MSP: Multi-Sequence Positioning of Wireless Sensor Nodes[*]

Ziguo Zhong
Computer Science and Engineering
University of Minnesota
zhong@cs.umn.edu

Tian He
Computer Science and Engineering
University of Minnesota
tianhe@cs.umn.edu

Abstract

Wireless Sensor Networks have been proposed for use in many location-dependent applications. Most of these need to identify the locations of wireless sensor nodes, a challenging task because of the severe constraints on cost, energy and effective range of sensor devices. To overcome limitations in existing solutions, we present a Multi-Sequence Positioning (MSP) method for large-scale stationary sensor node localization in outdoor environments. The novel idea behind MSP is to reconstruct and estimate two-dimensional location information for each sensor node by processing multiple one-dimensional *node sequences*, easily obtained through loosely guided event distribution. Starting from a basic MSP design, we propose four optimizations, which work together to increase the localization accuracy. We address several interesting issues, such as incomplete (partial) node sequences and sequence flip, found in the *Mirage* test-bed we built. We have evaluated the MSP system through theoretical analysis, extensive simulation as well as two physical systems (an indoor version with 46 MICAz motes and an outdoor version with 20 MICAz motes). This evaluation demonstrates that MSP can achieve an accuracy within one foot, requiring neither additional costly hardware on sensor nodes nor precise event distribution. It also provides a nice tradeoff between physical cost (anchors) and soft cost (events), while maintaining localization accuracy.

Categories and Subject Descriptors

C.2.4 [**Computer Communications Networks**]: Distributed Systems

General Terms

Algorithms, Measurement, Design, Performance, Experimentation

Keywords

Wireless Sensor Networks, Localization, Node Sequence Processing

1 Introduction

Although Wireless Sensor Networks (WSN) have shown promising prospects in various applications [5], researchers still face several challenges for massive deployment of such networks. One of these is to identify the location of individual sensor nodes in outdoor environments. Because of unpredictable flow dynamics in airborne scenarios, it is not currently feasible to localize sensor nodes during massive UVA-based deployment. On the other hand, geometric information is indispensable in these networks, since users need to know where events of interest occur (e.g., the location of intruders or of a bomb explosion).

Previous research on node localization falls into two categories: range-based approaches and range-free approaches. Range-based approaches [13, 17, 19, 24] compute per-node location information iteratively or recursively based on measured distances among target nodes and a few anchors which precisely know their locations. These approaches generally require costly hardware (e.g., GPS) and have limited effective range due to energy constraints (e.g., ultrasound-based TDOA [3, 17]). Although range-based solutions can be suitably used in small-scale indoor environments, they are considered less cost-effective for large-scale deployments. On the other hand, range-free approaches [4, 8, 10, 13, 14, 15] do not require accurate distance measurements, but localize the node based on network connectivity (proximity) information. Unfortunately, since wireless connectivity is highly influenced by the environment and hardware calibration, existing solutions fail to deliver encouraging empirical results, or require substantial survey [2] and calibration [24] on a case-by-case basis.

Realizing the impracticality of existing solutions for the large-scale outdoor environment, researchers have recently proposed solutions (e.g., Spotlight [20] and Lighthouse [18]) for sensor node localization using the spatiotemporal correlation of controlled events (i.e., inferring nodes' locations based on the detection time of controlled events). These solutions demonstrate that long range and high accuracy localization can be achieved simultaneously with little additional cost at sensor nodes. These benefits, however, come along with an implicit assumption that the controlled events can be precisely distributed to a specified location at a specified time. We argue that precise event distribution is difficult to achieve, especially at large scale when terrain is uneven, the event distribution device is not well calibrated and its position is difficult to maintain (e.g., the helicopter-mounted scenario in [20]).

To address these limitations in current approaches, in this paper we present a multi-sequence positioning (MSP) method

---

[*] This research was supported in part by NSF grants CNS-0626609 and CNS-0626614.

SenSys'07, November 6–9, 2007, Sydney, Australia.

for large-scale stationary sensor node localization, in deployments where an event source has line-of-sight to all sensors. The novel idea behind MSP is to estimate each sensor node's two-dimensional location by processing multiple easy-to-get one-dimensional node sequences (e.g., event detection order) obtained through loosely-guided event distribution.

This design offers several benefits. First, compared to a range-based approach, MSP does not require additional costly hardware. It works using sensors typically used by sensor network applications, such as light and acoustic sensors, both of which we specifically consider in this work. Second, compared to a range-free approach, MSP needs only a small number of anchors (theoretically, as few as two), so high accuracy can be achieved economically by introducing more events instead of more anchors. And third, compared to Spotlight, MSP does not require precise and sophisticated event distribution, an advantage that significantly simplifies the system design and reduces calibration cost.

This paper offers the following additional intellectual contributions:

- We are the first to localize sensor nodes using the concept of *node sequence*, an ordered list of sensor nodes, sorted by the detection time of a disseminated event. We demonstrate that making full use of the information embedded in one-dimensional node sequences can significantly improve localization accuracy. Interestingly, we discover that repeated reprocessing of one-dimensional node sequences can further increase localization accuracy.
- We propose a distribution-based location estimation strategy that obtains the final location of sensor nodes using the marginal probability of joint distribution among adjacent nodes within the sequence. This new algorithm outperforms the widely adopted Centroid estimation [4, 8].
- To the best of our knowledge, this is the first work to improve the localization accuracy of nodes by adaptive events. The generation of later events is guided by localization results from previous events.
- We evaluate line-based MSP on our new *Mirage* test-bed, and wave-based MSP in outdoor environments. Through system implementation, we discover and address several interesting issues such as partial sequence and sequence flips. To reveal MSP performance at scale, we provide analytic results as well as a complete simulation study. All the simulation and implementation code is available online at *http://www.cs.umn.edu/∼zhong/MSP*.

The rest of the paper is organized as follows. Section 2 briefly surveys the related work. Section 3 presents an overview of the MSP localization system. In sections 4 and 5, basic MSP and four advanced processing methods are introduced. Section 6 describes how MSP can be applied in a wave propagation scenario. Section 7 discusses several implementation issues. Section 8 presents simulation results, and Section 9 reports an evaluation of MSP on the *Mirage* test-bed and an outdoor test-bed. Section 10 concludes the paper.

2 Related Work

Many methods have been proposed to localize wireless sensor devices in the open air. Most of these can be classified into two categories: range-based and range-free localization. Range-based localization systems, such as GPS [23], Cricket [17], AHLoS [19], AOA [16], Robust Quadrilaterals [13] and Sweeps [7], are based on fine-grained point-to-point distance estimation or angle estimation to identify per-node location. Constraints on the cost, energy and hardware footprint of each sensor node make these range-based methods undesirable for massive outdoor deployment. In addition, ranging signals generated by sensor nodes have a very limited effective range because of energy and form factor concerns. For example, ultrasound signals usually effectively propagate 20-30 feet using an on-board transmitter [17]. Consequently, these range-based solutions require an undesirably high deployment density. Although the received signal strength indicator (RSSI) related [2, 24] methods were once considered an ideal low-cost solution, the irregularity of radio propagation [26] seriously limits the accuracy of such systems. The recently proposed RIPS localization system [11] superimposes two RF waves together, creating a low-frequency envelope that can be accurately measured. This ranging technique performs very well as long as antennas are well oriented and environmental factors such as multi-path effects and background noise are sufficiently addressed.

Range-free methods don't need to estimate or measure accurate distances or angles. Instead, anchors or controlled-event distributions are used for node localization. Range-free methods can be generally classified into two types: anchor-based and anchor-free solutions.

- For anchor-based solutions such as Centroid [4], APIT [8], SeRLoc [10], Gradient [13] , and APS [15], the main idea is that the location of each node is estimated based on the known locations of the anchor nodes. Different anchor combinations narrow the areas in which the target nodes can possibly be located. Anchor-based solutions normally require a high density of anchor nodes so as to achieve good accuracy. In practice, it is desirable to have as few anchor nodes as possible so as to lower the system cost.
- Anchor-free solutions require no anchor nodes. Instead, external event generators and data processing platforms are used. The main idea is to correlate the event detection time at a sensor node with the known space-time relationship of controlled events at the generator so that detection time-stamps can be mapped into the locations of sensors. Spotlight [20] and Lighthouse [18] work in this fashion. In Spotlight [20], the event distribution needs to be precise in both time and space. Precise event distribution is difficult to achieve without careful calibration, especially when the event-generating devices require certain mechanical maneuvers (e.g., the telescope mount used in Spotlight). All these increase system cost and reduce localization speed. StarDust [21], which works much faster, uses label relaxation algorithms to match *light spots* reflected by corner-cube retro-reflectors (CCR) with sensor nodes using various constraints. Label relaxation algorithms converge only when a sufficient number of robust constraints are obtained. Due to the environmental impact on RF connectivity constraints, however, StarDust is less accurate than Spotlight.

