Mobile Device Positioning using Learning and Cooperation

Aleksandar Jovičić†, Ivan Klimek†, Cyril Méasson†, Tom Richardson†, Lei Zhang‡
†Qualcomm NJRC, Bridgewater, NJ, USA,
‡Northwestern University, Evanston, IL, USA

***Abstract*—We present a probabilistic graphical framework for mobile device positioning. We study the performance of a positioning algorithm, which implements the message-passing paradigm, in an indoor environment where a mobile device measures fingerprints of received signals. The key innovation in our approach is a stochastic parametric model for the fingerprint map that is adaptively tuned using on-line position estimates. The framework naturally extends to enable cooperative positioning in a network of mobile devices and we study the case of vehicle positioning as an illustration.**

I. Introduction

This paper presents some of the early investigations on mobile positioning (a.k.a localization) which is a subject of on-going research at NJRC. Our motivation for studying mobile positioning has been the development of FlashLinQ which is a novel device-to-device communication technology specifically designed to enable an efficient platform for the *proximate Internet* [1]. In a world where devices autonomously discover each other's presence, there is great commercial and social value in augmenting the basic RF proximity information with accurate position information about the discovered wireless devices. On the other hand, accurate self-localization is in itself an important capability as it enables a host of applications, the most prominent one being user navigation.

While GNSS receivers may give satisfactory accuracy in outdoor settings, it is not possible to rely on them in indoor and dense urban environments due to poor signal penetration. The main thrust of our paper will be the application of a general positioning framework, that we derive, to the problem of indoor positioning. We will present our approach as a unified framework, based on graphical models, that could potentially include all available sensor and radio inputs provided that a reasonable probabilistic model exists for them.

Our general assumption will be that external signals are used, at least in part, for localization: the mobile device measures the characteristics of signals received from transmitters (a.k.a. *anchors*) whose locations are *a priori* known and compares them with a set of stored prediction values for those characteristics. This approach is known as signal "fingerprinting" in the positioning literature (see Chapter 12 [4] for an excellent survey of the field). In the prototype indoor positioning system that we built, the external signals were WiFi beacons transmitted from access points inside a building.

The accuracy of position estimation is highly sensitive to the accuracy of the underlying fingerprint map and obtaining/creating this map is one of the key challenges in indoor positioning systems that rely on external signals. Existing methods use either exhaustive measurement campaigns (see Chapter 15 of [4]) or off-line calibrations of parametric models using partial surveys [5]. Our approach is to use a coarse ray-tracing model and a learning algorithm which adaptively tunes the parameters of the model based *on-line* position estimates. This *unsupervised* method of computing the fingerprint map is one of the main contributions of our work and it is conceptually similar to joint iterative channel estimation and decoding in communication receivers [3].

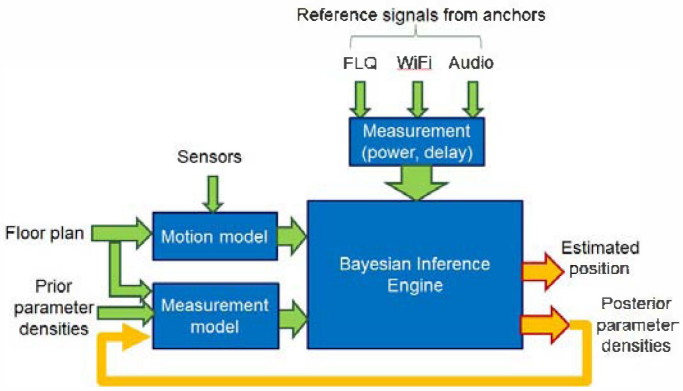

Fig. 1. A high-level block diagram of the joint positioning and learning algorithm. The Measurement Model is the fingerprint map which by ray-tracing in the case of indoor positioning.

