local map to a constant value
   $PosEst(x,y) = c \quad \forall(x,y) \in BoundingBox$
   and then each of the incoming (pseudo-)anchors that we have received so far is processed as follows:
   a) The incoming anchor information is used to create a "constraint" function on the possible locations of the node
      $PDF_{rssi} = N \sim (EstimatedDistance_{anchor}, RadioRangingVariance/Confidence_{Anchor})$
      $Constraint(x,y) = PDF_{RSSI}(distance((x,y),(x_{anchor}, y_{anchor}))) \quad \forall(x,y) \in BoundingBox$
   b) The node then multiplies each value in the map by the constraint function to generate an improved map
      $NewPosEst(x,y) = OldPosEst(x,y) \times Constraint(x,y) \quad \forall(x,y) \in BoundingBox$
4) The location on the map with the highest probability is determined (this is the most-likely location for this node)
   $(\hat{x}, \hat{y}) = maxarg\{PosEst(x,y) \mid (x,y) \in BoundingBox\}$
5) Finally, the map is normalised to provide an externally-usable probability value
   $NormConstant = \sum_{BoundingBox_{TL_x}}^{BoundingBox_{BR_x}} \sum_{BoundingBox_{TL_y}}^{BoundingBox_{BR_y}} PosEst(x,y)$
   $FinalPosEst(x,y) = PosEst(x,y)/NormConstant \quad \forall(x,y) \in BoundingBox$

This box will be larger than a box generated from an anchor node, because the location information is less accurate. However, this larger box may still be useful to other nodes in reducing their bounding boxes, and hence reducing the amount of computation that they need to perform.

Nodes that have position information, but do not exceed the probability threshold are considered "bad". These nodes have some position information, but either the information is insufficient, or it is of too low a quality to be fully usable. These do not broadcast their location information to other nodes.In our experiments comparing this multi-hop method with the original single-hop method, we see a similar average error in the locations of the good nodes, but a 38% average increase in the number of good nodes.

C. Symmetry problem

There are a number of situations where we will have multiple points that have equally high probabilities (or certainly very similar, and within the bounds of statistical error). One of the most likely instances of this problem is when the mobile anchor is travelling in a straight line. As the distributions of the broadcast anchors cross over equally on both sides of the line, a pair of possible good points will be created, each one equally far away from the line, but on opposite sides. Figure 4 is an example of this, showing the local probability map for a node with this particular problem.

The broadcasting of information from nodes with good location data reduces the symmetry problem, as this creates additional pseudo-anchors, allowing the possible locations for the node to be "pulled" in the direction of the correct point. However, in the event that the mobile anchor's path is a straight line, and that there are insufficient "good" nodes in the local area to broadcast pseudo-anchor information, then the problem still occurs. One solution is to avoid choosing straight paths for the mobile anchor - simple possibilities include curved paths, or using a "wobbly" path rather than a perfect straight line. Corners and curved sections on the mobile anchor's path reduces the chances of the symmetry problem considerably.

A node can determine whether or not it is likely to have multiple possible positions, based on its local probability map, by calculating the average of all of the anchor locations the node knows about (weighted according to their confidence values), and seeing how much each anchor's location differs from their average location in each separate axis. This allows the node to check whether its known anchors are mostly arranged along a straight line, or whether they have a more varied path. If there is a significantly greater total difference from the average point in one axis than another (indicating a mostly straight path), then the node will also test the other possible good points. These points can be found by taking the averaged anchor

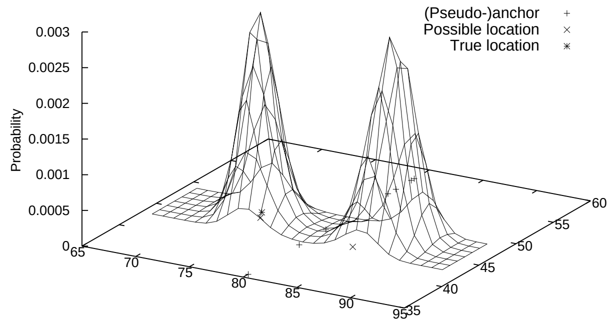

Fig. 4. Equal points

location, then looking at the points that are on the opposite side of the average point from the calculated most-likely location for this node. An average of the most-likely location and the other possible points (weighted according to their individual confidences) becomes the node's estimate of its true location.

