Sensor-Aided Navigation in GPS-Denied Environments

Ka Yang, Daji Qiao, and Wensheng Zhang

Iowa State University, Ames, IA, 50010, USA
{yangka,daji,wzhang}@iastate.edu

Abstract. In this work we propose a novel approach to navigate users in GPS-denied environments with the help of sensors. The basic idea is to deploy wireless sensors over the field of interest and use the change of signal strength from multiple sensors to navigate users to the destination. To deal with RSS irregularity in a practical environment, an adaptive scheme is proposed to decide the scheme parameters online. We evaluated the proposed scheme through simulations and experiments. Results show that the proposed scheme navigates users to the destination successfully and efficiently with low movement overhead.

Key words: Navigation, GPS-denied, Sensor, Received Signal Strength (RSS)

1 Introduction

Global Positioning System (GPS) based navigation has been widely used in both civil and military applications. To use GPS, a receiver should receive radio signal from at least four satellites in the outer space [10] simultaneously. Thus in situations where the satellite signals cannot reach the receiver, such as sand storm or blizzard, GPS may not function properly. Moreover, since the anti-satellite techniques have become more mature [9], it is likely that the satellite positioning systems may be disabled by adversaries during war time. Therefore, it is highly desirable to have advanced navigation schemes that work under GPS-denied environments.

Many navigation systems and methods have been devised to conduct navigation in GPS-denied environments [11]. We roughly classify them as *location based systems* and *direction based systems*. For location based systems, beacons are deployed at known locations. A user calculates its position based on beacon locations and signal information such as TDoA (Time Difference of Arrival), AoA (Angle of Arrival), etc. Then a user can find a path based on its location and the destination location. One drawback of such systems is that they require the location information for beacons and measurement of distance. For direction based systems, the user finds direction towards landmarks by detecting the direction of the arriving signal, which typically requires special directional antennas.

In this work, we propose to navigate a user to the destination in GPS-denied environments using only the information of signal strength, with low-cost sensor devices. Here we use "sensor" as a generic term to represent wireless devices that are low-cost and suitable for large-area deployment. The measurement of signal strength, usually known as received signal strength (RSS), is readily available on commercial wireless devices. Our work is motivated by the experimental observation that RSS and distance usually has a monotone relation. Therefore, the change of RSS can be used as an indicator of the distance relation. We show in [12] that ideally, based on the change of RSS, a user can be navigated to a destination efficiently.

However, in practice, the signal strength fluctuates and cannot reliably reflect distance. Based on our in-depth experimental observations, we propose an adaptive navigation scheme based on RSS from multiple sensors, which has the following features: i) It adaptively decides the scheme parameters according to the environment to deal with RSS irregularity while maintaining a low movement overhead. ii) It navigates the user to the destination solely based on the comparison of RSS along user's movement, which incurs very low computational overhead. iii) It only requires low-cost omni-directional wireless devices.

We have implemented a test bed system using 49 commercial sensor motes to evaluate the proposed navigation scheme. Experimental results show that the scheme can navigate a user to a destination successfully and efficiently with low movement overhead.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 explains the system components and the motivations. Sections 4 describes the proposed navigation scheme in detail. Section 5 presents the implementation of our test bed system and reports the experimental results. Section 6 concludes the paper.

2 Related Work

Using wireless sensors for navigation purposes has been proposed in recent years [5, 1, 4, 6]. Previous works on navigation with sensor network focus on providing user the "safest" path to the destination. To move along the path, the user needs either location information (as in [5, 1]) or the directions of signals from sensors (as in [4]).

The problem of navigation has been investigated extensively in the area of robotics. Typical navigation methods for robots utilize location, range and/or bearing information from landmarks [2]. In [6], the authors propose a range-only SLAM using wireless sensor networks. The algorithm uses a predetermined RSSI-distance model to measure distance and uses an extended Kalman Filter to deal with measurement errors from the sensed RSSI. RSSI-distance model is typically considered to be environment dependent.

The RSS-distance monotonicity has been reported and used in [13]. Based on the RSS-distance monotonicity, the authors propose to use signature distance to improve the performance of existing connectivity based localization algorithms.

Different from our work, in [13], the RSS-distance relation is assumed to be monotonic from the perspective of the receiver. This requires the transmitters to have approximately the same configuration. In our work, the RSS-distance relation is based on each transmitter-receiver pair, which does not need specific calibration in practice. The RSS-distance monotonicity for current commercial wireless devices has also been validated in previous works [8, 7, 3].

3 Observations and Motivations

The goal of this work is to design a scheme that can navigate a user to the destination using only the RSS information, with low-cost sensor devices and low user movement overhead. In this section, we first explain the components of our proposed sensor-aided navigation system. Then, we present experimental and simulation results on RSS irregularity, a preliminary navigation scheme, and how it performs in the presence of RSS irregularity, which motivate us to design an advanced multi-sensor navigation scheme that will be discussed in Sections 4.

3.1 System components

In our proposed sensor-aided navigation system, a large number of sensors are deployed in the field of interest. The sensors have limited energy and computing capabilities. Each sensor has a wireless interface that can communicate with nearby sensors and sample the signal strength of the received packets. All sensors are equipped with a simple omni-directional antenna. Upon probed by the user, the sensors send back *navigation beacons* to guide the user's movement.

After deployment, a destination sensor is set by, e.g., detecting a certain event of interest. Then, through network-wise flooding, sensors form a *navigation tree* rooted at the destination sensor. Each sensor is assigned a *navigation hop count* towards the destination, as shown in Fig. 3(i).

