Robust Location using System Dynamics and Motion Constraints

Camillo Gentile and Luke Klein-Berndt

National Institute of Standards and Technology

Wireless Communication Technologies Group

Email: {camillo.gentile,luke.klein-berndt}@nist.gov

Abstract— To our knowledge, the indoor location system which currently achieves the best performance using inexpensive off-the-shelf equipment locates a mobile within 1.5 meters with probability 77% in hallways. Even while maintaining this accuracy, the system often reports logical errors such as the mobile in the wrong cubicle of an office or even on the wrong side of a wall when broadening the domain of application to within rooms. We propose an extension of the work using the same Markov localization framework, however incorporating system dynamics (necessitating no post-processing of the output) and motion constraints which implicitly encode the physical properties of the survey area. Our system retains the advantages of its predecessor of low cost, wireless LAN connectivity and security, and large-scale deployment, however extending the survey area from simple hallways to the whole office environment, while maintaining the same precision without logical errors.

I. INTRODUCTION

While some inexpensive receivers using the Global Positioning System achieve location accuracy of 10 meters with probability 95% outdoors [1], systems based on this technology fail indoors due to limited reception and harsher environments created by severe multi-path fading. Indoor solutions exist which furnish submeter precision [2], [3], [4], however they require expensive hardware and/or sophisticated calibration. To our knowledge, the system described in [5] achieves the best performance available using cheap off-the-shelf equipment: this *benchmark* algorithm locates a mobile user in hallways within 1.5m with probability 77% given the proper placement of IEEE 802.11 base stations. Even while maintaining this accuracy, the system often reports logical errors such as in the wrong cubicle of an office or even on the wrong side of a wall when broadening the domain of application to within rooms. Such performance proves unacceptable for applications such as location-based recognition or emergency rescue operations.

In this paper, we describe an extension of the benchmark algorithm using the same Markov localization, however incorporating system dynamics (necessitating no post-processing of the output) and motion constraints which implicitly encode the physical properties of the survey area. Our system retains the benefits of theirs in its low cost, large-scale deployment to most indoor environments without precision calibration, along with all the features of wireless LAN including ETHERNET connection and security, however extending the survey area to the whole office environment while maintaining the same precision without logical errors. Like theirs, we describe a general Bayesian inference model which could be adapted to

a system with more sophisticated measurement instruments to achieve submeter accuracy.

The paper organization follows: Section II describes the physical system and defines the notation for the Markov localization framework. Our key contribution lies in Sections III and IV where we introduce the notions of tracks, neighbors, and allowed and restricted states. We conduct a series of experiments in Section V to quantify the performance of the proposed algorithm in comparison to the benchmark algorithm, followed by a brief conclusion in Section VI.

II. PRELIMINARIES

In our implementation, a mobile node estimates its location by measuring the signal strengths of packets received from a number N_b of fixed base stations deployed throughout a survey area. The integer readings range from -255 to 0dBm. The location system consists of two basic stages:

1) The *training stage*: Move the mobile node to a number of *anchor* locations throughout the survey area and compile signal strengths from the base stations to characterize each location.

2) The *localization stage*: Move the mobile node to a query location throughout the survey area and measure the signal strengths from the base stations. Report the location of the mobile node as the anchor location whose compiled signal strengths are closest (by some norm) to those measured at the query location.

An anchor point y_i in a set Y of N_p elements consists of a location $\mathbf{x}_i \in \mathcal{R}^d, d = \{2, 3\}$, an orientation $\theta_i = 1, -1$, and a collection of histograms for each base station, $h_{ik}, k = 0, \ldots, N_b - 1$. Since no single continuous density function for the measurement model adequately represents the signal strength distribution [6], we model the multi-path fading characteristics at each anchor point empirically through a series of measurements. An observation $\mathbf{z} = [z_0, \ldots, z_{N_b-1}]$ comprises a single signal strength reading from each of the base stations. A series of N_m observations $\mathbf{z}^l$, $l = 0, \ldots, N_m - 1$ at point y_i enables computing a histogram h_{ik} of signal strengths for each base station indexed through k:

$$h_{ik}(\zeta) = \frac{1}{N_m} \sum_{l=0}^{N_m-1} \delta(z_k^l - \zeta), \quad -255 \leq \zeta \leq 0 \quad (1)$$

where δ represents the Kronecker delta function.