In this paper, we propose a balanced solution that avoids the limitations of both anchor-based and anchor-free solutions. Unlike anchor-based solutions [4, 8], MSP allows a flexible tradeoff between the physical cost (anchor nodes) with the soft

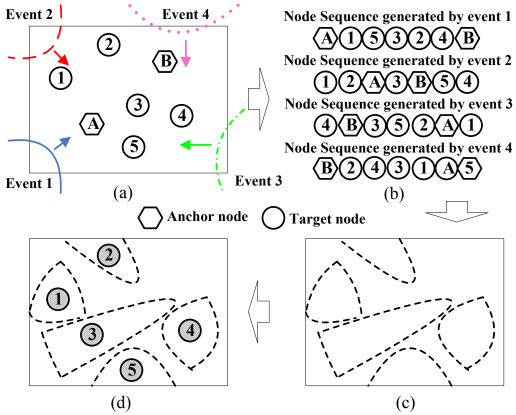


Figure 1. The MSP System Overview

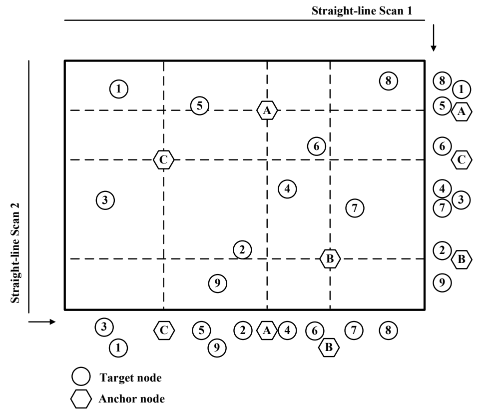


Figure 2. Obtaining Multiple Node Sequences

cost (localization events). MSP uses only a small number of anchors (theoretically, as few as two). Unlike anchor-free solutions, MSP doesn't need to maintain rigid time-space relationships while distributing events, which makes system design simpler, more flexible and more robust to calibration errors.

3 System Overview

MSP works by extracting relative location information from multiple simple one-dimensional orderings of nodes. Figure 1(a) shows a layout of a sensor network with anchor nodes and target nodes. Target nodes are defined as the nodes to be localized. Briefly, the MSP system works as follows. First, events are generated one at a time in the network area (e.g., ultrasound propagations from different locations, laser scans with diverse angles). As each event propagates, as shown in Figure 1(a), each node detects it at some particular time instance. For a single event, we call the ordering of nodes, which is based on the sequential detection of the event, a *node sequence*. Each node sequence includes both the targets and the anchors as shown in Figure 1(b). Second, a multi-sequence processing algorithm helps to narrow the possible location of each node to a small area (Figure 1(c)). Finally, a distribution-based estimation method estimates the exact location of each sensor node, as shown in Figure 1(d).

Figure 1 shows that the node sequences can be obtained much more economically than accurate pair-wise distance measurements between target nodes and anchor nodes via ranging methods. In addition, this system does not require a rigid time-space relationship for the localization events, which is critical but hard to achieve in controlled event distribution scenarios (e.g., Spotlight [20]).

For the sake of clarity in presentation, we present our system in two cases:

- **Ideal Case**, in which all the node sequences obtained from the network are complete and correct, and nodes are time-synchronized [12, 9].
- **Realistic Deployment**, in which (i) node sequences can be partial (incomplete), (ii) elements in sequences could flip (i.e., the order obtained is reversed from reality), and (iii) nodes are not time-synchronized.

To introduce the MSP algorithm, we first consider a simple straight-line scan scenario. Then, we describe how to implement straight-line scans as well as other event types, such as sound wave propagation.

4 Basic MSP

Let us consider a sensor network with N target nodes and M anchor nodes randomly deployed in an area of size S. The top-level idea for basic MSP is to split the whole sensor network area into small pieces by processing node sequences. Because the exact locations of all the anchors in a node sequence are known, all the nodes in this sequence can be divided into $O(M+1)$ parts in the area.

In Figure 2, we use numbered circles to denote target nodes and numbered hexagons to denote anchor nodes. Basic MSP uses two straight lines to scan the area from different directions, treating each scan as an event. All the nodes react to the event sequentially generating two node sequences. For vertical scan 1, the node sequence is (8,1,5,A,6,C,4,3,7,2,B,9), as shown outside the right boundary of the area in Figure 2; for horizontal scan 2, the node sequence is (3,1,C,5,9,2,A,4,6,B,7,8), as shown under the bottom boundary of the area in Figure 2.

Since the locations of the anchor nodes are available, the anchor nodes in the two node sequences actually split the area vertically and horizontally into 16 parts, as shown in Figure 2. To extend this process, suppose we have M anchor nodes and perform d scans from different angles, obtaining d node sequences and dividing the area into many small parts. Obviously, the number of parts is a function of the number of anchors M, the number of scans d, the anchors' location as well as the slop k for each scan line. According to the pie-cutting theorem [22], the area can be divided into $O(M^2d^2)$ parts. When M and d are appropriately large, the polygon for each target node may become sufficiently small so that accurate estimation can be achieved. We emphasize that accuracy is affected not only by the number of anchors M, but also by the number of events d. In other words, MSP provides a tradeoff between the physical cost of anchors and the soft cost of events.

Algorithm 1 depicts the computing architecture of basic MSP. Each node sequence is processed within line 1 to 8. For each node, *GetBoundaries()* in line 5 searches for the predecessor and successor anchors in the sequence so as to determine the boundaries of this node. Then in line 6 *UpdateMap()* shrinks the location area of this node according to the newly obtained boundaries. After processing all sequences, Centroid Estimation (line 11) set the center of gravity of the final polygon as the estimated location of the target node.

Basic MSP only makes use of the order information between a target node and the anchor nodes in each sequence. Actually, we can extract much more location information from

Algorithm 1 Basic MSP Process

```
Output: The estimated location of each node.
repeat
    GetOneUnprocessedSeqence();
    repeat
        GetOneNodeFromSequenceInOrder();
        GetBoundaries();
        UpdateMap();
    until All the target nodes are updated;
until All the node sequences are processed;
repeat
    GetOneUnestimatedNode();
    CentroidEstimation();
until All the target nodes are estimated;
```

each sequence. Section 5 will introduce advanced MSP, in which four novel optimizations are proposed to improve the performance of MSP significantly.

5 Advanced MSP

Four improvements to basic MSP are proposed in this section. The first three improvements do not need additional sensing and communication in the networks but require only slightly more off-line computation. The objective of all these improvements is to *make full use* of the information embedded in the node sequences. The results we have obtained empirically indicate that the implementation of the first two methods can dramatically reduce the localization error, and that the third and fourth methods are helpful for some system deployments.

5.1 Sequence-Based MSP

As shown in Figure 2, each scan line and M anchors, splits the whole area into $M+1$ parts. Each target node falls into one polygon shaped by scan lines. We noted that in basic MSP, only the anchors are used to narrow down the polygon of each target node, but actually there is more information in the node sequence that we can made use of.

Let's first look at a simple example shown in Figure 3. The previous scans narrow the locations of target node 1 and node 2 into two dashed rectangles shown in the left part of Figure 3. Then a new scan generates a new sequence (1, 2). With knowledge of the scan's direction, it is easy to tell that node 1 is located to the left of node 2. Thus, we can further narrow the location area of node 2 by eliminating the shaded part of node 2's rectangle. This is because node 2 is located on the right of node 1 while the shaded area is outside the lower boundary of node 1. Similarly, the location area of node 1 can be narrowed by eliminating the shaded part out of node 2's right boundary. We call this procedure *sequence-based MSP* which means that the whole node sequence needs to be processed node by node in order. Specifically, sequence-based MSP follows this exact processing rule:

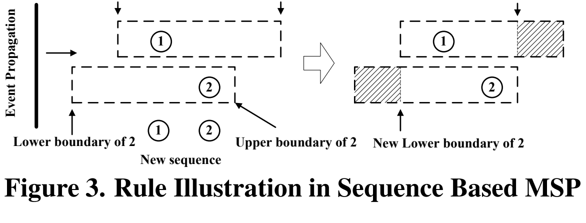


Figure 3. Rule Illustration in Sequence Based MSP

Algorithm 2 Sequence-Based MSP Process

```
Output: The estimated location of each node.
 1: repeat
 2:     GetOneUnprocessedSeqence();
 3:     repeat
 4:         GetOneNodeByIncreasingOrder();
 5:         ComputeLowbound();
 6:         UpdateMap();
 7:     until The last target node in the sequence;
 8:     repeat
 9:         GetOneNodeByDecreasingOrder();
10:         ComputeUpbound();
11:         UpdateMap();
12:     until The last target node in the sequence;
13: until All the node sequences are processed;
14: repeat
15:     GetOneUnestimatedNode();
16:     CentroidEstimation();
17: until All the target nodes are estimated;
```

Elimination Rule: *Along a scanning direction, the lower boundary of the successor's area must be equal to or larger than the lower boundary of the predecessor's area, and the upper boundary of the predecessor's area must be equal to or smaller than the upper boundary of the successor's area.*

In the case of Figure 3, node 2 is the successor of node 1, and node 1 is the predecessor of node 2. According to the elimination rule, node 2's lower boundary cannot be smaller than that of node 1 and node 1's upper boundary cannot exceed node 2's upper boundary.

Algorithm 2 illustrates the pseudo code of sequence-based MSP. Each node sequence is processed within line 3 to 13. The sequence processing contains two steps:

Step 1 (line 3 to 7): Compute and modify the lower boundary for each target node by increasing order in the node sequence. Each node's lower boundary is determined by the lower boundary of its predecessor node in the sequence, thus the processing must start from the first node in the sequence and by increasing order. Then update the map according to the new lower boundary.