On-line learning of propagation parameters using the EM algorithm has been studied in the context of positioning for some simple path loss models [6]. The crucial novelty of our approach is in the treatment of model parameters as *random variables* whose distributions are learned and updated using the soft outputs of the positioning algorithm. As we will see, the stochastic parameters are highly useful in overcoming unavoidable inaccuracies in the structure of the ray-tracing model that would normally cause the positioning algorithm to get stuck in only locally optimal solutions. In essence, stochastic parametrization allows us to use a very coarse, low-dimensional ray-tracing model and still get high positioning accuracy. The overall framework of joint positioning and learning given the various possible input signals and motion constraints are depicted in Fig.1.

The probabilistic graphical model framework enables a natural mechanism for performing cooperative positioning

978-1-4673-3140-1/12/$31.00 ©2012 IEEE

amongst mobile devices. If multiple devices are performing positioning and are able to communicate their position estimates to each other, the positioning accuracy of the entire system can be enhanced. This fact has been recognized in the literature, and studied in the context of UWB radio positioning [2], [12] and distributed vehicle localization [10]. In the final section, we briefly discuss the application of these ideas to the problem of improving on the GPS position accuracy in a network of vehicles.

II. Mobile positioning and learning

A. A probabilistic framework

Consider a *system* formed by one or more nodes that we want to geographically localize. For simplicity, one may first want to think of the system as a single mobile device although, in full generality, it will represent a collection of mobile devices. Let S denote the *position* of the system and let Y denote the measurement, or observation, of a characteristic of an input signal. The input signal may be an external signal transmitted by an anchor node such a WiFi AP or any other type of radio/sensor reading. The signal characteristic that we will eventually focus on in this paper is the received signal power.

Let the discrete time indices be $t \in \{1, \cdots, T\}$, $y^T = (y_1, y_2, \cdots, y_T)$ be the time sequence of measurements and $s^T = (s_1, s_2, \cdots, s_T)$ be the time sequence of positions to be estimated. Suppose first that a genie gives the likelihood function $p(y|s)$ which describes the probability density of the measurement y a mobile device would collect at location s. The positioning task is to compute the Maximum *A posteriori* Probability (MAP) position estimate at time T, i.e., $s_T^* = \text{argmax}_{s_T} p(s_T|y^T)$.

The likelihood function is the signal "fingerprint" map discussed in Section I. In practice, the likelihood function is not *a priori* known with high accuracy and one of the key tasks that we address in this paper is how to efficiently estimate it. To model the uncertainty in our knowledge of the likelihood function, we introduce a parameter (or a vector of parameters) θ and use the conditional likelihood function $p(y|s,\theta)$ in the positioning algorithm instead. One of our main contributions is the characterization of a low-dimensional parameter space, described in Section III-B, that is rich enough to be able to span a realistic set of realizations of the conditional likelihood function. The low-dimensionality of the space is important in that only few parameters must be estimated and the conditional likelihood function can be efficiently computed and stored. In the case where y corresponds to a received RF signal, the observation model depends on various electromagnetic propagation quantities such as path loss, wall loss and diffraction loss, which can be represented by the parameter θ.

As mentioned in the introduction, we treat θ as a random variable whose distribution, initially given by the prior $p(\theta)$, is estimated given the collected measurements. Then the Bayesian formulation

$$p(s^T, y^T, \theta) = \prod_{t=1}^{T} \underbrace{p(s_t|s_{t-1})}_{\text{motion model}} \cdot \underbrace{p(y_t|s_t,\theta)}_{\text{propagation model}} \cdot \underbrace{p(\theta) \cdot p(s_0)}_{\text{priors}} \tag{1}$$

gives a framework not only to estimate the positions but also to estimate the posterior distribution of the parameters. The later turns out to be very important in practice where model mismatch appears to be the first cause of positioning errors: we will turn to the question of modeling and parameter estimation in Section III-B. The graphical representation of the factorization of all relevant variables as shown in Eq.1 is depicted in Fig.2 using the standard formalism [8].