III. MARKOV LOCALIZATION

A. First-Order Markov Process

Our model consists of a finite number N_s of states. State $\mathbf{s}_j = \{s_j^0, \ldots, s_j^{n-1}\}$ is a sequence of ordered points from the set Y which represents the last n points traversed by the mobile, if in that state. The benchmark algorithm uses a single point as a state of the model; incorporating more points than one captures not only the location of the mobile at a single instant in time, but also the dynamics of motion. At each time step t, the algorithm calculates the a posteriori probabilities of the model $p(\mathbf{s}_j^t | \mathbf{z}^0, \ldots, \mathbf{z}^t)$ given the observations since initialization. A first-order Markov process governs the transition of states from a time step $t-1$ to the next [7]

$$p(\mathbf{s}_j^t | \mathbf{z}^0, \ldots, \mathbf{z}^t) = \eta^t \cdot p(\mathbf{z}^t | \mathbf{s}_j^t) \sum_{\bar{j}=0}^{N_s-1} p(\mathbf{s}_j^t | \mathbf{s}_{\bar{j}}^{t-1}) \cdot p(\mathbf{s}_{\bar{j}}^{t-1} | \mathbf{z}^0, \ldots, \mathbf{z}^{t-1}),$$

$$(2)$$

where the normalizing term $\eta^t = 1 / \sum_{j=0}^{N_s-1} p(\mathbf{s}_j^t | \mathbf{z}^0, \ldots, \mathbf{z}^t)$ enforces the law of total probability. The algorithm reports the output point (location and orientation) of the system at each step t as s_*^{n-1}, $\mathbf{s}_* = \arg\max_j p(\mathbf{s}_j^t | \mathbf{z}^0, \ldots, \mathbf{z}^t)$.

Since an observation at time t affects only the component of a state at the same time (and after), the a priori probability $p(\mathbf{z}^t | \mathbf{s}_j^t) = p(\mathbf{z}^t | y_i = s_j^{n-1})$ is given by the histograms at point y_i

$$p(\mathbf{z}^t | y_i) = \prod_{k=0}^{N_b-1} h_{ik}(z_k^t).$$

$$(3)$$

To improve the reliability of the system, our implementation actually measures the signal strengths of packets both to and from the base stations as two independent readings. Each point in turn maintains two histograms per base station rather than one, doubling the elements of the product in (3).

B. System Dynamics

As described in [5] and confirmed through our testing, signal strengths fluctuate over time at the *same* point. Rather than translate these perturbations into random motion, the benchmark algorithm employs temporal averaging and outlier suppression by post-processing the output. While this improves location tracking in their experiments knowing the trajectories for simple walks up a hall and back, it fails with more complex trajectories within a room. Rather than process the output a posteriori, we encode the system dynamics in the *time-invariant* state transition probabilities a priori to promote fluent motion with constant velocity or direction, yet providing for abrupt changes.

Each state maintains a limited number of states to which it can transition at the next step in order to ensure continuity of location in time. More precisely, the state transition probability $p(\mathbf{s}_j | \mathbf{s}_{\bar{j}})$ is non-zero only for those *candidate* states $\mathbf{s}_j$ which meet the condition $\mathbf{s}_j^{l-1} = \mathbf{s}_{\bar{j}}^l, l = 1, \ldots, n-1$; in otherwords, if $\mathbf{s}_j$ is a left-shift of $\mathbf{s}_{\bar{j}}$ with replacement of s_j^{n-1} with another element in Y. The system employs the first $n-1$ points of

state $\mathbf{s}_{\bar{j}}$ to predict the next location $\hat{\mathbf{x}}$ in the sequence through the $(n-1)$-tap FIR filter

$$\hat{\mathbf{x}} = \sum_{l=1}^{n-1} \alpha^l \cdot \mathbf{x}_{\bar{j}}^l = \sum_{l=0}^{n-2} \alpha^l \cdot \mathbf{x}_j^l,$$

$$(4)$$

where the coefficients α^l denote the parameters of the optimum prediction filter [8] used in our implementation, but others filters may be applied as well. The state transition probabilities below encode the system dynamics by favoring points s_j^{n-1} closest to the predicated location $\hat{\mathbf{x}}$ assuming Gaussian noise for the location model,

$$p(\mathbf{s}_j | \mathbf{s}_{\bar{j}}) = \frac{1}{\sigma\sqrt{2\pi}^d} e^{-\frac{1}{2\sigma^2}||\mathbf{x}_j^{n-1} - \hat{\mathbf{x}}||^2},$$

$$(5)$$

where σ controls the degree of falloff.