Step 2 (line 8 to 12): Compute and modify the upper boundary for each node by decreasing order in the node sequence. Each node's upper boundary is determined by the upper boundary of its successor node in the sequence, thus the processing must start from the last node in the sequence and by decreasing order. Then update the map according to the new upper boundary.

After processing all the sequences, for each node, a polygon bounding its possible location has been found. Then, center-of-gravity-based estimation is applied to compute the exact location of each node (line 14 to 17).

An example of this process is shown in Figure 4. The third scan generates the node sequence (B,9,2,7,4,6,3,8,C,A,5,1). In addition to the anchor split lines, because nodes 4 and 7 come after node 2 in the sequence, node 4 and 7's polygons could be narrowed according to node 2's lower boundary (the lower right-shaded area); similarly, the shaded area in node 2's rectangle could be eliminated since this part is beyond node 7's upper boundary indicated by the dotted line. Similar eliminating can be performed for node 3 as shown in the figure.

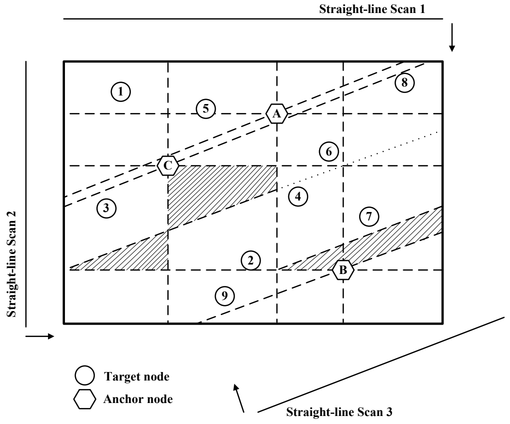


Figure 4. Sequence-Based MSP Example

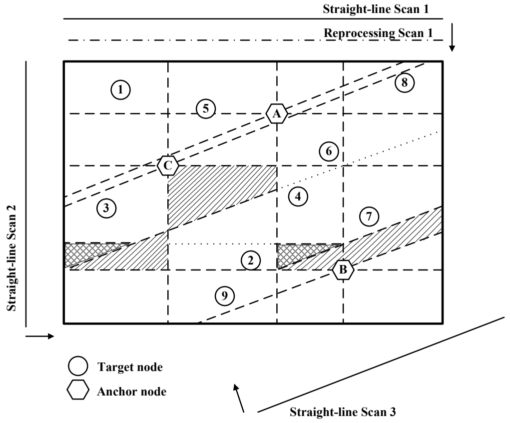


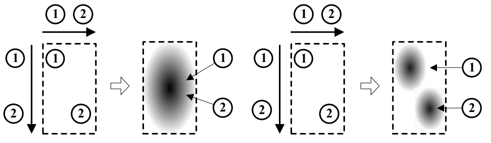
Figure 5. Iterative MSP: Reprocessing Scan 1

From above, we can see that the sequence-based MSP makes use of the information embedded in every sequential node pair in the node sequence. The polygon boundaries of the target nodes obtained in prior could be used to further split other target nodes' areas. Our evaluation in Sections 8 and 9 shows that sequence-based MSP considerably enhances system accuracy.

5.2 Iterative MSP

Sequence-based MSP is preferable to basic MSP because it extracts more information from the node sequence. In fact, further useful information still remains! In sequence-based MSP, a sequence processed later benefits from information produced by previously processed sequences (e.g., the third scan in Figure 5). However, the first several sequences can hardly benefit from other scans in this way. Inspired by this phenomenon, we propose iterative MSP. The basic idea of iterative MSP is to process all the sequences iteratively several times so that the processing of each single sequence can benefit from the results of other sequences.

To illustrate the idea more clearly, Figure 4 shows the results of three scans that have provided three sequences. Now if we process the sequence (8,1,5,A,6,C,4,3,7,2,B,9) obtained from scan 1 again, we can make progress, as shown in Figure 5. The reprocessing of the node sequence 1 provides information in the way an additional vertical scan would. From sequence-based MSP, we know that the upper boundaries of nodes 3 and 4 along the scan direction must not extend beyond the upper boundary of node 7, therefore the grid parts can be eliminated for the nodes 3 and node 4, respectively, as shown in Figure 5. From this example, we can see that iterative processing of the sequence could help further shrink the polygon of each target node, and thus enhance the accuracy of the system.



(a) Central of Gravity (b) Joint Distribution

Figure 6. Example of Joint Distribution Estimation

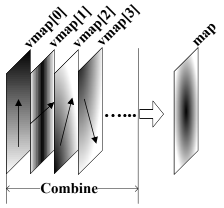


Figure 7. Idea of DBE MSP for Each Node

The implementation of iterative MSP is straightforward: process all the sequences multiple times using sequence-based MSP. Like sequence-based MSP, iterative MSP introduces *no additional event cost*. In other words, reprocessing does not actually repeat the scan physically. Evaluation results in Section 8 will show that iterative MSP contributes noticeably to a lower localization error. Empirical results show that after 5 iterations, improvements become less significant. In summary, iterative processing can achieve better performance with only a small computation overhead.

5.3 Distribution-Based Estimation

After determining the location area polygon for each node, estimation is needed for a final decision. Previous research mostly applied the Center of Gravity (COG) method [4] [8] [10] which minimizes average error. If every node is independent of all others, COG is the statistically best solution. In MSP, however, each node may not be independent. For example, two neighboring nodes in a certain sequence could have overlapping polygon areas. In this case, if the marginal probability of joint distribution is used for estimation, better statistical results are achieved.

Figure 6 shows an example in which node 1 and node 2 are located in the same polygon. If COG is used, both nodes are localized at the same position (Figure 6(a)). However, the node sequences obtained from two scans indicate that node 1 should be to the left of and above node 2, as shown in Figure 6(b).

The high-level idea of distribution-based estimation proposed for MSP, which we call **DBE MSP**, is illustrated in Figure 7. The distributions of each node under the *i*th scan (for the *i*th node sequence) are estimated in *node.vmap[i]*, which is a data structure for remembering the marginal distribution over scan *i*. Then all the *vmap*s are combined to get a single map and weighted estimation is used to obtain the final location.

For each scan, all the nodes are sorted according to the *gap*, which is the diameter of the polygon along the direction of the scan, to produce a second, gap-based node sequence. Then, the estimation starts from the node with the smallest gap. This is because it is statistically more accurate to assume a uniform distribution of the node with smaller gap. For each node processed in order from the gap-based node sequence, either if

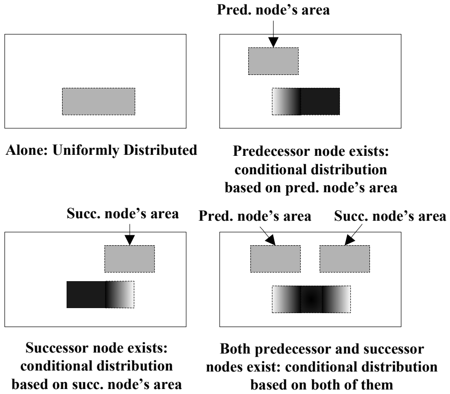


Figure 8. Four Cases in DBE Process

no neighbor node in the original event-based node sequence shares an overlapping area, or if the neighbors have not been processed due to bigger gaps, a uniform distribution *Uniform()* is applied to this *isolated* node (the *Alone* case in Figure 8). If the distribution of its neighbors sharing overlapped areas has been processed, we calculate the joint distribution for the node. As shown in Figure 8, there are three possible cases depending on whether the distribution of the overlapping predecessor and/or successor nodes have/has already been estimated.

The estimation's strategy of starting from the most accurate node (*smallest gap* node) reduces the problem of estimation error propagation. The results in the evaluation section indicate that applying distribution-based estimation could give statistically better results.

5.4 Adaptive MSP

So far, all the enhancements to basic MSP focus on improving the multi-sequence processing algorithm given *a fixed set of scan directions*. All these enhancements require only more computing time without any overhead to the sensor nodes. Obviously, it is possible to have some choice and optimization on how events are generated. For example, in military situations, artillery or rocket-launched mini-ultrasound bombs can be used for event generation at some selected locations. In adaptive MSP, we carefully generate each new localization event so as to maximize the contribution of the new event to the refinement of localization, based on feedback from previous events.

Figure 9 depicts the basic architecture of adaptive MSP. Through previous localization events, the whole map has been partitioned into many small location areas. The idea of adaptive MSP is to generate the next localization event to achieve *best-effort elimination*, which ideally could shrink the location area of individual node as much as possible.

We use a *weighted voting mechanism* to evaluate candidate localization events. Every node wants the next event to split its area evenly, which would shrink the area fast. Therefore, every node votes for the parameters of the next event (e.g., the scan angle k of the straight-line scan). Since the area map is maintained centrally, the vote is *virtually* done and there is no need for the real sensor nodes to participate in it. After gathering all the voting results, the event parameters with the most votes win the election. There are two factors that determine the weight of each vote:

- The vote for each candidate event is weighted according to the diameter D of the node's location area. Nodes with bigger location areas speak louder in the voting, because overall system error is reduced mostly by splitting the larger areas.