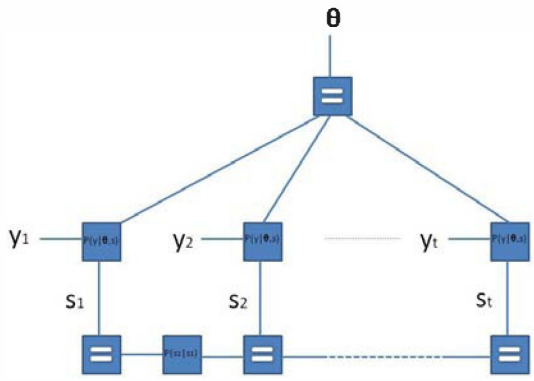

Fig. 2. Forney-Style Factor Graph for Joint Positioning and Parameter Estimation.

The transition probabilities $p(s_{t+1}|s_t)$ represent the *motion model* between positions at two consecutive time instants. Note that we have implicitly assumed that this is a Markov chain which may seem restrictive at first but which we have found to be a good enough approximation of human mobility in indoor spaces. Instead of just the basic random walk, the motion model $p(s_{t+1}|s_t)$ may further incorporate a velocity component $p(v_t)$ such that $p(s_{t+1}|s_t) = \int p(s_{t+1}|s_t, v_t)\text{d}p(v_t)$; more generally, it will make use of inertial sensors (speed and/or direction indicator, accelerometer, gyroscope, compass, etc).

B. The joint positioning and learning algorithm

We are now ready to describe our algorithm which uses the sum-product algebra and standard message-passing rules on the graph of Fig. 2. Given the density $p(y|s,\theta)$ and a prior distribution on θ, $p_o(\theta)$, a single iteration of both the positioning and the learning algorithm can indeed be stated as follows:

1) Compute the likelihood function

$$p(y|s) := \mathbb{E}_\theta \left[p(y|s,\theta) \right], \tag{2}$$

where the expectation is taken over $p(\theta)$. If this is the first recursion, use $p(\theta) := p_o(\theta)$;

2) Given a sequence of measurements y^T, compute the MAP trajectory estimate s^T and its probability $p(s^T)$

using the likelihood function $p(y|s)$. The position estimate output of the algorithm is given by $s_T^* := s_T$;

3) Compute the posterior distribution of the parameters as an integral over $p(s^T)$:

$$p(\theta) := \mathbb{E}_{s^T}\left[p(\theta|y^T, s^T)\right]. \quad (3)$$

We first observe that step 2 is the positioning algorithm which is shared and coupled with the learning algorithm, which is captured in steps 1 and 3. Second, we observe that, unlike the positioning algorithm, the learning algorithm benefits from non-causal conditioning inherent in trajectory estimation, i.e., given y^T, an estimate for s_k for some $k \in \{1, 2, \ldots, T\}$ cannot be less accurate than an estimate for s_T. Non-causal conditioning is a result of the fact that learning need not be done in real-time and can be performed using post-processing of a collection of measurements. [1]

The MAP position estimation in step 2 can be implemented using a number of techniques. In Section III-A we will discretize the state space and run the sum-product algorithm on the resulting trellis. In Section IV we will use particle filters [9] instead. The choice of technique, in general, will depend on the size of the state space.

The computation of the posterior parameter distribution in step 3 may be simplified by performing "hard-decisions" on the position estimates, i.e., approximating the densities $p(s^T)$ as point masses using an empirically derived threshold. This gets around the computationally intensive task of computing the integral. The computation of $p(\theta|y^T, s^T)$ can, in general, be a complex task. One possible approach to efficiently compute this quantity is shown in the Appendix.

III. An application to indoor positioning

A. State space and mobility model

An important practical consideration is the choice of representation of the possible positions s_t of the mobile device and the associated probabilities $p(s_t)$. Options here would be using a continuous state-space and Gaussian densities, a continuous state-space and sampling general continuous densities or by using a discrete state space and a probability mass function on that space. Algorithms corresponding to these representations would be Kalman filters, particle filters and message passing algorithms on a trellis, respectively (see [7] for a survey of all Bayesian filtering techniques). For the indoor positioning problem addressed in this section, we chose the latter mainly due to its simplicity, although for complexity reasons, sampling methods à la the particle filter would most likely be deployed in a final commercial product.