C. The CONDENSATION Algorithm

Grid-based Markov localization handles multi-modal and non-Gaussian densities and can solve the *wake-up robot* and *kidnapped robot* problems [9] by discretizing the the state space at some resolution; however it suffers from the disadvantages of computational overhead and a priori commitment to size of the state space. Computational overhead is indeed an issue for location instruments which are often very small in size [4], [11]. In our implementation, the number of potential states increases exponentially with the order of the system n, potentially burdening our COMPAQ IPAQs running only at 200MHz, especially with most of the processing power dedicated to the communication services.

The CONDENSATION algorithm, which falls into into the general class of *particle filters*, offers an alternative approach to grid-based Markov localization. Rather than represent a posteriori density for each discretized state in the model, it maintains a set of $N_c \ll N_s$ samples drawn from $p(\mathbf{s}_j^{t-1} | \mathbf{z}^0, \ldots, \mathbf{z}^{t-1})$ and propagates the density in an iterative fashion to the next time step t. It has proven to be a powerful tool in recent years in the context of Bayesian estimation and computer vision. The details of the algorithm can be found in [10].

IV. MOTION CONSTRAINTS

Radio location using signal strengths alone proves a difficult task due to multipath fading and its intrinsic non-linearity: points close to each other may bear very different signal strengths while points farther apart very similar ones. This makes raw mapping a query point to the closest anchor point unreliable. The office environment typically offers between 25% and 40% walking space between desks, bookshelves, cubicles, and other furniture and equipment. While such obstacles severely constrain the paths of a human, they cannot reliable differentiate two points a meter apart on opposite sides of a desk given the free space between them, or even two points a meter apart on opposite sides of a plaster or drywall; yet applications such as location-based recognition or fire-rescue operations require robust discrimination in these examples respectively. While the signal strengths alone from

multiple base stations may not suffice to isolate a signal point in the training space, the sequence of points from the past observations may be able to do so.

A. Tracks and Neighbors

Fig. 1 illustrates a section of the fourth floor of the NIST North building in Gaithersburg, Maryland (not to scale to avoid clutter). Six paths displayed in different shadings connect any two locations in the office environment. The two arrows at each location represent the two anchor points trained with opposite antenna orientations, splitting a path into two *tracks*. By training each point with the antenna orientation aligned with the heading of the person, a human walking forward on a path follows the points on either one track or its complement track. Under the assumption that a human walks only forward and that the antenna orientation remains constant with respect to a person's heading, we can impose motion constraints to the system to promote motion along the tracks. This assumption holds in many location systems today such as popular *ActiveBat*, *Cricket*, or *Radar*, each requiring a beacon device which could be fastened to a human's belt or security badge, or in other applications to a firefighter's helmet.

Let each point in Y have a limited set of points, called *neighbors*, that can immediately succeed it in a state seqeunce. We impose motion constraints such that a mobile must traverse a sequence of neighbors in order to reach one point from another. Using a sequence of neighbors as an *allowed* state of the model (and restricting states which do not so conform) proves a highly effective manner to reconstruct a path from a series of observations. While classical Kalman filtering may predict the trajectory of a human through a wall by considering only trajectory locations, motion constraints provide a blueprint of the area encoded through the allowed states, so the system knows that humans must go through doors to reach locations on opposite sides of a wall. Both the notion of neighbors and tracks can be encoded in the state model of the system.

Most points have three neighbors: 1. *itself* to allow static motion in time; 2. *next point on same track* to allow motion in the same direction; 3. *point at same location on the complement track* to allow a change in direction. Exceptions occur for points at the end of a track with no next point and as a result only two neighbors, i.e. point 36; and for points falling at the T-junction of paths with additional neighbors enabling the mobile to switch paths, i.e. points 19 and 54 have additional neighbor 105 to enter *Room 444* from either direction in *Hallway*, and point 94 has neighbors 19 and 54 to exit the room walking into either direction. Note that point 105 does not have neighbors 19 and 54 and points 19 and 54 do not have neighbor 94, prohibiting walking backwards. In order to promote motion along tracks, we restrict states with more than one track transition between any two consecutive points in its sequence and penalize state transition probabilities in (5) between $\mathbf{s}_{\bar{j}}$ with no track transitions and $\mathbf{s}_j$ with one track transition by multiplying them by $0 < p_{trans} < 1$. Table I lists examples of allowed and restricted states in reference to Fig. 1.