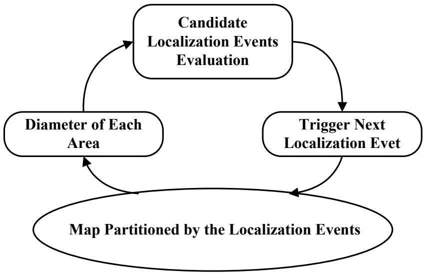


Figure 9. Basic Architecture of Adaptive MSP

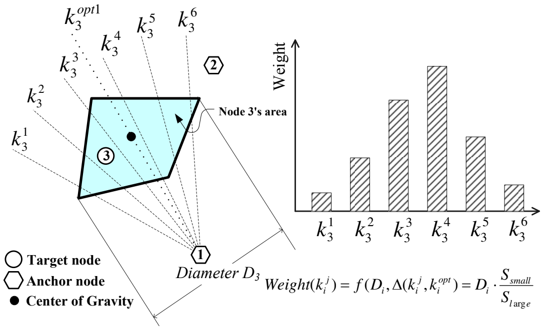


Figure 10. Candidate Slops for Node 3 at Anchor 1

- The vote for each candidate event is also weighted according to its *elimination efficiency* for a location area, which is defined as how equally in size (or in diameter) an event can cut an area. In other words, an optimal scan event cuts an area in the middle, since this cut shrinks the area quickly and thus reduces localization uncertainty quickly.

Combining the above two aspects, the weight for each vote is computed according to the following equation (1):

$$Weight(k_i^j) = f(D_i, \triangle(k_i^j, k_i^{opt})) \tag{1}$$

k_i^j is node i's jth supporting parameter for next event generation; D_i is diameter of node i's location area; $\triangle(k_i^j, k_i^{opt})$ is the *distance* between k_i^j and the optimal parameter k_i^{opt} for node i, which should be defined to fit the specific application.

Figure 10 presents an example for node 1's voting for the slopes of the next straight-line scan. In the system, there are a fixed number of candidate slopes for each scan (e.g., $k_1, k_2, k_3, k_4 ...$). The location area of target node 3 is shown in the figure. The candidate events $k_3^1, k_3^2, k_3^3, k_3^4, k_3^5, k_3^6$ are evaluated according to their effectiveness compared to the optimal ideal event which is shown as a dotted line with appropriate weights computed according to equation (1). For this specific example, as is illustrated in the right part of Figure 10, $f(D_i, \triangle(k_i^j, k_i^{opt}))$ is defined as the following equation (2):

$$Weight(k_i^j) = f(D_i, \triangle(k_i^j, k_i^{opt})) = D_i \cdot \frac{S_{small}}{S_{large}} \tag{2}$$

S_{small} and S_{large} are the sizes of the smaller part and larger part of the area cut by the candidate line respectively. In this case, node 3 votes 0 for the candidate lines that do not cross its area since $S_{small} = 0$.

We show later that adaptive MSP improves localization accuracy in WSNs with irregularly shaped deployment areas.

5.5 Overhead and MSP Complexity Analysis

This section provides a complexity analysis of the MSP design. We emphasize that MSP adopts an asymmetric design in which sensor nodes need only to detect and report the events. They are blissfully oblivious to the processing methods proposed in previous sections. In this section, we analyze the computational cost on the node sequence processing side, where resources are plentiful.

According to Algorithm 1, the computational complexity of Basic MSP is $O(d \cdot N \cdot S)$, and the storage space required is $O(N \cdot S)$, where d is the number of events, N is the number of target nodes, and S is the area size.

According to Algorithm 2, the computational complexity of both sequence-based MSP and iterative MSP is $O(c \cdot d \cdot N \cdot S)$, where c is the number of iterations and $c = 1$ for sequence-based MSP, and the storage space required is $O(N \cdot S)$. Both the computational complexity and storage space are equal within a constant factor to those of basic MSP.

The computational complexity of the distribution-based estimation (DBE MSP) is greater. The major overhead comes from the computation of joint distributions when both predecessor and successor nodes exit. In order to compute the marginal probability, MSP needs to enumerate the locations of the predecessor node and the successor node. For example, if node A has predecessor node B and successor node C, then the marginal probability $P_A(x,y)$ of node A's being at location (x,y) is:

$$P_A(x,y) = \sum_i \sum_j \sum_m \sum_n \left(\frac{1}{N_{B,A,C}} \cdot P_B(i,j) \cdot P_C(m,n) \right) \quad (3)$$

$N_{B,A,C}$ is the number of valid locations for A satisfying the sequence (B, A, C) when B is at (i,j) and C is at (m,n); $P_B(i,j)$ is the available probability of node B's being located at (i,j); $P_C(m,n)$ is the available probability of node C's being located at (m,n). A naive algorithm to compute equation (3) has complexity $O(d \cdot N \cdot S^3)$. However, since the marginal probability indeed comes from only one dimension along the scanning direction (e.g., a line), the complexity can be reduced to $O(d \cdot N \cdot S^{1.5})$ after algorithm optimization. In addition, the final location areas for every node are much smaller than the original field S; therefore, in practice, DBE MSP can be computed much faster than $O(d \cdot N \cdot S^{1.5})$.

6 Wave Propagation Example

So far, the description of MSP has been solely in the context of straight-line scan. However, we note that MSP is conceptually independent of how the event is propagated as long as node sequences can be obtained. Clearly, we can also support wave-propagation-based events (e.g., ultrasound propagation, air blast propagation), which are polar coordinate equivalences of the line scans in the Cartesian coordinate system. This section illustrates the effects of MSP's implementation in the wave propagation-based situation. For easy modelling, we have made the following assumptions:

- The wave propagates uniformly in all directions, therefore the propagation has a circle frontier surface. Since MSP does not rely on an accurate space-time relationship, a certain distortion in wave propagation is tolerable. If any directional wave is used, the propagation frontier surface can be modified accordingly.

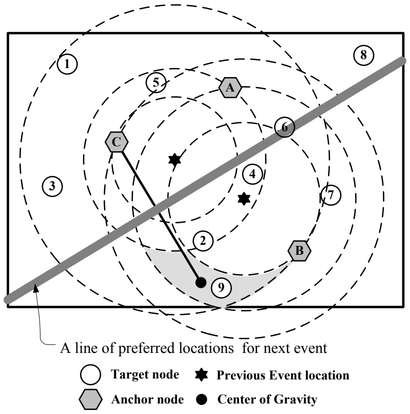


Figure 11. Example of Wave Propagation Situation

- Under the situation of line-of-sight, we allow obstacles to reflect or deflect the wave. Reflection and deflection are not problems because each node reacts only to the first detected event. Those reflected or deflected waves come later than the line-of-sight waves. The only thing the system needs to maintain is an appropriate time interval between two successive localization events.
- We assume that background noise exists, and therefore we run a band-pass filter to listen to a particular wave frequency. This reduces the chances of false detection.

The parameter that affects the localization event generation here is the *source location of the event*. The different distances between each node and the event source determine the rank of each node in the node sequence. Using the node sequences, the MSP algorithm divides the whole area into many non-rectangular areas as shown in Figure 11. In this figure, the stars represent two previous event sources. The previous two propagations split the whole map into many areas by those dashed circles that pass one of the anchors. Each node is located in one of the small areas. Since sequence-based MSP, iterative MSP and DBE MSP make no assumptions about the type of localization events and the shape of the area, all three optimization algorithms can be applied for the wave propagation scenario.

However, adaptive MSP needs more explanation. Figure 11 illustrates an example of nodes' voting for next event source locations. Unlike the straight-line scan, the critical parameter now is the location of the event source, because the distance between each node and the event source determines the rank of the node in the sequence. In Figure 11, if the next event breaks out along/near the solid thick gray line, which perpendicularly bisects the solid dark line between anchor C and the center of gravity of node 9's area (the gray area), the wave would reach anchor C and the center of gravity of node 9's area at roughly the same time, which would relatively equally divide node 9's area. Therefore, node 9 prefers to vote for the positions around the thick gray line.

7 Practical Deployment Issues

For the sake of presentation, until now we have described MSP in an ideal case where a complete node sequence can be obtained with accurate time synchronization. In this section we describe how to make MSP work well under more realistic conditions.

7.1 Incomplete Node Sequence

For diverse reasons, such as sensor malfunction or natural obstacles, the nodes in the network could fail to detect localization events. In such cases, the node sequence will not be complete. This problem has two versions:

- **Anchor nodes are missing in the node sequence**

  If some anchor nodes fail to respond to the localization events, then the system has fewer anchors. In this case, the solution is to generate more events to compensate for the loss of anchors so as to achieve the desired accuracy requirements.

- **Target nodes are missing in the node sequence**

  There are two consequences when target nodes are missing. First, if these nodes are still be useful to sensing applications, they need to use other backup localization approaches (e.g., Centroid) to localize themselves with help from their neighbors who have already learned their own locations from MSP. Secondly, since in advanced MSP each node in the sequence may contribute to the overall system accuracy, dropping of target nodes from sequences could also reduce the accuracy of the localization. Thus, proper compensation procedures such as adding more localization events need to be launched.

7.2 Localization without Time Synchronization

In a sensor network without time synchronization support, nodes cannot be ordered into a sequence using timestamps. For such cases, we propose a *listen-detect-assemble-report* protocol, which is able to function independently without time synchronization.

listen-detect-assemble-report requires that every node listens to the channel for the node sequence transmitted from its neighbors. Then, when the node detects the localization event, it assembles itself into the newest node sequence it has heard and reports the updated sequence to other nodes. Figure 12 (a) illustrates an example for the *listen-detect-assemble-report* protocol. For simplicity, in this figure we did not differentiate the target nodes from anchor nodes. A solid line between two nodes stands for a communication link. Suppose a straight line scans from left to right. Node 1 detects the event, and then it broadcasts the sequence (1) into the network. Node 2 and node 3 receive this sequence. When node 2 detects the event, node 2 adds itself into the sequence and broadcasts (1, 2). The sequence propagates in the same direction with the scan as shown in Figure 12 (a). Finally, node 6 obtains a complete sequence (1,2,3,5,7,4,6).