Consider a system formed by a single mobile device moving through a 2-dimensional physical space. A typical example is the localization of a wireless device on the floor of a corporate building. As a running example for the exposition of our indoor positioning work, we use the 4-th floor of Qualcomm's NJRC building. The red lines indicate the positions of the walls and this information is obtained from the building blueprint.[2] The planar state space is discretized according to a square lattice having a resolution of one meter as shown in the upper portion of Fig. 3, hence the set of all grid points is the set of possible positions for the mobile device. Let the set of all grid points be denoted by $\mathcal{G}$. Then, $s_t \in \mathcal{G}$ for all t. Given such a grid, the mobility model $p(s_t|s_{t-1})$ is formed by simultaneously satisfying two conditions:

1) For any two states $i, j \in \mathcal{G}$ $p(i|j) = 0$ if a straight line joining i and j intersects a red line (wall) and $p(i|j) = 1$ otherwise, *and*
2) $p(i|j) = 0$ if the distance between state i and j is greater than M and $p(i|j) = 1$ otherwise.

Finally, the transition probabilities are normalized so that $\sum_i p(i|j) = 1$ for each $j \in \mathcal{G}$. Here M is the maximum distance between two states. Naturally, this maximum distance takes into account the rate of measurement update of the device. In our prototype the measurements update rate is one measurement per second. Taking into account the maximum velocity of a human user is two meters per second, we choose $M = 2$ in our experiments.

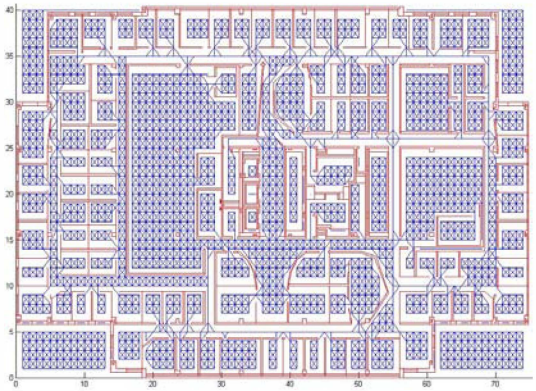

Fig. 3. Quantization of the State Space: NJRC's 4-th Floor.

B. The measurement model

We assume that and the observations are time-wise independent power measurements (also known in the language of WiFi as Received Signal Strength Indicators or RSSIs) of signals transmitted by several WiFi Access Points. In our running example will use six WiFi APs which are well distributed over the floor. The distribution of the power measurements we assume to be Rayleigh, i.e., we model the conditional likelihood function $p(y|s, \theta)$ as exponentially distributed with the mean determined by the parametric ray-tracing model.

Several ray-tracing tools have been developed in both the industry and academia, for example models based on a dominant path [11]. Because the ray-tracing model is at

[1] Furthermore, the positioning and learning tasks may be performed on different devices: the positioning may happen on the mobile device whereas the learning may occur at a central server which can benefit from collecting measurements from multiple devices.

[2] Blueprints for office buildings and malls are easy to obtain. There are a number of companies such as Navteq (www.navteq.com) which maintain databases of a number of commerical buildings in the US.

the heart of the learning algorithm (it is used to compute Eq. 3), we developed our own simplified, flexible and highly computationally efficient model.

The idea behind the model is the observation that the discrete state-space motion model developed in Section III-A can be used for efficiently[3] computing an approximation to the two dominant paths of electromagnetic propagation from a transmitter to every point on the floor. More precisely, a Viterbi algorithm can be used to find the "diffracted path", i.e., the path that bends around wall corners. The other path is the direct path from the transmitter that goes through obstructions. The two paths are illustrated in Fig. 4.