If the sum of the confidences for the most-likely point and the best of the other candidate points, divided by a scaling factor, is above the standard threshold for transmission of the calculated location, then we transmit both locations. The scaling factor varies according to the degree of difference between the two confidences i.e. how good the second confidence is compared to the first.

$$ScalingFactor = 2 \times \frac{confidence(Best)+confidence(SecondBest)}{confidence(Best)}$$

If the node decides to transmit its current guesses, then the confidences for both points are transmitted, and the node is treated as two separate nodes by its neighbours, one at each of the two possible points, but each with a reduced confidence (compared to the calculated confidence for the point).

D. Heavy data-processing

One downside of statistic-based methods is the amount of data processing required to calculate the local maps. The bounding boxes reduce this significantly (a factor of 8), by eliminating many regions that this node can not be located at. For best results, there should be a waiting period for a short amount of time (e.g. 5 seconds) after the last piece of anchor information has been received, before calculating the local map, in order to work with the smallest possible bounding box. This will slow down the calculation of this node's location, but given that it is necessary to re-calculate the data if we receive another anchor, this can reduce the amount of redundant calculations significantly. The waiting period should be calibrated such that if we have not seen a new anchor for that amount of time, then we are unlikely to receive more anchor information in the near future. Good values for this would be at least as large as the interval between broadcasts of the mobile anchor.

The energy costs associated with statistic-based localisation are higher than for most localisation techniques, due to the large number of probability calculations required (Lateration being a notable exception, due to the use of matrix multiplications), but this additional cost is in most cases a one-off initialisation cost. Simulation results show an increase in processing time for refined statistic-based localisation over deterministic techniques ([3], [4], [5]) by approximately a factor of two; an increase from ~5 to ~10 seconds on a typical node CPU (Texas Instruments MSP430) and an average bounding box taken from experimental testing. This order of processing time is not an unacceptable start-up cost for a long running application, given the significant improvements in the derived location information. The probability calculations can also be performed by many nodes at the same time without additional costs, as opposed to other localisation techniques requiring large numbers of radio messages (which would exhibit increased numbers of packet collisions if several nodes are transmitting at the same time). Refined Statistic-based localisation has been deliberately optimised towards reduced radio traffic with this aim in mind.

V. RESULTS

Using the Positif simulation framework for localisation algorithm testing [7], we have performed a series of comparison tests between our refined statistic-based method, and three deterministic localisation techniques (Euclidean [3], Hop-Terrain [4] and Multilateration [5]), using a mobile anchor scenario in all cases, and with a variety of ranging errors between nodes.

In each case, all of the algorithms have been tested with the same set of data, and each result is the average of 10 runs of the simulation with varying random-number seeds. The ranging error is modelled as a Gaussian distribution, with the mean as the actual range, and the range variance as a percentage of the radio range. The internal model of the refined statistic algorithm in all cases is set to a Gaussian distribution with the mean as the incoming range information, and the estimated range variance set to 20% of the radio range. In all scenarios, there are 226 sensor nodes randomly placed, with a uniform distribution, within a square area. The mobile anchor is modelled as a formation of 111 "virtual" anchors within this sensor grid. The grid has a size of 100x100, and the radio range is set to 14 providing the nodes with an average connectivity of 19.

For the sake of brevity, we have only shown one of the mobile anchor scenarios we have tested - a generic "square" formation, with a mobile anchor moving along a square path around the centre of the grid. The straight lines of this topology have been deliberately chosen to cause difficulties to the refined statistic algorithm.