Clearly the system can also be trained for humans walking backwards, penalizing the corresponding states with a smaller

TABLE I
ALLOWED AND RESTRICTED STATES FOR $n = 4$

Allowed	Restricted	Reason
0-0-1-2	0-2-2-3	2 not neighbor of 0
0-1-1-1	0-1-2-1	1 not neighbor of 2
0-1-72-73	0-1-71-72	71 not neighbor of 1
72-73-0-1	72-73-0-73	Changes tracks twice in sequence
71-71-72-73	2-71-2-71	Changes tracks twice in sequence

probability of occurrence. The system can also localize in large, open spaces by creating a grid of points rather than tracks and applying appropriate motion constraints.

V. EXPERIMENTAL SETUP AND RESULTS

A. Experimental Setup

In order to assess the performance of the proposed algorithm in comparison with the benchmark algorithm, we conduct four separate experiments in the confined areas labeled *Room 441*, *Room 444*, *Conference Room*, and *Hallway* in Fig. 1. The circles indicate the $N_b = 5$ base stations placed strategically throughout the floor, from which we gather $N_m = 100$ measurements at each of the $N_p = 124$ points at the rate of 10Hz to compile the histograms; so it took roughly a half hour to train the system plus the assignment of neighbors. The localization stage was run at 2Hz: high enough to allow tracking from normal to fast walking speed, yet low enough to recognize state sequences with points at multiple locations.

To replicate the benchmark algorithm we took our algorithm and set $n = 1$, removing our system dynamics and motion constraints and replacing them with more relaxed ones such "that people don't travel too fast or change directions too frequently," as described in [5]. Our algorithm was run with $n = 5$ resulting in $N_s = 3334$ allowed states. The CONDENSATION algorithm was used for both the proposed and benchmark algorithms, using $N_c = 200$ samples for each. The parameters for both algorithms were tuned to obtain the best results for each independently, with $P_{trans} = 0.5$ for the proposed algorithm.

Our experiments not only quantify the location accuracy of the two algorithms, but also classify logical errors reported according to the individual area. The system reports a location correctly if both the query and output points are:

- *Rooms 441*: on the same side of the partition (shaded in black).
- *Room 444*: on the same side of the bookcase (shaded in black).
- *Conference Room*: on the same half of the table (shaded in black).
- *Hallway*: in the same of the three segments.

We carried out 100 trials for each experiment and compiled statistics for both algorithms. Each trial consisted of the following steps:

1) Move to an anchor point in the confined area and initialize both algorithms with probability 1.0 at that point.
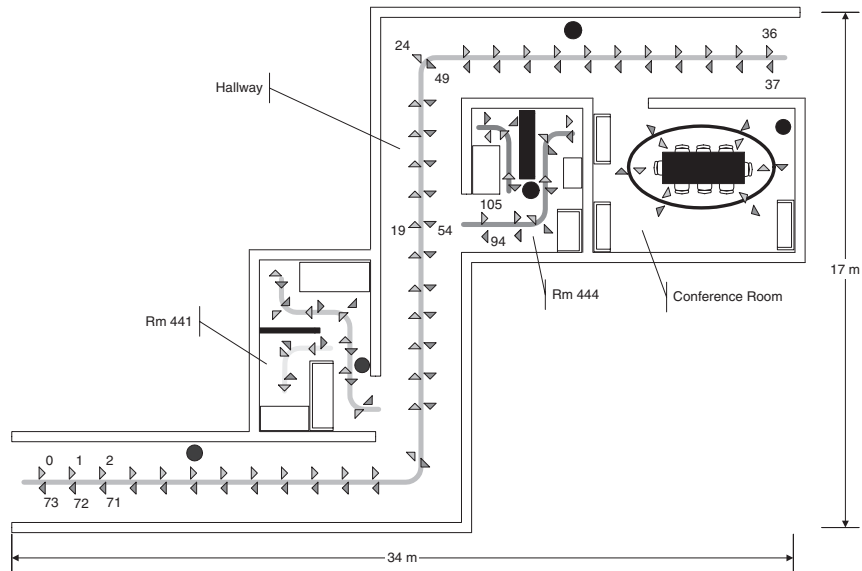2) Start both algorithms.

Fig. 1. Section of the fourth floor of the NIST North building in Gaithersburg, Maryland.