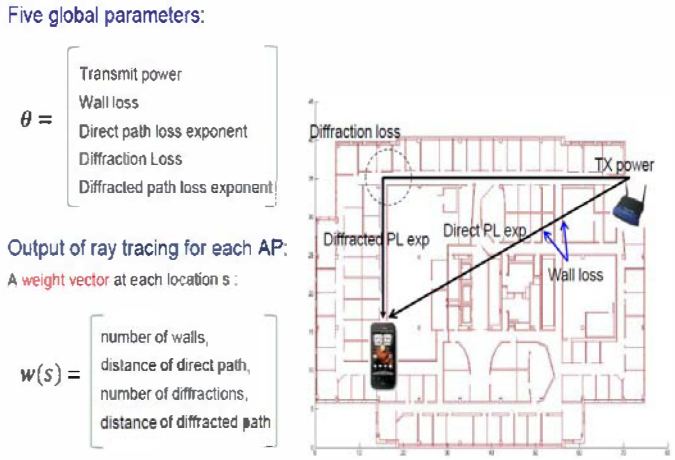

Fig. 4. The parametric ray-tracing model based on two paths: the direct path that goes through obstructions and the diffracted path that goes around wall corners.

The ray-tracing model has only five parameters which are captured by the vector θ in Fig. 4. It is then straightforward to compute the mean of the conditional likelihood function $p(y|s,\theta)$ in terms of the parameter θ and weight vector $\mathbf{w}(s)$, for every location s on the floor. This model is highly amenable to being used in the learning algorithm because it decouples the computation of the *structure* or form of the likelihood function from its *evaluation* for a particular distribution of the parameters θ. In this way, whenever the learning algorithm produces a new update of the posterior distribution of θ, the conditional likelihood functions do not need to be recomputed.

C. Results from an experiment

In this section we report some of the field test results of the joint positioning and learning algorithm. The algorithm was implemented on HTC incredible smartphone mobile devices running a version of the Android operating system. The experiment results that we show have been conducted on the 4th floor of the NJRC building, as used in our running example throughout this paper. The external signals were beacon frames transmitted by six WiFi Access Points (APs) that have already been deployed in the building for WLAN access. The beacon frames were transmitted every 100 ms by each AP but the mobile device scanned the WiFi channel once per second. Within one second the mobile device was capable receiving a beacon frame from each AP that is within RF range.

To highlight the benefits of our learning algorithm, we first illustrate the pitfalls of using non-adaptive, deterministic parameters in the ray-tracing model. We recall that the five parameters are listed in Fig. 4. The value of the parameters was chosen based on an off-line surveying campaign that was conducted in the same venue: we used the Least Squares metric to find the set of parameter values that best explained the collected measurements at the user-input positions. Once the values of the parameters were determined they were used directly in the positioning algorithm (Eq. 2) without the learning steps. The results of the experiment for a 180 meter trajectory taken by the mobile device are shown in Fig. 5.

Fig. 5. The performance of the positioning algorithm using deterministic parameters. The green dots lie on the actual trajectory taken by the mobile and the blue squares represent the estimated trajectory. The estimated trajectory is color-coded to indicate the reliability associated with the estimated position as reported by the algorithm: the darker the color, the higher the reported reliability. The key point is that inaccurately estimated positions are reported as reliable by the algorithm, which is an artifact of model error.

The key observation that we draw from this experiment is that inaccurately estimated positions are reported to be reliable (in other words, to be of high probability) by the algorithm. The reason for this is that the ray-tracing model, which is inherently a coarse approximation for electromagnetic propagation, creates an inaccurate likelihood function which traps the MAP position estimates into locally optimal solutions. The fact that the reliability metrics are *themselves* unreliable means that a learning algorithm that attempts to use the estimated positions will quickly diverge and produce even worse parameter estimates.