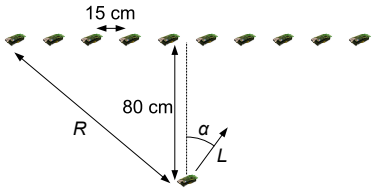
Each user, e.g., a person or a robot that needs to be navigated to the destination, carries a user sensor which is the same as the sensors deployed in the field. The user moves by steps. In each step, the user moves towards a certain direction for a certain step length. At both ends of each step, the user sensor probes nearby sensors, samples the RSS of the received navigation beacons, and makes the decision about the next step based on the collected information.

3.2 A preliminary navigation scheme

Under the ideal situation when the RSS-distance relation is strictly monotonic, we show in [12] that a user can be navigated to the destination efficiently using a naive one-sensor navigation scheme (denoted as NoS) which is based on RSS information from only one sensor. Unfortunately, RSS is notorious for its irregularity and unreliability in real-world scenarios. A strict monotonic RSS-distance relation does not always hold in practice. As a result, NoS scheme performs poorly in certain practical scenarios [12].

To deal with RSS-distance irregularity, it is naturally an attractive idea to use multiple sensors to guide user's movement. To do this, we use a metric called *increasing RSS ratio* (IRR, denoted as ρ) which is defined as the percentage of sensors whose average RSS measured by the user increases at the end of a step. We conduct experiments to study the relation between ρ and user's moving direction. Experimental setup is shown in Fig. 1, where 10 sensors (all of which transmit at -24 dBm) are placed indoor along a straight line with 15 cm apart between adjacent sensors. The user sensor is placed at 80 cm away from the center of them. In each experiment, the user moves in a different direction (α) with a different step length (L). The user collects 100 RSS readings from each sensor to determine ρ. We repeat each experiment 100 times and plot the ECDF (Empirical Cumulative Distribution Function) of ρ in Fig. 2. We have a few observations from the experiments.

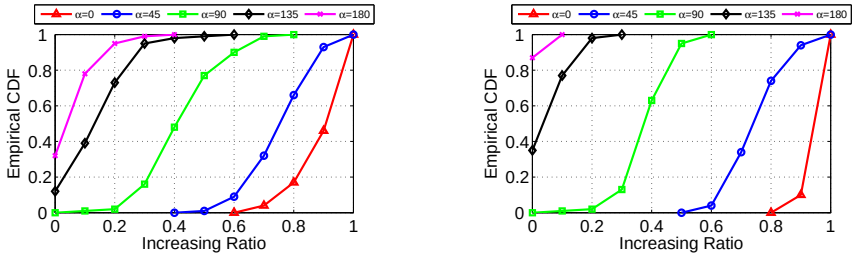
- ρ is affected by the moving direction α. As α decreases, ρ increases. For example, Fig. 2(i) reads that, if the observed ρ is higher than 60%, it is most likely that α is less than or equal to 45 degree, while there is a small possibility that α is 90 degree.
- It is interesting to see that, when α is 180 degree, we still observe non-zero ρ values in certain scenarios, meaning that even when the user moves away from all the beacon sensors, some of the beacon sensors' average RSS indeed increase. This is due mainly to the high RSS irregularity in environments with obstacles.
- ρ is affected by the step length L. As shown in Fig. 2, a larger L results in more separation between ECDF curves of ρ for different α values. As a result, the relation between ρ and α becomes less ambiguous. For example, Fig. 2(ii) reads that, if the observed ρ is higher than 60%, α is less than or equal to 45 degree with 100% probability.



(i)

(ii)

Fig. 1. Experiments for testing multiple sensors' RSS. (i) α denotes the angle between the receiver's moving direction and the perpendicular bisector of the transmitters. R is the sensor's communication range, which is about one meter. (ii) Experimental setup.



(i) $L = 10$ cm (ii) $L = 25$ cm

Fig. 2. ECDF of ρ for different α and L.

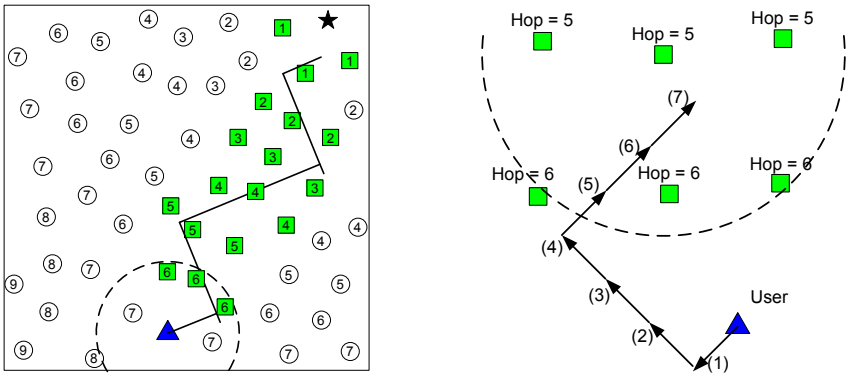
Based on the above observations, we propose a basic multi-sensor navigation scheme, denoted as BmS, which uses ρ as an indicator of moving direction. It consists of the following steps.

- *Pick beacon sensors*: The user sensor collects the hop count information of the sensors within its communication range and selects *beacon sensors* to move towards. The beacon sensors are these that have the smallest hop count in the user sensor's communication range, which are potentially closest to the destination. For example, in Fig. 3(i), at the starting point, the user sensor overhears sensors with a hop count of 6 or 7. All the sensors with a hop count of 6 are selected as beacon sensors. Then the user takes the first step by moving towards a random direction.
- *Compare RSS*: After each step, the user sensor calculates ρ for beacon sensors at both ends of the step. If ρ is greater than a predefined threshold ρ_{th}, the user will take the next step in the same direction as the previous step. Otherwise, the user will make a clockwise turn by $(-1)^i \times (i+1) \times 90^\circ$ for the next step, where i is the number of previous consecutive turns. An example is shown in Fig. 3(ii).
- *Switch beacon sensors*: After each step, if the user sensor overhears sensors with a hop count smaller than the current beacon sensors, it will select new beacon sensors as described in the first step.