3) Move to a query location[1] within 1.5m of any anchor point in the same confined area. While covering most of the confined area, this also ensures that perfect localization guarantees accuracy within 1.5m for all trials.
4) Let the algorithms stabilize and record for each:
   - IF the algorithm commits a logical error, record X.
   - ELSE, record the error distance between the query location and the output point rounded to 0.25m.
5) Goto step 1.

B. Results

The results in [5] are also compiled from four experiments, however just in separate hallways; the best of the four[2] locates 77% of the trials within 1.5m before post-processing. In our experiments with the benchmark algorithm, a significant fraction of the trials which indeed localize within 1.5m are now categorized as logical errors, causing this value to drop to 46% (38% X) for *Room 441*, 60% (27% X) for *Room 444*, 37% (48% X) for *Conference Room*, and 51% (42% X) for *Hallway*. The respective values for the proposed algorithm follow: 74% (20% X), 76% (15% X), 75% (10% X), and 77% (9% X). The cumulative distribution function for the four experiments we conducted appear in Fig. 2 for the proposed algorithm (solid line) and the benchmark algorithm (dashed line).

The experiments conducted in *Conference Room* reveal the greatest disparity in performance between the benchmark and proposed algorithms. Here only free space lies between points on opposite sides of the table, and as a result they bear histograms too similar for robust discrimination between them. This experiment underscores the strength of the proposed

algorithm, where information about the final point does not suffice to isolate it; rather information about the trajectory from the initial point must also be taken into consideration. The proposed algorithm converged quickly and stably to a point, while the benchmark algorithm often fluctuated between two points (in which the average error was reported in compiling the statistics), necessitating post-processing.

Despite the temporal fluctuations in the point histograms, the system performs reliably after one month elapsed upon training, even after returning the base stations to their cabinets numerous times for recharging and again redeploying them each time to their marked positions. While the actual motion of the mobile may not conform to any state of the model, the algorithm estimates the best one according to the observations, and proves robust to deviations. While augmenting the number of states in the model by training the system with locations less than a meter apart (or more orientations than two) may enhance the recognition capabilities of the system, the corresponding histograms may not differ sufficiently between them, or even may be more similar to a point at a different location. This makes the system more susceptible to to noise and false alarm, fluctuating between a number of points.

VI. CONCLUSIONS AND FURTHER WORK

The application of our system easily extends to multiple floors. Indeed the proposed algorithm proves more robust for example by forcing the mobile user to descend a flight of stairs to reach a lower floor through the allowed states, rather than oscillate between two points a few meters apart with similar histograms on different floors.

Research on our location system falls in the context of a *First-Responders Testbed*, stressing the need for the ad-hoc network to organize independently of any human interaction. We are currently investigating approaches to merge the training and localization stages, hence have the system learn the point histograms as the mobile moves about with no a priori

---

[1]The trajectory between the initial anchor point and the query location was often complicated, i.e. moving up to two times from one side of the bookcase to another in *Room 444* or around the table, switch direction, and back again in *Conference Room*.

[2]Attributed to a relatively favorable placement of the base station in the middle of this hallway.