Our method of overcoming this limitation is to model the parameters as random variables. By doing so, we establish a way of capturing the uncertainty about our knowledge of both the best parameter values for the model at hand, *as well as the*

[3]It takes an Intel I7 CPU roughly 150 milliseconds to compute the entire conditional likelihood function w.r.t. an anchor.

structure of the ray-tracing model itself. For instance, if we know nothing except the bounds on the range of the parameter values, we can simply choose a uniform distribution as a prior. The effect of choosing such a high-variance prior is that the conditional likelihood function $p(y|s,\theta)$ is smoothed and its support is enlarged by the integration in step 1, Eq. 2. This, in turn, reduces the probability that the positioning algorithm will get stuck in local optima. We have attempted this very experiment, using uniform parameter priors, and the result is shown in Fig. 6.

Fig. 6. The performance of the positioning algorithm using stochastic parameters. In particular, the five global parameters are considered to be independent and uniformly distributed on a wide support. The key point to note is that the inaccurate position estimates are detected by the algorithm and labeled with low reliability metrics (the light-blue positions).

From Fig. 6 we observe that, though the position estimates are only slightly more accurate, the reliability metrics are much more accurate: this is seen in the fact that the inaccurate position estimates are considered unreliable and given a lower probability than in the previous experiment. As a result, the learning algorithm in Eq. 3 can use the estimated positions, weighed with the correct reliability metrics, to compute the posterior parameter densities. We apply the learning algorithm and use the posterior parameter densities to run the positioning algorithm on a new set of measurements taken by another mobile at a later time. The result is shown in Fig. 7

From Fig. 7 we can derive several observations. Firstly, using the uniform parameter priors has the effect of reducing the tail of the CDF curve as compared to the case of deterministic parameters. The large positioning errors are removed because the positioning algorithm is able to correctly determine the reliability of the estimates that it generates. As a result, the outliers can be easily filtered out of the data set and the information displayed to the user. Secondly, the uniform priors cause an increase in the number of small errors in the positioning algorithm. This is because integration of the likelihood function with a high variance prior naturally disturbs position estimates that are already reasonably accurate even with deterministic parameters (there will always be such

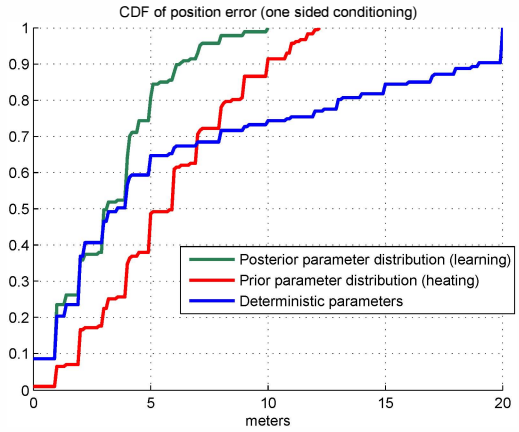

Fig. 7. The cumulative distribution function of position error in meters. The blue curve was obtained using deterministic parameters, the red curve using uniform parameter priors and the green curve using the learned posterior parameter distributions.

locations due to the power of motion constraints alone). Finally, as we expect, the learned posterior parameter densities yield the best positioning accuracy.

IV. Cooperative Positioning

If the mobile devices are able to communicate, the positioning accuracy of the overall system can be improved. If we view each mobile device in the network as a *probabilistic anchor* that is able to send and receive position distributions to/from other devices, we can immediately see that the message passing paradigm naturally allows for the inclusion of such information in the computation of the marginal position distributions for every user. The individual factor graphs are coupled