3.3 Observations from simulations

We define *stretching factor* as the metric to quantitatively evaluate the performance of a navigation scheme, which is the ratio of the straight-line distance between the user and the destination, to the actual distance traversed by the user to reach the destination. Clearly, the closer the stretching factor is to one, the better the navigation scheme performs. We evaluate the performance of BmS using simulations. We use the following formula to generate the RSS samples:

$$\mathrm{RSS}_d = \mathrm{RSS}_0 - 10 \times \beta \times \log(d/d_0) + X, \tag{1}$$



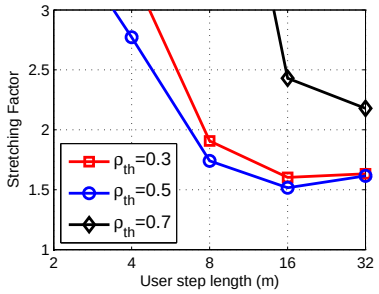
(i) (ii)

Fig. 3. The basic multi-sensor navigation scheme (BmS). In this example, ρ_{th} is set to 0.5. (i) An example network and a sample trace of a BmS user. Star: destination sensor; triangle: user sensor; squares: beacon sensors; the dashed circle represents the communication range of the user sensor. (ii) A zoom-in view of the user trace. After step 5, the user sensor switches to three new beacon sensors.

where RSS_0 is the reference RSS at a short distance of d_0. d is the distance between the transmitter and the receiver. β is the path loss factor. X is the zero-mean Gaussian noise, i.e., $X \sim N(0, \sigma^2)$. We evaluate BmS scheme under the RSS noise level of $\sigma^2 = 12$ $(\text{dBm})^2$ that approximately corresponds to the sample variance observed in our indoor experiments [12]. The simulation parameters are listed in Fig. 4(i).

parameter	value
number of sensors	500
area size	500m × 500m
sensor TX power	0 dBm
antenna sensitivity	-94 dBm
path loss factor	3
RSS sample size	100

(i)



(ii)

Fig. 4. (i) Simulation parameters. (ii) Simulation results: the stretching factor of the BmS scheme under high noise level of $\sigma^2 = 12$ $(\text{dBm})^2$. Each point is averaged over 50 runs.

From the results plotted in Fig. 4(ii), we observe interesting tradeoffs in setting proper L and ρ_{th}. On one hand, as L increases, the RSS-distance mono-

tonicity becomes more reliable, which helps to make more proper movement decisions and reduce the stretching factor. However, as L increases, if the user moves in a wrong direction, a larger movement overhead is incurred. So there is a tradeoff for choosing a proper step length. For example, when $\rho_{th} = 0.5$, the optimal step length that yields the smallest stretching factor is 16 meters.

On the other hand, for a fixed L, there is also a corresponding optimal ρ_{th}. For example, when $L = 16$ meters, $\rho_{th} = 0.5$ yields the better performance (stretching factor around 1.5), while other ρ_{th} values result in a larger stretching factor. The rationale behind this is that, if ρ_{th} is set too low, the range of α which results in $\rho \geqslant \rho_{th}$ may be too large. As a result, the user may make wrong decisions and move towards wrong directions. However, if ρ_{th} is set too high, the range of α which results in $\rho \geqslant \rho_{th}$ may be too small. Since BmS scheme makes turns in units of 90 degrees, it may not be able to find a proper direction that satisfies the ρ_{th} requirement. In fact, from the experiments we can see that user's turning angle (denote as ϕ) is another parameter that affects the performance of the navigation scheme.

Motivated by the observations on RSS irregularity and the performances of BmS schemes, we propose an advanced multi-sensor navigation scheme (denoted as AmS) and the details of AmS scheme will be presented in the next section. AmS is based on BmS but with more sophisticated control on setting the following three parameters: decision threshold for the increasing RSS ratio (ρ_{th}), the step length (L) and the turning angle (ϕ).

4 The Proposed AmS Navigation Scheme

The advanced navigation scheme proposed in this work consists of two components: *the navigation network* and *the user navigation scheme*. The navigation network serves as a base for the navigation scheme. The goal of constructing the navigation network is to let each sensor obtain its navigation hop count towards the destination, which generally reflects its distance to the destination. Due to space limitation, we do not present how the navigation network is constructed but only explain the user navigation scheme in this paper. Please refer to [12] for details about construction of the navigation network.

4.1 User step length (L)

As discussed in Section 3.2, the user step length should be long enough to deal with RSS irregularity but should be kept small to reduce the movement overhead. In this section we describe an online probing-based heuristic for the user to find a proper step length.

Searching for a proper user step length is initiated at the beginning of the navigation process, or triggered by change of the environment, which will be explained in Section 4.2. The flowchart of how the user step length is decided is shown in Fig. 5. The heuristic is based on the fact that the average IRR

decreases as the corresponding moving direction deviates from the destination direction. Therefore, if we let the user make probing steps in different directions, by checking whether the relation of the IRRs are consistence with the direction relation, we can verify whether the user step length is proper. We explain the proposed heuristic using the example shown in Fig. 6.

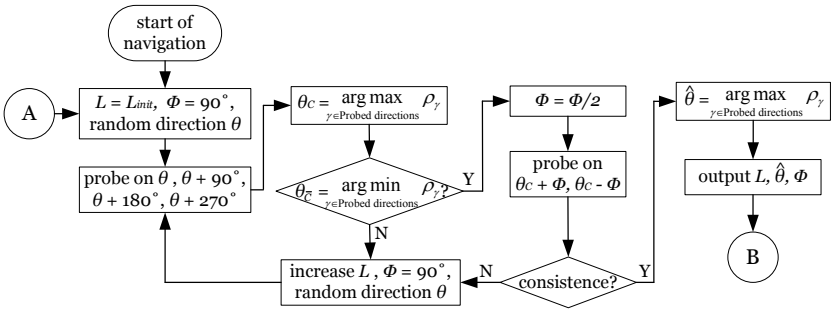


Fig. 5. The flowchart of deciding a proper L. The consistency conditions are explained in Fig. 6.