In the case of ultrasound propagation, because the event propagation speed is much slower than that of radio, the *listen-detect-assemble-report* protocol can work well in a situation where the node density is not very high. For instance, if the distance between two nodes along one direction is 10 meters, the 340m/s sound needs 29.4ms to propagate from one node to the other. While normally the communication data rate is 250Kbps in the WSN (e.g., CC2420 [1]), it takes only about 2 $\sim$ 3 ms to transmit an assembled packet for one hop.

One problem that may occur using the *listen-detect-assemble-report* protocol is multiple partial sequences as shown in Figure 12 (b). Two separate paths in the network may result in two sequences that could not be further combined. In this case, since the two sequences can only be processed as separate sequences, some order information is lost. Therefore the accuracy of the system would decrease.

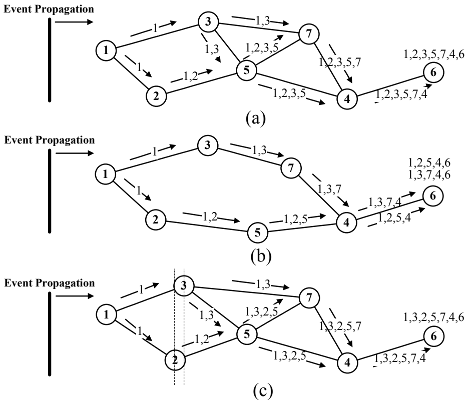


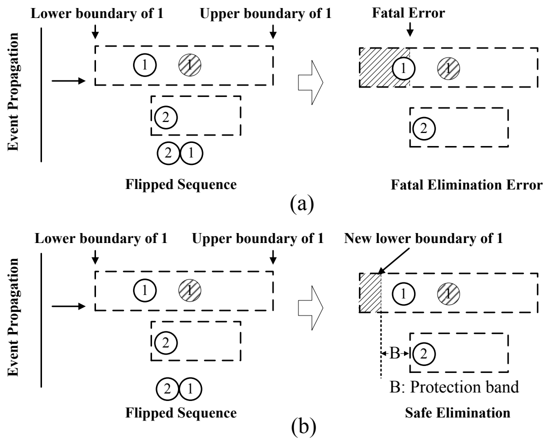
Figure 12. Node Sequence without Time Synchronization

The other problem is the sequence flip problem. As shown in Figure 12 (c), because node 2 and node 3 are too close to each other along the scan direction, they detect the scan almost simultaneously. Due to the uncertainty such as media access delay, two messages could be transmitted out of order. For example, if node 3 sends out its report first, then the order of node 2 and node 3 gets flipped in the final node sequence. The sequence flip problem would appear even in an accurately synchronized system due to random jitter in node detection if an event arrives at multiple nodes almost simultaneously. A method addressing the sequence flip is presented in the next section.

7.3 Sequence Flip and Protection Band

Sequence flip problems can be solved with and without time synchronization. We firstly start with a scenario applying time synchronization. Existing solutions for time synchronization [12, 6] can easily achieve sub-millisecond-level accuracy. For example, FTSP [12] achieves 16.9μs (microsecond) average error for a two-node single-hop case. Therefore, we can comfortably assume that the network is synchronized with maximum error of 1000μs. However, when multiple nodes are located very near to each other along the event propagation direction, even when time synchronization with less than 1ms error is achieved in the network, sequence flip may still occur. For example, in the sound wave propagation case, if two nodes are less than 0.34 meters apart, the difference between their detection timestamp would be smaller than 1 millisecond.

We find that sequence flip could not only damage system accuracy, but also might cause a fatal error in the MSP algorithm. Figure 13 illustrates both detrimental results. In the left side of Figure 13(a), suppose node 1 and node 2 are so close to each other that it takes less than 0.5ms for the localization event to propagate from node 1 to node 2. Now unfortunately, the node sequence is mistaken to be (2,1). So node 1 is expected to be located to the right of node 2, such as at the position of the dashed node 1. According to the elimination rule in sequence-based MSP, the left part of node 1's area is cut off as shown in the right part of Figure 13(a). This is a potentially fatal error, because node 1 is actually located in the dashed area which has been eliminated by mistake. During the subsequent eliminations introduced by other events, node 1's area might be cut off completely, thus node 1 could consequently be erased from the map! Even in cases where node 1 still survives, its area actually does not cover its real location.



(a)

(b)

Figure 13. Sequence Flip and Protection Band

Another problem is not fatal but lowers the localization accuracy. If we get the right node sequence (1,2), node 1 has a new upper boundary which can narrow the area of node 1 as in Figure 3. Due to the sequence flip, node 1 loses this new upper boundary.

In order to address the sequence flip problem, especially to prevent nodes from being erased from the map, we propose a *protection band* compensation approach. The basic idea of *protection band* is to extend the boundary of the location area a little bit so as to make sure that the node will never be erased from the map. This solution is based on the fact that nodes have a high probability of flipping in the sequence if they are near to each other along the event propagation direction. If two nodes are apart from each other more than some distance, say, B, they rarely flip unless the nodes are faulty. The width of a protection band B, is largely determined by the maximum error in system time synchronization and the localization event propagation speed.

Figure 13(b) presents the application of the *protection band*. Instead of eliminating the dashed part in Figure 13(a) for node 1, the new lower boundary of node 1 is set by shifting the original lower boundary of node 2 to the left by distance B. In this case, the location area still covers node 1 and protects it from being erased. In a practical implementation, supposing that the ultrasound event is used, if the maximum error of system time synchronization is 1ms, two nodes might flip with high probability if the timestamp difference between the two nodes is smaller than or equal to $1ms$. Accordingly, we set the protection band B as 0.34m (the distance sound can propagate within 1 millisecond). By adding the protection band, we reduce the chances of fatal errors, although at the cost of localization accuracy. Empirical results obtained from our physical test-bed verified this conclusion.

In the case of using the *listen-detect-assemble-report* protocol, the only change we need to make is to select the protection band according to the maximum delay uncertainty introduced by the MAC operation and the event propagation speed. To bound MAC delay at the node side, a node can drop its report message if it experiences excessive MAC delay. This converts the sequence flip problem to the incomplete sequence problem, which can be more easily addressed by the method proposed in Section 7.1.

8 Simulation Evaluation

Our evaluation of MSP was conducted on three platforms: (i) an indoor system with 46 MICAz motes using straight-line scan, (ii) an outdoor system with 20 MICAz motes using sound wave propagation, and (iii) an extensive simulation under various kinds of physical settings.

In order to understand the behavior of MSP under numerous settings, we start our evaluation with simulations. Then, we implemented basic MSP and all the advanced MSP methods for the case where time synchronization is available in the network. The simulation and implementation details are omitted in this paper due to space constraints, but related documents [25] are provided online at *http://www.cs.umn.edu/∼zhong/MSP*. Full implementation and evaluation of system without time synchronization are yet to be completed in the near future.

In simulation, we assume all the node sequences are perfect so as to reveal the performance of MSP achievable in the absence of incomplete node sequences or sequence flips. In our simulations, all the anchor nodes and target nodes are assumed to be deployed uniformly. The mean and maximum errors are averaged over 50 runs to obtain high confidence. For legibility reasons, we do not plot the confidence intervals in this paper. All the simulations are based on the straight-line scan example. We implement three scan strategies:

- Random Scan: The slope of the scan line is randomly chosen at each time.
- Regular Scan: The slope is predetermined to rotate uniformly from 0 degree to 180 degrees. For example, if the system scans 6 times, then the scan angles would be: 0, 30, 60, 90, 120, and 150.
- Adaptive Scan: The slope of each scan is determined based on the localization results from previous scans.

We start with basic MSP and then demonstrate the performance improvements one step at a time by adding (i) sequence-based MSP, (ii) iterative MSP, (iii) DBE MSP and (iv) adaptive MSP.

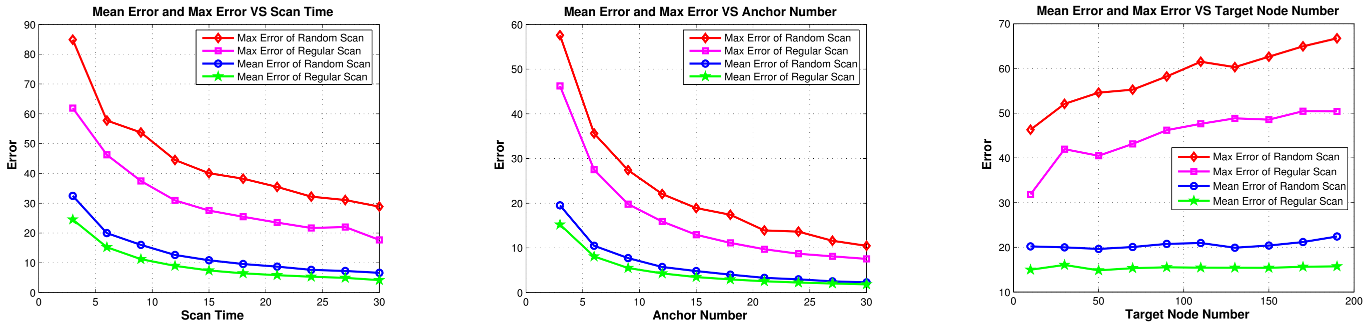
8.1 Performance of Basic MSP

The evaluation starts with basic MSP, where we compare the performance of random scan and regular scan under different configurations. We intend to illustrate the impact of the number of anchors M, the number of scans d, and target node density (number of target nodes N in a fixed-size region) on the localization error. Table 1 shows the default simulation parameters. The error of each node is defined as the distance between the estimated location and the real position. We note that by default we only use three anchors, which is considerably fewer than existing range-free solutions [8, 4].