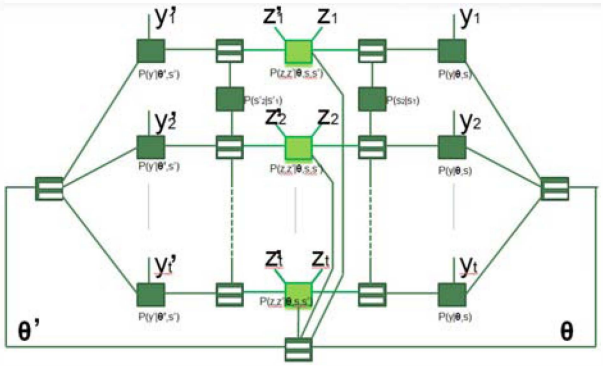

Fig. 8. The factor graph for cooperative positioning

via the inter-device signal measurement likelihood function $p(z,z'|\theta,s,s')$, where (z,z') and (s,s') are the received inter-device signal measuremnents and positions of the two devices, respectively. One practical example of this function is a pairwise distance measurement function using Round-Trip-Time estimation (RTT), which could be represented by a Gaussian distribution centered around the Euclidean distance between s and s'.

In this section we describe some of our early results of applying the cooperative positioning framework to the problem of vehicle positioning. [4] We assume that the vehicles are able to, amongst each other, communicate and observe RTT measurements using Direct Short Range Communication (DSRC) signaling. DSRC is an upcoming technology for inter-vehicle communication that is currently being standardized across the world. Vehicle communication and positioning are subjects of current research at NJRC.

We choose the particle filter method largely due to the physical area of the positioning venue and the high position resolution requirements. For external signals, we use simulated GPS measurements which are assumed to be time-independent[5] sequences of Gaussian densities with fluctuating variances. The model for inter-vehicle distance measurements is a mixture of a Gaussian density centered at the true distance and a uniform density which models multipath errors. The motion model is Markov and assumes that the vehicle can measure (via inertial sensors) a noisy displacement from its previous position (which is 1 second earlier): the noise is assumed to be a Gaussian random variable with zero mean and a variance of 1 meter.

The results of the experiment are shown in Fig. 9. From the figure we observe that using the single-user positioning algorithm can already significantly improve the positioning accuracy over that of stand-alone GPS: errors as high as 30 meters happen with a probability of 0.6, whereas the positioning algorithm reduces this number to ten meters. Cooperative positioning can further reduce this number to five meters.

Though these results are encouraging, they serve only as an illustration of the potential of cooperative positioning. Much further work is required to test the performance of the algorithm in more realistic scenarios, with more realistic models for the GPS measurements and the inertial sensors.

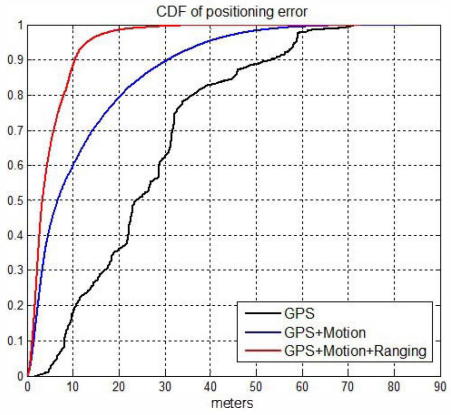

Fig. 9. The CDF of the vehicle position error, averaged over 20 vehicles.

APPENDIX

It is straightforward to show that the posterior density $p\left(\theta|y^T,s^T\right)$ can be iteratively computed using the relation $p\left(\theta|y^T,s^T\right) = c_T p\left(y_T|\theta,s_T\right) p\left(\theta|y^{T-1},s^{T-1}\right)$, where c_T is a normalization constant, and y^0, s^0 are defined to be the empty sets so that, by definition, $p(\theta|y^0,s^0) := p_o(\theta)$, where $p_o(\theta)$ is the prior parameter density. By the independence of ray-tracing parameters $p(\theta|y,s)$ is a product of the individual $p(\theta_i|y,s)$, hence, the key to the overall posterior density computation is the computation of the single-letter quantity $p(\theta_i|y,s)$. Since it does not admit a closed form expression, we approximate it using a Monte-Carlo simulation $\hat{p}\left(\theta_i|y,s\right) = \frac{1}{L}\sum_{l=1}^{L} \frac{p\left(y|\theta_i,\hat{\theta}_l^{-i},s\right)}{\hat{p}(y|s)}$. The variable $\hat{\theta}_l^{-i}$ is a random vector sampled from the joint prior distribution on the set of parameters *not including* θ_i. Note that the term in the numerator is the conditional likelihood function and that it can be evaluated using the method described in Section III-B. The approximate total probability can evaluated as $\hat{p}(y|s) = \frac{1}{L}\sum_l p(y|\hat{\theta}_l,s)$ where $\hat{\theta}_l$ is a random vector sampled from the joint prior distribution of all the parameters. It was found by experiment that $L = 100$ yields sufficiently accurate results.