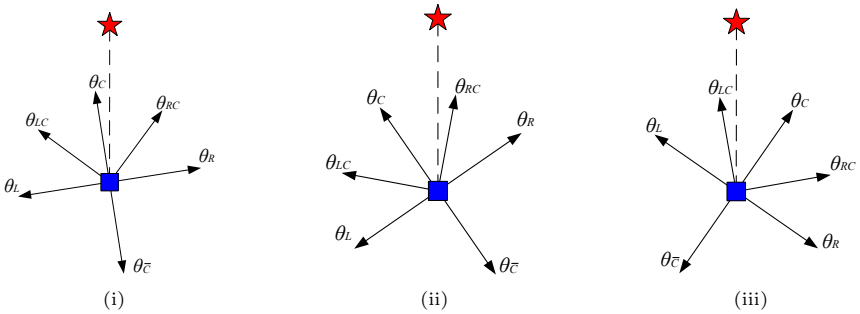


Fig. 6. Consistency conditions: (i) $\rho_{\theta_C} > \rho_{\theta_{LC}} > \rho_{\theta_L}$, and $\rho_{\theta_C} > \rho_{\theta_{RC}} > \rho_{\theta_R}$; (ii) $\rho_{\theta_{RC}} > \rho_{\theta_C} > \rho_{\theta_{LC}} > \rho_{\theta_L}$; (iii) $\rho_{\theta_{LC}} > \rho_{\theta_C} > \rho_{\theta_{RC}} > \rho_{\theta_R}$. Star indicates the location of the destination.

The searching process starts with initial step length L_{init} and initial turning angle $\phi = 90°$. L_{init} could be the minimal step length the user can take in practice. Firstly, the user selects a random direction, say, θ and takes a step towards direction θ, $\theta + 180°$, $\theta + 90°$ and $\theta + 270°$ respectively. We denote the direction that results in the highest IRR as θ_C. Denote $\theta_C + 90°$, $\theta_C + 270°$, $\theta_C + 180°$ as θ_R, θ_L, $\theta_{\overline{C}}$ respectively (see Fig. 6). Apparently, if the step length is long enough, the IRR towards the direction of $\theta_{\overline{C}}$ should have the lowest IRR. If this condition is satisfied, the user moves on to the secondary probing stage.

Otherwise, we consider the current step length to be shorter than necessary and the process restarts with a longer step length.

In the secondary probing stage, the user reduces ϕ to half of the previous ϕ (i.e., 45°) and continues to probe two directions around θ_C (i.e., $\theta_{RC} = \theta_C + \phi$ and $\theta_{LC} = \theta_C$ - ϕ), as shown in Fig. 6. Then it checks the consistency between the IRRs and the moving directions. Based on the direction of θ_C, there might be three different consistency conditions: namely, (i) $\rho_{\theta_C} > \rho_{\theta_{LC}} > \rho_{\theta_L}$ and $\rho_{\theta_C} > \rho_{\theta_{RC}} > \rho_{\theta_R}$; (ii) $\rho_{\theta_{RC}} > \rho_{\theta_C} > \rho_{\theta_{LC}} > \rho_{\theta_L}$; (iii) $\rho_{\theta_{LC}} > \rho_{\theta_C} > \rho_{\theta_{RC}} > \rho_{\theta_R}$. If any of the three conditions is satisfied, the current step length is output as the desired step length and the direction with the highest IRR is set as the moving direction for the next step, denoted as $\hat{\theta}$. For example, in Figs. 6(i)-(iii), $\hat{\theta}$ is set to be θ_C, θ_{RC} and θ_{LC} respectively.

4.2 Decision threshold for IRR (ρ_{th}) and turning angle (ϕ)

In this section, we describe how a user decides ρ_{th} and ϕ. The flowchart of this process is shown in Fig. 7.

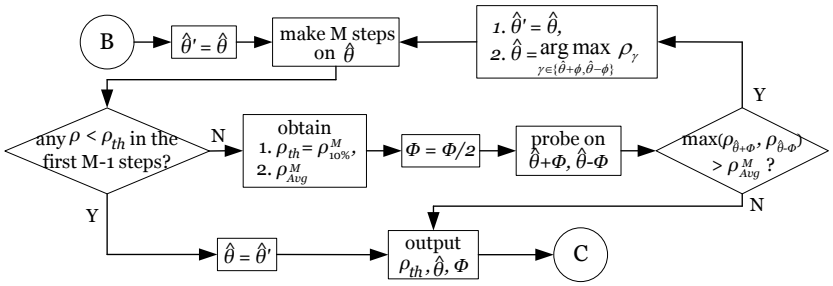


Fig. 7. The flowchart of deciding ρ_{th} and ϕ.

Recall that ρ_{th} is used to decide whether the moving direction has deviated from the desired range of moving directions. To decide ρ_{th}, we let the user take M steps towards the direction of $\hat{\theta}$. We rank the M IRRs in the increasing order and get its 10^{th} percentile (denoted as $\rho^M_{10\%}$) value. We set $\rho_{th} = \rho^M_{10\%}$. Statistically, if M is large, steps in the current moving direction has a probability of about 90% that the resulting IRR is greater than ρ_{th}. On the other hand, we want to keep M small so that if the current direction is not good, the user can react quickly to reduce the movement overhead. In our scheme, we use $M = 11$ and $\rho^M_{10\%}$ will be the second smallest IRR. Besides, the user also calculates the average of the M IRRs, denoted as ρ^M_{avg}, which will be used for the following steps.