Impact of the Number of Scans: In this experiment, we compare regular scan with random scan under a different number of scans from 3 to 30 in steps of 3. The number of anchors

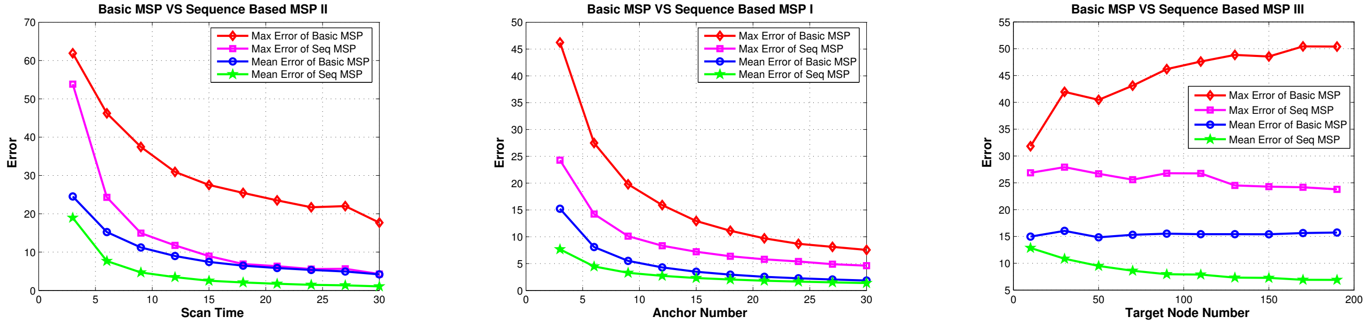
Table 1. Default Configuration Parameters

Parameter	Description
Field Area	200×200 (Grid Unit)
Scan Type	Regular (Default)/Random Scan
Anchor Number	3 (Default)
Scan Times	6 (Default)
Target Node Number	100 (Default)
Statistics	Error Mean/Max
Random Seeds	50 runs



(a) Error vs. Number of Scans (b) Error vs. Anchor Number (c) Error vs. Number of Target Nodes

Figure 14. Evaluation of Basic MSP under Random and Regular Scans



(a) Error vs. Number of Scans (b) Error vs. Anchor Number (c) Error vs. Number of Target Nodes

Figure 15. Improvements of Sequence-Based MSP over Basic MSP

is 3 by default. Figure 14(a) indicates the following: (i) as the number of scans increases, the localization error decreases significantly; for example, localization errors drop more than 60% from 3 scans to 30 scans; (ii) statistically, regular scan achieves better performance than random scan under identical number of scans. However, the performance gap reduces as the number of scans increases. This is expected since a large number of random numbers converges to a uniform distribution. Figure 14(a) also demonstrates that MSP requires only a small number of anchors to perform very well, compared to existing range-free solutions [8, 4].

Impact of the Number of Anchors: In this experiment, we compare regular scan with random scan under different number of anchors from 3 to 30 in steps of 3. The results shown in Figure 14(b) indicate that (i) as the number of anchor nodes increases, the localization error decreases, and (ii) statistically, regular scan obtains better results than random scan with identical number of anchors. By combining Figures 14(a) and 14(b), we can conclude that MSP allows a flexible tradeoff between physical cost (anchor nodes) and soft cost (localization events).

Impact of the Target Node Density: In this experiment, we confirm that the density of target nodes has no impact on the accuracy, which motivated the design of sequence-based MSP. In this experiment, we compare regular scan with random scan under different number of target nodes from 10 to 190 in steps of 20. Results in Figure 14(c) show that mean localization errors remain constant across different node densities. However, when the number of target nodes increases, the average maximum error increases.

Summary: From the above experiments, we can conclude that in basic MSP, regular scan are better than random scan under different numbers of anchors and scan events. This is because regular scans uniformly eliminate the map from different directions, while random scans would obtain sequences with redundant overlapping information, if two scans choose two similar scanning slopes.

8.2 Improvements of Sequence-Based MSP

This section evaluates the benefits of exploiting the order information among target nodes by comparing sequence-based MSP with basic MSP. In this and the following sections, regular scan is used for straight-line scan event generation. The purpose of using regular scan is to keep the scan events and the node sequences identical for both sequence-based MSP and basic MSP, so that the only difference between them is the sequence processing procedure.

Impact of the Number of Scans: In this experiment, we compare sequence-based MSP with basic MSP under different number of scans from 3 to 30 in steps of 3. Figure 15(a) indicates significant performance improvement in sequence-based MSP over basic MSP across all scan settings, especially when the number of scans is large. For example, when the number of scans is 30, errors in sequence-based MSP are about 20% of that of basic MSP. We conclude that sequence-based MSP performs extremely well when there are many scan events.

Impact of the Number of Anchors: In this experiment, we use different number of anchors from 3 to 30 in steps of 3. As seen in Figure 15(b), the mean error and maximum error of sequence-based MSP is much smaller than that of basic MSP. Especially when there is limited number of anchors in the system, e.g., 3 anchors, the error rate was almost halved by using sequence-based MSP. This phenomenon has an interesting explanation: the cutting lines created by anchor nodes are exploited by both basic MSP and sequence-based MSP, so as the

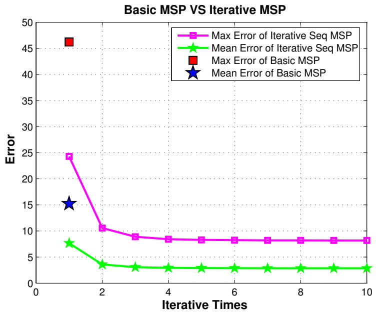


Figure 16. Improvements of Iterative MSP

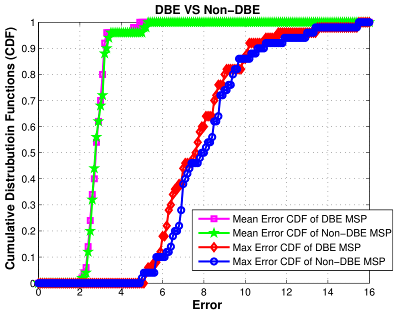


Figure 17. Improvements of DBE MSP

(a) Adaptive MSP for 500 by 80 field

(b) Impact of the Number of Candidate Events

Figure 18. The Improvements of Adaptive MSP

number of anchor nodes increases, anchors tend to dominate the contribution. Therefore the performance gaps lessens.

Impact of the Target Node Density: Figure 15(c) demonstrates the benefits of exploiting order information among target nodes. Since sequence-based MSP makes use of the information among the target nodes, having more target nodes contributes to the overall system accuracy. As the number of target nodes increases, the mean error and maximum error of sequence-based MSP decreases. Clearly the mean error in basic MSP is not affected by the number of target nodes, as shown in Figure 15(c).

Summary: From the above experiments, we can conclude that exploiting order information among target nodes can improve accuracy significantly, especially when the number of events is large but with few anchors.

8.3 Iterative MSP over Sequence-Based MSP

In this experiment, the same node sequences were processed iteratively multiple times. In Figure 16, the two single marks are results from basic MSP, since basic MSP doesn't perform iterations. The two curves present the performance of iterative MSP under different numbers of iterations c. We note that when only a single iteration is used, this method degrades to sequence-based MSP. Therefore, Figure 16 compares the three methods to one another.

Figure 16 shows that the second iteration can reduce the mean error and maximum error dramatically. After that, the performance gain gradually reduces, especially when $c > 5$. This is because the second iteration allows earlier scans to exploit the new boundaries created by later scans in the first iteration. Such exploitation decays quickly over iterations.

8.4 DBE MSP over Iterative MSP

Figure 17, in which we augment iterative MSP with distribution-based estimation (DBE MSP), shows that DBE MSP could bring about statistically better performance. Figure 17 presents cumulative distribution localization errors. In general, the two curves of the DBE MSP lay slightly to the left of that of non-DBE MSP, which indicates that DBE MSP has a smaller statistical mean error and averaged maximum error than non-DBE MSP. We note that because DBE is augmented on top of the best solution so far, the performance improvement is not significant. When we apply DBE on basic MSP methods, the improvement is much more significant. We omit these results because of space constraints.

8.5 Improvements of Adaptive MSP

This section illustrates the performance of adaptive MSP over non-adaptive MSP. We note that feedback-based adaptation can be applied to all MSP methods, since it affects only the scanning angles but not the sequence processing. In this experiment, we evaluated how adaptive MSP can improve the best solution so far. The default angle granularity (step) for adaptive searching is 5 degrees.