(a) *Room 441*

(b) *Room 444*

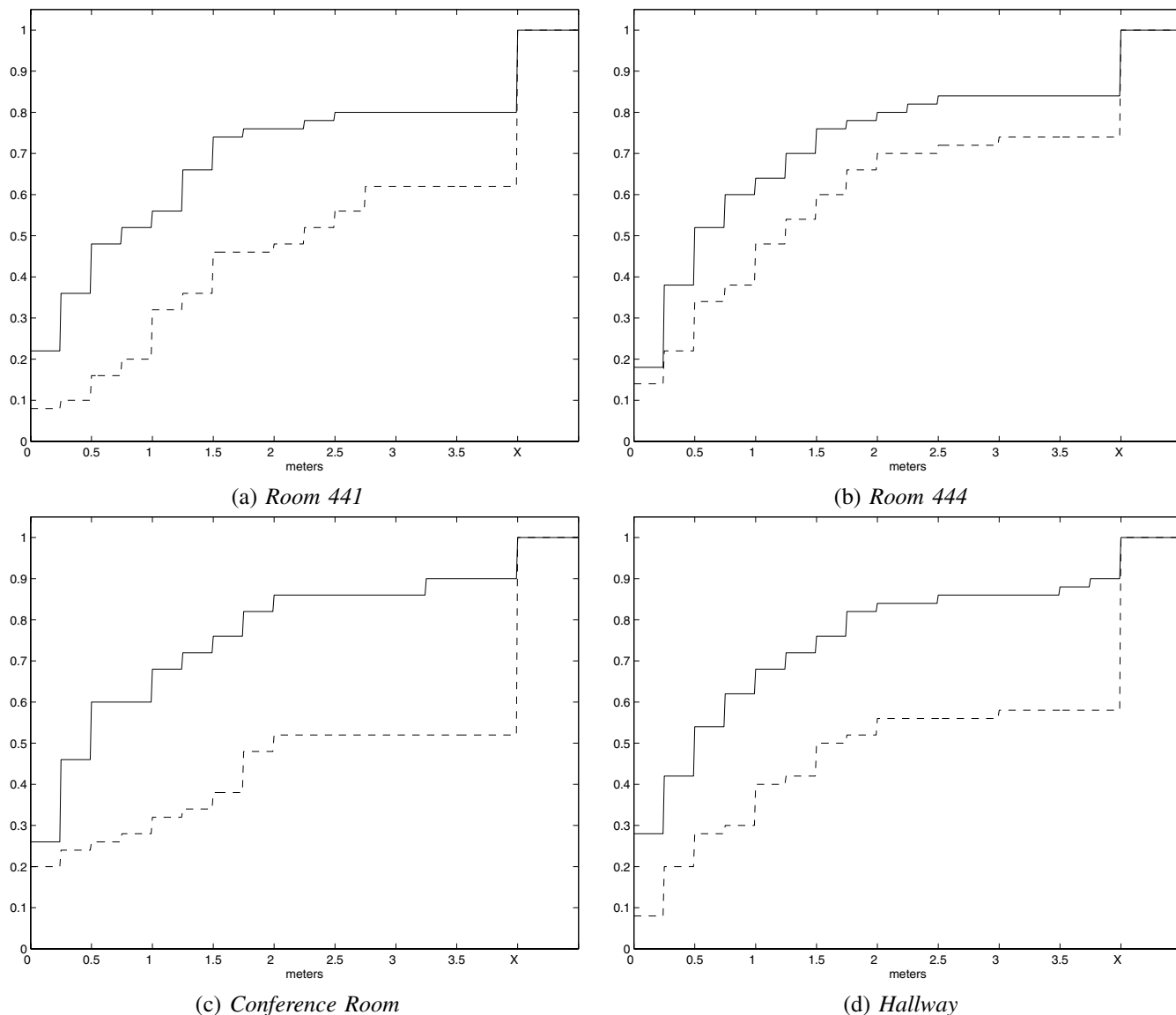(c) *Conference Room*

(d) *Hallway*

Fig. 2. Comparison of the proposed algorithm (solid line) to the benchmark algorithm (dashed line).

knowledge besides a blueprint of the survey area downloaded from a public server and the positions of the base stations deployed on site.

REFERENCES

[1] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, Vol. 34, No. 8, pp. 57-66, Aug. 2001.

[2] R.J. Fontana and S.J. Gunderson, "Ultra-wideband precision asset location system," *IEEE Conf. on Ultra Wideband Systems and Technologies*, pp. 147-150, 2002.

[3] F. Raab, E. Blood, T. Steiner, and H. Jones, "Magnetic position and orientation tracking system," *IEEE Trans. on Aerospace and Electronic Systems*, Vol 15:5, pp. 709-717, Sept. 1979.

[4] A. Harter, A. Hopper, P. Stuggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," *ACM Conf. on Mobile Computing and Networking*, pp. 59-68, Aug. 1999.

[5] A.M. Ladd, K.E. Bekris, A. Rudys, G. Marceau, L.E. Kavraki, and D.S. Wallach, "Robotics-based location sensing using wireless ethernet," *ACM Conf. on Mobile Computing and Networking*, Sept. 2002.

[6] H. Hashemi, "The indoor radio propagation channel," *Proc. of the IEEE*, Vol. 81:7, pp. 943-968, July 1993.

[7] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, Vol. 11, pp. 391-427, Nov. 1999.

[8] J.G. Proakis and D.G. Manolakis, *Digital Signal Processing*, Second Edition, MacMillan, New York, NY, pp. 828-834, 1992.

[9] W. Burgard, D. Fox, D. Hennig, and T. Schmidt, "Estimating the absolute position of a mobile robot using position probability grids," *AAAI Conf. on Artificial Intelligence*, pp. 896-901, 1996.

[10] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal on Computer Vision*, Vol 29, No. 1, pp. 5-28, 1998.

[11] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," *ACM Conf. on Mobile Computing and Networking*, Aug. 2000.