After the step length is decided, it is possible that the output $\hat{\theta}$ is apart from the destination direction and we need to check whether there is a better moving direction. To do this, we let the user probe two directions around $\hat{\theta}$ (i.e., $\hat{\theta} + \phi$ and $\hat{\theta}$ - ϕ) with a smaller ϕ that is half of the previous ϕ. If any of the

probed directions is better than the current direction, its expected IRR should be greater than ρ_{avg}^{M}. Therefore, if the user gets $\rho > \rho_{avg}^{M}$ from any of the probed directions, it will update the moving direction to be the direction which has greater IRR. Otherwise, the user stops probing and continues along the current direction of $\hat{\theta}$.

If the user changes the direction of $\hat{\theta}$, to avoid the situation where the user mistakenly chooses a bad direction, we let the user make another M steps towards the new direction of $\hat{\theta}$. If any of the first $M-1$ steps yields an IRR smaller than ρ_{th}, the new direction is considered improper and the user will change back to the previous $\hat{\theta}$. Otherwise, the user calculates ρ_{avg}^{M} and $\rho_{10\%}^{M}$, which becomes the new ρ_{th}. The user repeats this procedure, with smaller ϕ each time, until it cannot get a better direction.

4.3 Runtime adaptation

The turning angle ϕ is used when a user encounters $\rho < \rho_{th}$. If $\hat{\theta}$ deviates from the desired range of moving direction, the user expects to find the correct direction from its current close-by directions. Thus we let the user probe two directions around θ (i.e., $\hat{\theta} + \phi$ and $\hat{\theta}$ - ϕ). To deal with outlier ρ, after the two probes, the user also makes a step towards the direction of $\hat{\theta}$. Then the user gets three IRRs, namely, $\rho_{\hat{\theta}-\phi}$, $\rho_{\hat{\theta}+\phi}$, and $\rho_{\hat{\theta}}$. Because at least one of the three probed directions is expected to be in the range of direction corresponding to ρ_{th}, without change of environmental noise, the max of $\rho_{\hat{\theta}-\phi}$, $\rho_{\hat{\theta}+\phi}$, and $\rho_{\hat{\theta}}$ should be greater than ρ_{th} with high probability. Thereby the user can continue to move towards the direction with max ρ.

In case the environment changes, e.g., the RSS noise level increases, it will render the current L too short to deal with RSS irregularity. Specifically, larger noise will increase the probability that $\rho < \rho_{th}$. Therefore, after the aforementioned probing process, the max of $\rho_{\hat{\theta}-\phi}$, $\rho_{\hat{\theta}+\phi}$, and $\rho_{\hat{\theta}}$ may still be smaller than ρ_{th}. To deal with such outlier, we let the user repeat the probing one more time. If it still cannot get $\rho > \rho_{th}$, it indicates that the environment may have changed and the user should restart from the step length searching process to find a proper L and ρ_{th}. Fig. 8 shows the flowchart of deciding whether to start a new search.

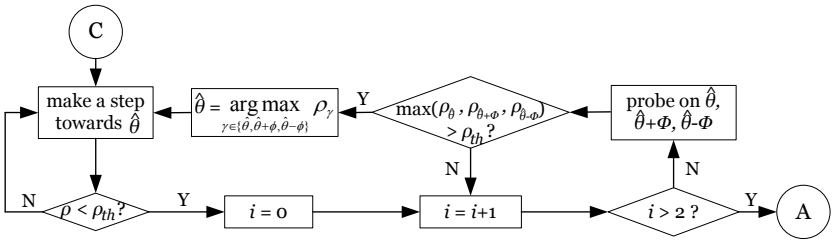


Fig. 8. The flowchart of deciding whether to start a new search.

4.4 Example trace

Fig. 9 shows an example trace of the proposed navigation scheme from simulation, whose setup is the same as described in Section 3.2. The user starts from step length of 4 meters and increases the step length by 2 meters each time. The trace shows the first 50 steps in one simulation run. For each step, we plot ρ, ρ_{th}, L and ϕ.

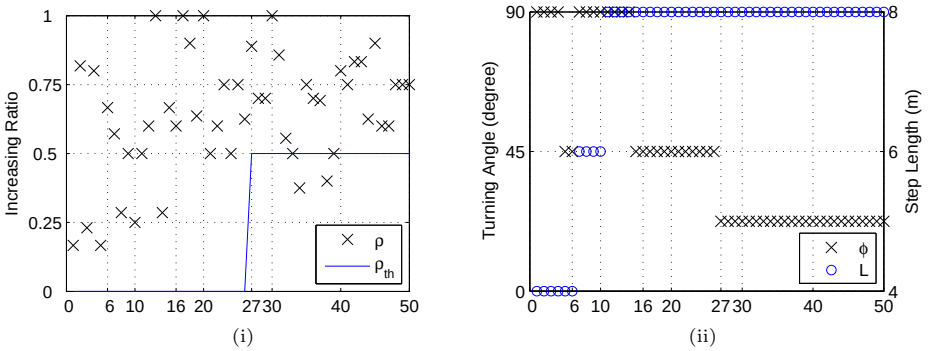


Fig. 9. Example trace of the proposed scheme from simulation. The X-axis shows the step index. (i) plots instant ρ and ρ_{th}. (ii) plots L and ϕ.