Figure 5 is a visualisation of the individual node locations for an example experiment performed using the refined statistic algorithm. The nodes marked with a "•" are anchor nodes, the others are sensor nodes; the ones marked with a "*" are good nodes, nodes marked with a "+" are bad nodes, and the "△" nodes have no position data at all. Lines attached to nodes show the path from a node's true position to where it thinks it is. The longer the line, the less accurate the estimated position. Note that, in general, the Statistic algorithm does an accurate job of classifying the nodes into good and bad ones, but occasionally generates both false positives (good nodes with long lines) and false negatives (bad nodes with short lines). These anomalies generally occur outside the area directly covered by the mobile anchor. Since the node classification is largely correct, applications should be able to exploit that knowledge.

In Figure 6, we show the average accuracy of the good nodes for all of the algorithms. The Statistic algorithm provides information about bad nodes (as opposed to the other algorithms, which only give a node as either "good" or un-positioned), so we also show the accuracy for a weighted average of both good and bad nodes. Figure 7 shows the average percentages of positioned nodes in each of these cases. Note the poor coverage (generally less than 50% of the nodes obtain a position estimate), as a result of the non-uniform anchor distribution.

In most cases the Statistic algorithm has the lowest percentage error in its "good" positions. Euclidean only outperforms it under ideal circumstances (i.e. no range errors); in all other cases (error variance $\geq$ 5%) the Statistic algorithm provides (much) more accurate position estimates. In general, localisation algorithms can trade-off