Impact of Area Shape: First, if system settings are regular, the adaptive method hardly contributes to the results. For a square area (regular), the performance of adaptive MSP and regular scans are very close. However, if the shape of the area is not regular, adaptive MSP helps to choose the appropriate localization events to compensate. Therefore, adaptive MSP can achieve a better mean error and maximum error as shown in Figure 18(a). For example, adaptive MSP improves localization accuracy by 30% when the number of target nodes is 10.

Impact of the Target Node Density: Figure 18(a) shows that when the node density is low, adaptive MSP brings more benefit than when node density is high. This phenomenon makes statistical sense, because *the law of large numbers* tells us that node placement approaches a truly uniform distribution when the number of nodes is increased. Adaptive MSP has an edge

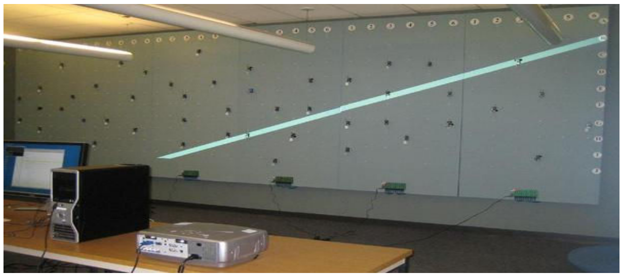


Figure 19. The Mirage Test-bed (Line Scan)

Figure 20. The 20-node Outdoor Experiments (Wave)

when layout is not uniform.

Impact of Candidate Angle Density: Figure 18(b) shows that the smaller the candidate scan angle step, the better the statistical performance in terms of mean error. The rationale is clear, as wider candidate scan angles provide adaptive MSP more opportunity to choose the one approaching the optimal angle.

8.6 Simulation Summary

Starting from basic MSP, we have demonstrated step-by-step how four optimizations can be applied on top of each other to improve localization performance. In other words, these optimizations are compatible with each other and can jointly improve the overall performance. We note that our simulations were done under assumption that the complete node sequence can be obtained without sequence flips. In the next section, we present two real-system implementations that reveal and address these practical issues.

9 System Evaluation

In this section, we present a system implementation of MSP on two physical test-beds. The first one is called Mirage, a large indoor test-bed composed of six 4-foot by 8-foot boards, illustrated in Figure 19. Each board in the system can be used as an individual sub-system, which is powered, controlled and metered separately. Three Hitachi CP-X1250 projectors, connected through a Matorx Triplehead2go graphics expansion box, are used to create an ultra-wide integrated display on six boards. Figure 19 shows that a long tilted line is generated by the projectors. We have implemented all five versions of MSP on the Mirage test-bed, running 46 MICAz motes. Unless mentioned otherwise, the default setting is 3 anchors and 6 scans at the scanning line speed of 8.6 feet/s. In all of our graphs, each data point represents the average value of 50 trials. In the outdoor system, a Dell A525 speaker is used to generate 4.7KHz sound as shown in Figure 20. We place 20 MICAz motes in the backyard of a house. Since the location is not completely open, sound waves are reflected, scattered and absorbed by various objects in the vicinity, causing a multi-path effect. In the system evaluation, simple time synchronization mechanisms are applied on each node.

9.1 Indoor System Evaluation

During indoor experiments, we encountered several real-world problems that are not revealed in the simulation. First, sequences obtained were partial due to misdetection and message losses. Second, elements in the sequences could flip due to detection delay, uncertainty in media access, or error in time synchronization. We show that these issues can be addressed by using the *protection band* method described in Section 7.3.

9.1.1 On Scanning Speed and Protection Band

In this experiment, we studied the impact of the scanning speed and the length of protection band on the performance of the system. In general, with increasing scanning speed, nodes have less time to respond to the event and the time gap between two adjacent nodes shrinks, leading to an increasing number of partial sequences and sequence flips.

Figure 21 shows the node flip situations for six scans with distinct angles under different scan speeds. The x-axis shows the distance between the flipped nodes in the correct node sequence. y-axis shows the total number of flips in the six scans. This figure tells us that faster scan brings in not only increasing number of flips, but also longer-distance flips that require wider protection band to prevent from fatal errors.

Figure 22(a) shows the effectiveness of the protection band in terms of reducing the number of unlocalized nodes. When we use a moderate scan speed (4.3feet/s), the chance of flipping is rare, therefore we can achieve 0.45 feet mean accuracy (Figure 22(b)) with 1.6 feet maximum error (Figure 22(c)). With increasing speeds, the protection band needs to be set to a larger value to deal with flipping. Interesting phenomena can be observed in Figures 22: on one hand, the protection band can sharply reduce the number of unlocalized nodes; on the other hand, protection bands enlarge the area in which a target would potentially reside, introducing more uncertainty. Thus there is a concave curve for both mean and maximum error when the scan speed is at 8.6 feet/s.

9.1.2 On MSP Methods and Protection Band

In this experiment, we show the improvements resulting from three different methods. Figure 23(a) shows that a protection band of 0.35 feet is sufficient for the scan speed of 8.57feet/s. Figures 23(b) and 23(c) show clearly that iterative MSP (with adaptation) achieves best performance. For example, Figures 23(b) shows that when we set the protection band at 0.05 feet, iterative MSP achieves 0.7 feet accuracy, which is 42% more accurate than the basic design. Similarly, Figures 23(b) and 23(c) show the double-edged effects of protection band on the localization accuracy.

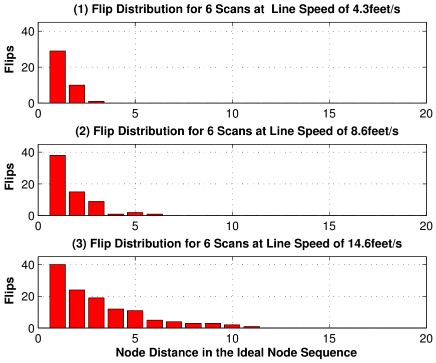
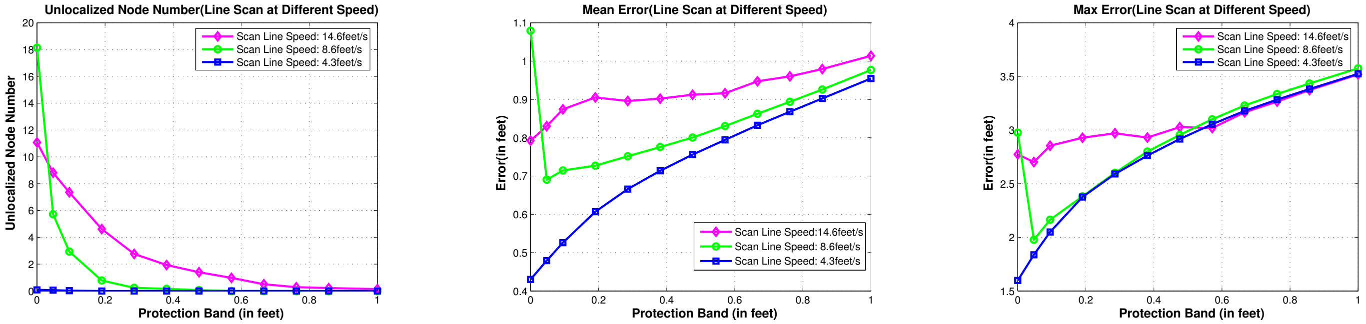