During steps 1∼6 and 7∼10, the user probes in step length of 4 meters and 6 meters. But the increasing RSS ratios do not pass the consistence check. The user then probes in step length of 8 meters and pass the consistence check after step 16. Then the user probes on the direction with maximum increasing RSS ratio until step 27, where the user get $\rho_{th} = 0.5$. Then the user reduces ϕ and probes around the current moving direction in steps 28 and 29. But none of the increasing RSS ratio is greater than the average increasing RSS ratio in the previous 11 steps. Therefore, the user stops searching for L and ρ_{th} at step 30. At step 34, the increasing RSS ratio goes under ρ_{th}, then the user probes around the current moving direction in steps 35, 36, and 37. Then it changes its direction to that of step 35 because it has the highest ρ. We can see that the user probes again in steps 39∼41 because the increasing RSS ratio drops below ρ_{th} again. Because the user stops probing for the third time, the user does not restart the searching process.

5 Performance Evaluation

We have conducted both experiments and simulations to evaluate the effectiveness of the proposed AmS scheme. In this section, we report some selected results from our experiments and simulations.

5.1 Experiments

We have implemented a small-scale test bed system consisting of 24 MicaZ motes and 25 TelosB motes. We implemented the AmS scheme on a TelosB mote. For comparison purpose, we also implemented the NoS scheme [12] and a revised NoS scheme, in which a user makes turns by 45 degree in a way same as the AmS scheme. In this section, we report some selected results of the experiments conducted using the test bed system. The implementation details of the test bed system are omitted due to space limitation. Please refer to our technical report [12] for details.

We conducted the experiments in an office of size 7 m $\times$ 4 m. The office consisted of tables, chairs and shelves which should cause strong multi-path effect. During the experiments, there were people randomly moving in the office, which caused the sensors' signal strength to be more irregular. We used this configuration to represent the worst-case scenario where there are many static and moving obstacles. The navigation network were placed on a table of size 3.2 m $\times$ 2.5 m and were deployed in two ways, namely, regular deployment and random deployment (see Fig. 10). In the regular deployment, sensors were deployed in a 7$\times$7 grid with grid size of 0.4 m $\times$ 0.4 m. In the random deployment, we also used the same 7$\times$7 grid, but each deploying location randomly uniformly lay in a disk centered in the grid intersection with a radius of 0.2 m. Each mote was mounted to a height of about 15 cm. Navigation sensors transmit at -24 dBm and -23 dBm for TelosB motes and MicaZ motes respectively, so that they may have a similar communication range.

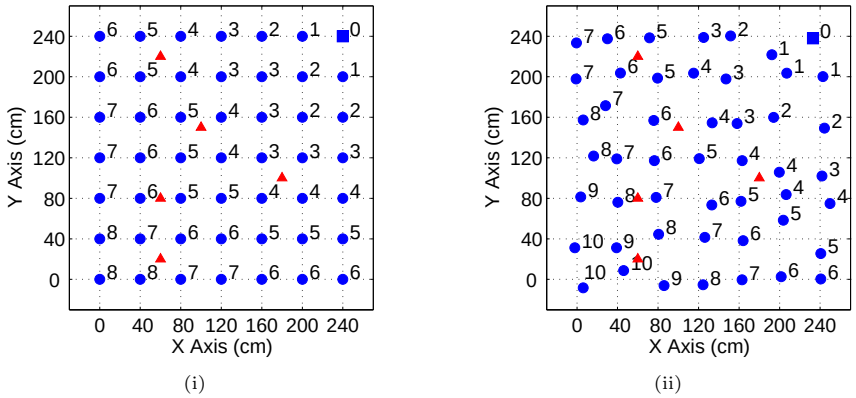


Fig. 10. The network deployment. (i) and (ii) show the network layout. Square: destination; dots: navigation sensors; triangles: the starting points for step length test in experiment I. The number to the right of each dot is the sensor's navigation hop count.

For each deployment, we ran the network formation process only once. The destination sits at the top-right corner as shown by the square marker in Fig. 10. The navigation hop count for each sensor is also shown in the figure.

Due to the limited scale of the test bed, it is infeasible to run the complete proposed scheme within the test bed. Specifically, as shown in Fig. 7, to obtain ρ_{th}, the user needs to probe M steps after deciding L, which does not always fit in the test bed. In order to evaluate the effectiveness of the proposed navigation scheme, we conducted the following two experiments.

Experiment I: user step length In the first experiment, we only ran the step length searching heuristic to obtain a proper user step length under the test environment. In each run, the user sensor started from a random location with a random initial moving direction. There were a total of 5 starting locations, as shown by the triangles in Fig. 10. For each location the user ran the heuristic for 5 times, with different random initial directions. The initial step length was set to be 10 cm and the increase unit was 5 cm. Fig. 11(i) shows the results of the 25 runs in both deployments. Under the test environment, the searching heuristic returns the step length of around 25 cm in most times. Therefore, we used step length 25 cm for the rest of the experiments.

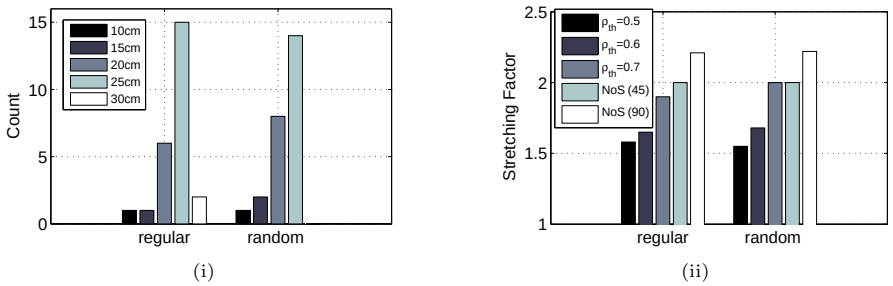


Fig. 11. (i) The results of user step length searching from 25 runs. Y axis is the count of time that the searching heuristic returns the corresponding result. (ii) Average stretching factors for different schemes in different networks.