[4] We emphasize that the results here are for purposes of illustration of the graphical model framework in the network context. The authors in [10] have studied the slightly different problem using virtually the same approach: they have assumed a deployment of infrastructure anchors to aid the distributed algorithm, whereas we assume only that the vehicles have GPS.

[5] This assumption is not necessarily valid in practice in urban environments. The effects of multi-path and satellite geometry are spatially correlated due to the presence of large buildings and other obstructions. Our model is a first step towards understanding the performance of cooperative positioning.

REFERENCES

[1] M Scott Corson, Rajiv Laroia, Junyi Li, Vincent Park, Tom Richardson, George Tsirtsis, *The Internet of Things: Towards Proximity-Aware Internetworking*, IEEE Wireless Communications Magazine, December 2010, pp. 26-33.

[2] H. Wymeersch, J. Lien, and M. Z. Win, *Cooperative localization in wireless networks*, Proc. IEEE, vol. 97, no. 2, pp. 427450, Feb. 2009, special issue on Ultra-Wide Bandwidth (UWB) Technology and Emerging Applications, Invited Paper.

[3] C. Komninakis and R. Wesel, *Joint Iterative Channel Estimation and Decoding in Flat Correlated Rayleigh Fading*, IEEE Journal on Selected Areas in Communications, vol. 19, no. 9, Sep. 2001.

[4] R. Zekavat, R. Buehrer, *Handbook of Position Location:Theory, Practice and Advances*, IEEE Series on Digital and Mobile Communication, Wiley-IEEE Press, November 2011.

[5] G.V. Zaruba, M. Huber, F.A. Kamangar, and I. Chlamtac, *Indoor location tracking using rssi readings from a single wi-fi access point*, Wireless Networks, v 13, n 2, April,2007, p 221-235, 2007.

[6] Hui Wang, *Bayesian radio map learning for robust indoor positioning*, Conference on Indoor Positioning and Indoor Navigation (IPIN), 21-23 Sept. 2011

[7] Z. Chen, *Bayesian Filtering: From Kalman Filter to Particle Filters, and Beyond*, available online at *http://www.damas.ift.ulaval.ca/_seminar/filesA11/10.1.1.107.7415.pdf*.

[8] F.R. Kschischang and B.J. Frey and H.-A. Loeliger,*Factor graphs and the sum-product algorithm*, IEEE Trans. Inform. Theory, vol. 47, pp. 498-519, Feb. 2001.

[9] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, *A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,* IEEE Transactions on Signal Processing, Vol. 50, No. 2, Feb 2002.

[10] V.N. Ekambaram, K.Ramchandran, *Distributed High Accuracy Peer-to-Peer Localization in Mobile Multipath Environments*, Global Telecommunications Conference (GLOBECOM 2010), December 2010

[11] R. Wahl, G. Wlfle, *Combined Urban and Indoor Network Planning Using the Dominant Path Propagation Model*, 1st European Conference on Antennas and Propagation (EuCAP 2006), Nice, November 2006.

[12] Y. Shen and H. Wymeersch and M. Z. Win, *Fundamental limits of wideband localization Part II: Cooperative networks*, IEEE Trans. Inf. Theory, vol. 56, no. 10, pp. 4981-5000, Oct. 2010.