Figure 21. Number of Flips for Different Scan Speeds

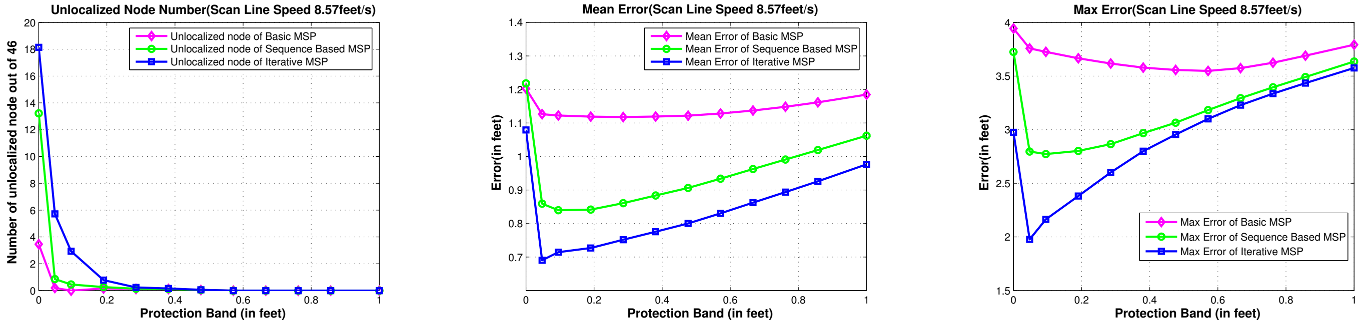


(a) Number of Unlocalized Nodes

(b) Mean Localization Error

(c) Max Localization Error

Figure 22. Impact of Protection Band and Scanning Speed

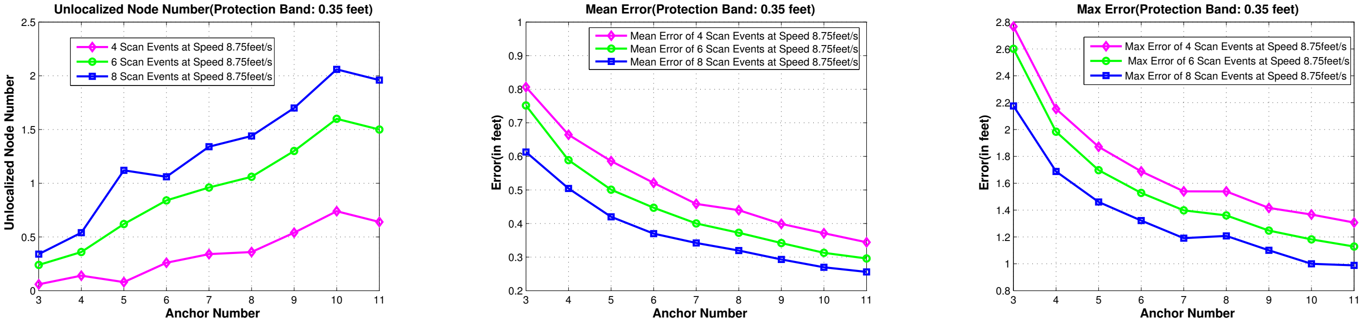


(a) Number of Unlocalized Nodes

(b) Mean Localization Error

(c) Max Localization Error

Figure 23. Impact of Protection Band under Different MSP Methods



(a) Number of Unlocalized Nodes

(b) Mean Localization Error

(c) Max Localization Error

Figure 24. Impact of the Number of Anchors and Scans

9.1.3 On Number of Anchors and Scans

In this experiment, we show a tradeoff between hardware cost (anchors) with soft cost (events). Figure 24(a) shows that with more cutting lines created by anchors, the chance of unlocalized nodes increases slightly. We note that with a 0.35 feet protection band, the percentage of unlocalized nodes is very small, e.g., in the worst-case with 11 anchors, only 2 out of 46 nodes are not localized due to flipping. Figures 24(b) and 24(c) show the tradeoff between number of anchors and the number of scans. Obviously, with the number of anchors increases, the error drops significantly. With 11 anchors we can achieve a localization accuracy as low as $0.25 \sim 0.35$ feet, which is nearly a 60% improvement. Similarly, with increasing number of scans, the accuracy drops significantly as well. We can observe about 30% across all anchor settings when we increase the number of scans from 4 to 8. For example, with only 3 anchors, we can achieve 0.6-foot accuracy with 8 scans.

9.2 Outdoor System Evaluation

The outdoor system evaluation contains two parts: (i) effective detection distance evaluation, which shows that the node sequence can be readily obtained, and (ii) sound propagation based localization, which shows the results of wave-propagation-based localization.

9.2.1 Effective Detection Distance Evaluation

We firstly evaluate the sequence flip phenomenon in wave propagation. As shown in Figure 25, 20 motes were placed as five groups in front of the speaker, four nodes in each group at roughly the same distances to the speaker. The gap between each group is set to be 2, 3, 4 and 5 feet respectively in four experiments. Figure 26 shows the results. The x-axis in each subgraph indicates the group index. There are four nodes in each group (4 bars). The y-axis shows the detection rank (order) of each node in the node sequence. As distance between each group increases, number of flips in the resulting node sequence

Figure 25. Wave Detection

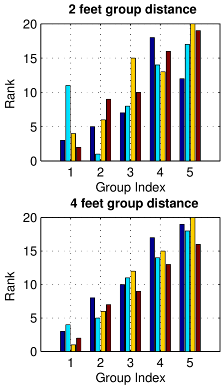


Figure 26. Ranks vs. Distances

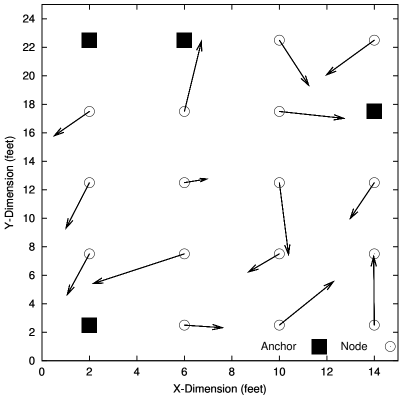


Figure 27. Localization Error (Sound)

decreases. For example, in the 2-foot distance subgraph, there are quite a few flips between nodes in adjacent and even non-adjacent groups, while in the 5-foot subgraph, flips between different groups disappeared in the test.

9.2.2 Sound Propagation Based Localization

As shown in Figure 20, 20 motes are placed as a grid including 5 rows with 5 feet between each row and 4 columns with 4 feet between each column. Six 4KHz acoustic wave propagation events are generated around the mote grid by a speaker. Figure 27 shows the localization results using iterative MSP (3 times iterative processing) with a protection band of 3 feet. The average error of the localization results is 3 feet and the maximum error is 5 feet with one un-localized node.

We found that sequence flip in wave propagation is more severe than that in the indoor, line-based test. This is expected due to the high propagation speed of sound. Currently we use MICAz mote, which is equipped with a low quality microphone. We believe that using a better speaker and more events, the system can yield better accuracy. Despite the hardware constrains, the MSP algorithm still successfully localized most of the nodes with good accuracy.

10 Conclusions

In this paper, we present the first work that exploits the concept of *node sequence* processing to localize sensor nodes. We demonstrated that we could significantly improve localization accuracy by making full use of the information embedded in multiple easy-to-get one-dimensional node sequences. We proposed four novel optimization methods, exploiting order and marginal distribution among non-anchor nodes as well as the feedback information from early localization results. Importantly, these optimization methods can be used together, and improve accuracy additively. The practical issues of partial node sequence and sequence flip were identified and addressed in two physical system test-beds. We also evaluated performance at scale through analysis as well as extensive simulations. Results demonstrate that requiring neither costly hardware on sensor nodes nor precise event distribution, MSP can achieve a sub-foot accuracy with very few anchor nodes provided sufficient events.

11 References

[1] *CC2420 Data Sheet*. Avaiable at http://www.chipcon.com/.

[2] P. Bahl and V. N. Padmanabhan. Radar: An In-Building RF-Based User Location and Tracking System. In *IEEE Infocom '00*.

[3] M. Broxton, J. Lifton, and J. Paradiso. Localizing A Sensor Network via Collaborative Processing of Global Stimuli. In *EWSN '05*.

[4] N. Bulusu, J. Heidemann, and D. Estrin. GPS-Less Low Cost Outdoor Localization for Very Small Devices. *IEEE Personal Communications Magazine*, 7(4), 2000.

[5] D. Culler, D. Estrin, and M. Srivastava. Overview of Sensor Networks. *IEEE Computer Magazine*, 2004.

[6] J. Elson, L. Girod, and D. Estrin. Fine-Grained Network Time Synchronization Using Reference Broadcasts. In *OSDI '02*.

[7] D. K. Goldenberg, P. Bihler, M. Gao, J. Fang, B. D. Anderson, A. Morse, and Y. Yang. Localization in Sparse Networks Using Sweeps. In *MobiCom '06*.

[8] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-Free Localization Schemes in Large-Scale Sensor Networks. In *MobiCom '03*.

[9] B. Kusy, P. Dutta, P. Levis, M. Mar, A. Ledeczi, and D. Culler. Elapsed Time on Arrival: A Simple and Versatile Primitive for Canonical Time Synchronization Services. *International Journal of Ad hoc and Ubiquitous Computing*, 2(1), 2006.

[10] L. Lazos and R. Poovendran. SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks. In *WiSe '04*.

[11] M. Maroti, B. Kusy, G. Balogh, P. Volgyesi, A. Nadas, K. Molnar, S. Dora, and A. Ledeczi. Radio Interferometric Geolocation. In *SenSys '05*.

[12] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi. The Flooding Time Synchronization Protocol. In *SenSys '04*.

[13] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust Distributed Network Localization with Noise Range Measurements. In *SenSys '04*.

[14] R. Nagpal and D. Coore. An Algorithm for Group Formation in an Amorphous Computer. In *PDCS '98*.

[15] D. Niculescu and B. Nath. Ad-Hoc Positioning System. In *GlobeCom '01*.

[16] D. Niculescu and B. Nath. Ad hoc Positioning System (APS) Using AOA. In *InfoCom '03*.

[17] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *MobiCom '00*.

[18] K. Römer. The Lighthouse Location System for Smart Dust. In *MobiSys '03*.

[19] A. Savvides, C. C. Han, and M. B. Srivastava. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In *MobiCom '01*.

[20] R. Stoleru, T. He, J. A. Stankovic, and D. Luebke. A High-Accuracy, Low-Cost Localization System for Wireless Sensor Networks. In *SenSys '05*.

[21] R. Stoleru, P. Vicaire, T. He, and J. A. Stankovic. StarDust: a Flexible Architecture for Passive Localization in Wireless Sensor Networks. In *SenSys '06*.

[22] E. W. Weisstein. Plane Division by Lines. *mathworld.wolfram.com*.

[23] B. H. Wellenhoff, H. Lichtenegger, and J. Collins. *Global Positions System: Theory and Practice,Fourth Edition*. Springer Verlag, 1997.

[24] K. Whitehouse. The Design of Calamari: an Ad-hoc Localization System for Sensor Networks. In *University of California at Berkeley*, 2002.

[25] Z. Zhong. *MSP Evaluation and Implementation Report*. Avaiable at http://www.cs.umn.edu/∼zhong/MSP.

[26] G. Zhou, T. He, and J. A. Stankovic. Impact of Radio Irregularity on Wireless Sensor Networks. In *MobiSys '04*.