Experiment II: stretching factor The purpose of our scheme is to navigate users to the destination successfully and efficiently with low movement overhead. The goal of this experiment is to show that our proposed scheme incurs reasonably low overhead in practice. Due to the limited scale of the test bed, we skipped the process that decides ρ_{th} online. Instead, we ran the navigation scheme with fixed L and ρ_{th}. The scheme was slightly modified accordingly: i) in the L-deciding phase as shown in Fig. 5, the consistence check is skipped (or always returns "Y") and the user direction θ is set as the direction with maximum ρ as usual; ii) in the ρ_{th}-deciding phase as shown in Fig. 7, we set M=1 because ρ_{th} is decided in advance; iii) φ was always set to be 45 degree as a result of ii).

We used L obtained from experiment I (i.e., 25 cm) and different ρ_{th} (namely, 0.5, 0.6, 0.7), and for each combination we ran the scheme for 10 times. In each run the user sensor starts from a random location near the boundaries of the

network. Each run terminates when the user is within a distance of 0.4 meter to the destination. The average stretching factors of different schemes are shown in Fig. 11(ii). The results using NoS are also presented for comparison. We can see that in the best case, i.e., when $L = 25$ cm and $\rho_{th} = 0.5$, the proposed scheme achieves an average stretching factor of about 1.6. Its overhead is about half of the overhead incurred in NoS. Moreover, in the proposed navigation scheme, the first 4 probing steps always let the user return to the starting location. In the experiments, these 4 steps typically contribute to about 0.45 in the stretching factor. We conjecture that the overhead brought by the searching process can be reduced if the scale of the network increases, which is evidenced by our simulation results in [12]. If we exclude the first 4 steps, the average stretching factors in the best case is around 1.15.

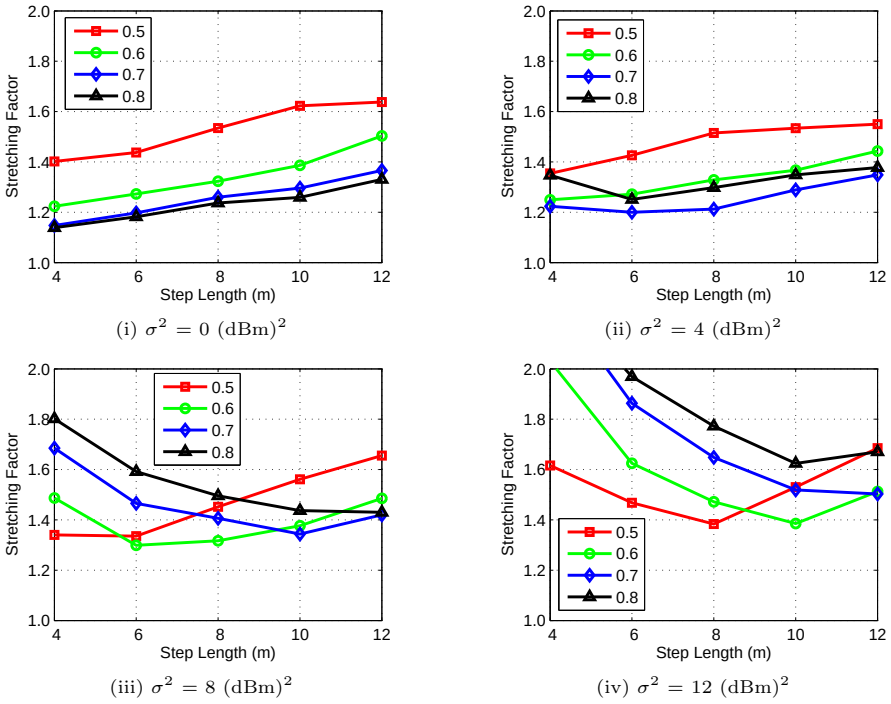
5.2 Simulations

From the experimental results we can see that our proposed scheme successfully finds a proper step length and can complete the navigation with reasonably low overhead with proper L and ρ_{th}. But due to the scale limitation, we did not evaluate our searching heuristic for deciding ρ_{th}. In this section, we evaluate the AmS scheme through simulations with large-scale networks to demonstrate the adaptiveness of the AmS scheme.

Simulation Setup We generate RSS samples in the same way as explained in Section 3.2. We use different σ^2 to represent environments with different noise levels. Note that in the simulation we did not use any of the explicit RSS-distance model information. In the simulation, the area of interest is a 2-D plane of size 1 km $\times$ 1 km. 2000 sensors are randomly uniformly deployed. The path loss factor and the antenna parameters are the same as listed in Fig. 4(i). The results are averaged over 50 simulation runs.

Adaptiveness Our proposed navigation scheme decides the user step length L and the decision threshold for increasing RSS ratio ρ_{th} online in a heuristic manner. Ideally, the scheme should eventually return L and ρ_{th} that result in optimal performance, i.e., smallest stretching factor. Because it is hard to derive theoretically optimal L and ρ_{th}, we firstly run simulations to find the optimal combination. Specifically, we simulated the scheme with different fixed L and ρ_{th}, and the combinations resulting in the smallest stretching factor are considered "optimal". To use fixed L and ρ_{th} and compare performance with the adaptive scheme, we modify the adaptive navigation scheme in the same way as in our test bed experiment II described in Section 5.1. We ran simulations under different noise levels and selected results are shown in Fig. 12.