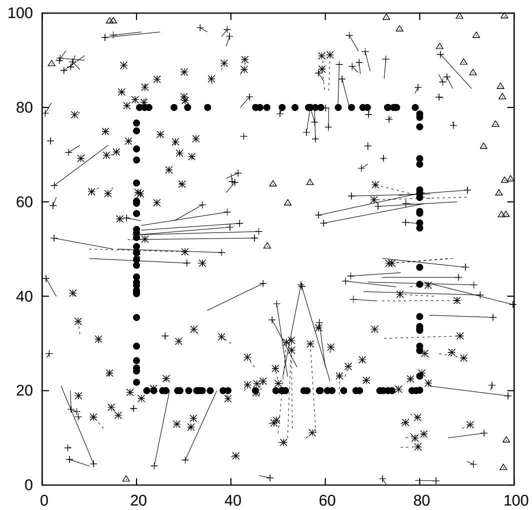

Fig. 5. Square topology, 20% range error variance

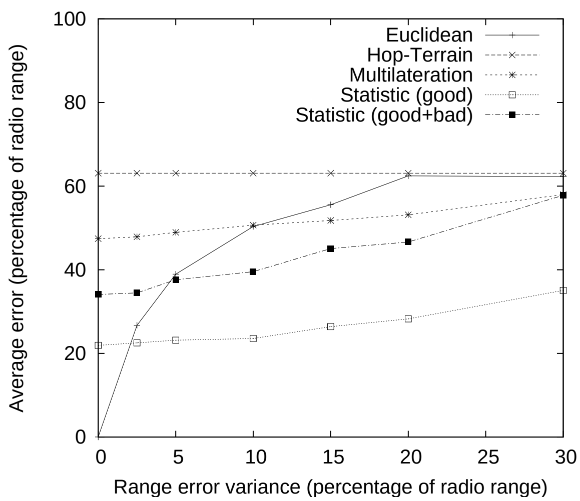

Fig. 6. Position errors for square topology

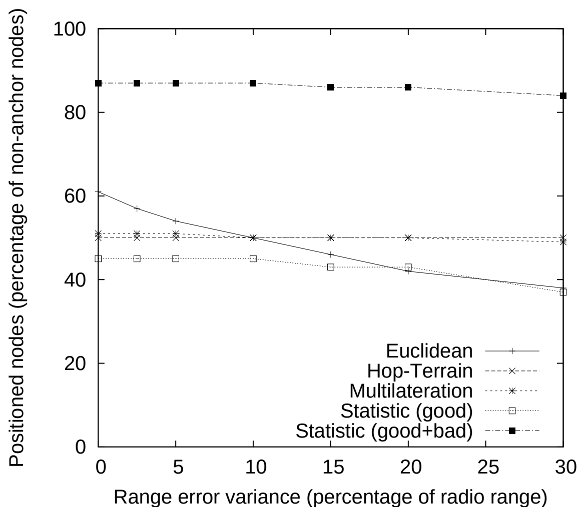

Fig. 7. Positioned nodes for square topology

accuracy for coverage [7]. The Statistic algorithm, however, combines high accuracy with reasonable coverage. For low error variances, the Statistic algorithm has similar numbers of good nodes as the Hop-Terrain and Multilateration algorithms, and only at higher values the statistic algorithm classifies more nodes as being bad. The combination of the Statistic good and bad nodes however, gives a comparable level of error to the other algorithms, but with up to a doubled number of positioned nodes.

VI. Conclusions and Future Work

We presented here an approach that can provide good location information, even with non-uniform anchor distributions and considerable inaccuracies in the incoming ranging data. Refined Statistic-based localisation provides a good solution to the problem of localisation even in small, resource-limited sensor networks. We showed that we can calculate accurate position data for a high percentage of the sensor nodes in a network. We have improved both the quality and quantity of positioned nodes in sensor networks, both versus the earlier Statistic-based and deterministic localisation methods.

All of this has been tested using mobile anchor scenarios, which we have shown to be a realistic and usable method for the distribution of anchor data, as well as a cost-effective one - both in terms of energy costs for the sensor nodes of the network, and in terms of the necessary hardware required to create the sensor network. Getting rid of the errors in sensor measurements is hard, but that is the price of gathering data from the real world. With statistical approaches, we have shown that it is possible to work around these errors, and derive good location information. Statistical approaches are somewhat more computationally expensive, but given the significant improvements in the location information, and that the computational expense results in a reduced level of required radio traffic during the localisation process (which increases the capability of other nearby nodes to do radio-dependent work efficiently during the localisation process), we believe that the trade-offs are worth it.

In the future, we hope to expand on our work here to attempt to further improve the location information that can be gathered, by integrating more accurate models of various ranging sensors, and also testing to see whether a combined model from several sensors may improve accuracy.

References

[1] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, pp. 57–66, Aug. 2001.

[2] N. Bulusu, J. Heidemann, V. Bychkovskiy, and D. Estrin, "Density-adaptive beacon placement algorithms for localization in ad hoc wireless networks," in *IEEE Infocom 2002*, (New York, NY), June 2002.

[3] D. Niculescu and B. Nath, "Ad-hoc positioning system," in *IEEE GlobeCom*, pp. 2926–2931, Nov. 2001.

[4] C. Savarese, K. Langendoen, and J. Rabaey, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *USENIX technical annual conference*, (Monterey, CA), pp. 317–328, June 2002.

[5] A. Savvides, H. Park, and M. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *First ACM Int. Workshop on Wireless Sensor Networks and Application (WSNA)*, (Atlanta, GA), pp. 112–121, Sept. 2002.

[6] M. Sichitiu and V. Ramadurai, "Localization of Wireless Sensor Networks with a Mobile Beacon," Tech. Rep. TR-03/06, Center for Advances Computing and Communications (CACC), Raleigh, NC, July 2003.

[7] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: A quantitative comparison," *Computer Networks, special issue on Wireless Sensor Networks*, no. 43, pp. 500–518, 2003.

[8] K. Whitehouse and D. Culler, "Calibration as parameter estimation in sensor networks," in *First ACM Int. Workshop on Wireless Sensor Networks and Application (WSNA)*, (Atlanta, GA), pp. 59–67, Sept. 2002.

[9] S. Simic and S. Sastry, "Distributed localization in wireless ad hoc networks," Tech. Rep. UCB/ERL M02/26, UC Berkeley, 2002.