From Fig. 12 we can see that the optimal combination of L and ρ_{th} exists and it changes as the noise level changes. For example, when $\sigma^2 = 0$ (in Fig. 12(i)), the optimal (L, ρ_{th}) = (4 m, 0.8); when $\sigma^2 = 4$ (dBm)^2 (in Fig. 12(ii)), the optimal (L, ρ_{th}) = (6 m, 0.7). Moreover, there might be more than one optimal combinations. For example, when σ^2=12 (dBm)^2 (in Fig. 12(iv)), the optimal



(i) $\sigma^2 = 0$ (dBm)^2 (ii) $\sigma^2 = 4$ (dBm)^2

(iii) $\sigma^2 = 8$ (dBm)^2 (iv) $\sigma^2 = 12$ (dBm)^2

Fig. 12. The stretching factor with fixed L and ρ_{th} under different noise level σ^2.

(L, ρ_{th}) = (8 m, 0.5) or (10 m, 0.6). We can also see that the optimal L and ρ_{th} under one noise level could perform much worse than these under a different noise level. For example, (L, ρ_{th}) = (8 m, 0.5) and (10 m, 0.6) yield the best performance when $\sigma^2 = 12$ (dBm)^2. However, when $\sigma^2 = 4$ (dBm)^2, they both incur more than twice of the overhead incurred by the optimal. This highlights the necessity of an adaptive navigation scheme in an unknown environment.

We plot the optimal stretching factors and stretching factors under different noise levels using AmS scheme in Fig. 13. We can see that our scheme performs closely to the optimal. On the other hand, fixed L and ρ_{th} may perform much worse than the optimal under certain environments. The overhead from AmS is a little higher than the optimal because: (i) the scheme has to search for proper (L, ρ_{th}) at the beginning which increases the overhead; (ii) due to the randomness of RSS, the scheme may not always be able to find the optimal (L, ρ_{th}). Fig. 14 shows an example distribution of the searching results from AmS. We can see that in most times the searching heuristic returns L and ρ_{th} close to the optimal ones. This proves that our heuristic in finding the L and ρ_{th} is effective.

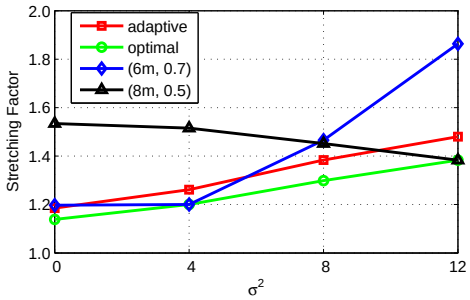


Fig. 13. Stretching factors under our adaptive navigation scheme. For comparison purpose, we also plot stretching factors under two fixed L and ρ_{th}, namely, (L, ρ_{th}) = (6 m, 0.7) and (L, ρ_{th}) = (8 m, 0.5). For simplicity, we do not plot all (L, ρ_{th}). Readers can get the performance from Fig. 12 for any (L, ρ_{th}) if interested. Clearly, the performance using fixed L and ρ_{th} varies drastically in different environments.

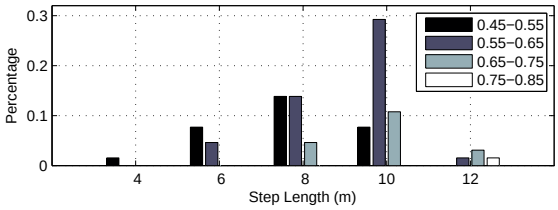


Fig. 14. We ran the L and ρ_{th} searching process for 50 times, where $\sigma^2 = 12$ (dBm)^2. This figure plots the distribution of the resulting (L, ρ_{th}). From Fig. 12(iv), we can see that the optmal (L, ρ_{th}) is (8m, 0.5) or (10m, 0.6). The distribution of (L, ρ_{th}) from our searching heuristic concentrates around these values.

6 Conclusions

In this work we investigate the problem of navigating a user to the destination using only RSS information. Our approach is based on the experimental observation that the RSS-distance relation is approximately monotonic. To deal with RSS irregularity, we propose to use increasing RSS ratio from multiple sensors as an indicator of user's moving direction. Using increasing RSS ratio, we propose an adaptive navigation scheme to decide the user step length, the decision threshold for increasing RSS ratio, and the turning angle, according to the environmental RSS noise level. We conduct both experiments and simulations to evaluate the proposed scheme. Results show that our proposed scheme can navigate users to the destination successfully and efficiently with low movement overhead under various environmental noise levels.

Acknowledgments

The research reported in this paper was supported in part by the Information Infrastructure Institute (iCube) of Iowa State University and the National Science Foundation under Grants CNS 0716744 and CNS 0831874.

References

1. C. Buragohain, D. Agrawal, and S. Suri. Distributed navigation algorithms for sensor networks. In *Proc. of INFOCOM'06*, 2006.
2. G. N. DeSouza and A. C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intellegence*, 2(24), Feb 2002.
3. M. M. Holland, R. G. Aures, and W. B. Heinzelman. Experimental investigation of radio performance in wireless sensor networks. In *Proc. of SECON'06*, 2006.
4. M. Li, Y. Liu, J. Wang, and Z. Yang. Sensor network navigation without locations. In *Proc. of INFOCOM'09*, 2009.
5. Q. Li, M. D. Rosa, and D. Rus. Distributed algorithms for guiding navigation across a sensor network. In *Proc. of MobiCom'03*, 2003.
6. E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, and E. Pagello. Range-only slam with a mobile robot and a wireless sensor networks. In *Proc. of ICRA'09*, 2009.
7. J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *Proc. of IPSN'05*, 2005.
8. W. Su and M. Alzaghal. Channel propagation characteristics of wireless micaz sensor nodes. *Ad Hoc Networks*, 7(6), Aug 2009.
9. Wikipedia. Anti-satellite weapon.
10. Wikipedia. Global positioning system.
11. Wikipedia. Navigation.
12. K. Yang, D. Qiao, and W. Zhang. Sensor aided navigation in gps-denied environment. *Technical Report*, 2010.
13. Z. Zhong and T. He. Achieving range-free localization beyond connectivity. In *Proc. of SenSys'09*, 2009.