WiFi Position Estimation in Industrial Environments Using Gaussian Processes

Felix Duvallet Ashley D. Tews
Autonomous Systems Lab, CSIRO
PO Box 883; Kenmore, QLD 4069; Australia
Email: firstname.lastname@csiro.au

***Abstract*—The increased popularity of wireless networks has enabled the development of localization techniques that rely on WiFi signal strength. These systems are cheap, effective, and require no modifications to the environment. In this paper, we present a WiFi localization algorithm that generates WiFi maps using Gaussian process regression, and then estimates the global position of an autonomous vehicle in an industrial environment using a particle filter. This estimate can be used for bootstrapping a higher-resolution localizer, or for cross-checking and localization redundancy. The system has been designed to operate both indoors and outdoors, using only the existing wireless infrastructure. It has been integrated with an existing laser-beacon localizer to aid during initialization and for recovery after a failure. Experiments conducted at an industrial site using a large forklift-type autonomous vehicle are presented.**

I. Introduction

As autonomous vehicles play more prominent roles in industrial environments, the importance of accurately determining the location of the vehicle in the environment becomes increasingly crucial. For safe and reliable performance, an autonomous vehicle should not rely on a single localization solution alone, as a failure at a critical point could bring down the entire system. Thus, there is a need for redundant position estimators that can be utilized in the event of a failure in a primary localization subsystem. There are many available approaches for vehicle localization including using GPS, artificial or natural landmarks, and visual maps. However, many of these approaches require expensive equipment or considerable effort in sensor processing or environment modeling.

In contrast, WiFi devices are cheap, lightweight, and relatively low-power [1], and WiFi access points are becoming increasingly ubiquitous in many of the environments robots operate in (offices, schools, industrial work-sites). A WiFi localizer can be an integral part of a robust navigation system as it has several advantages:

- It is an efficient method for global localization since access points are uniquely identifiable;
- WiFi can be used to localize both indoors and outdoors;
- No modifications to the environment are required, instead relying on the existing infrastructure;
- Completely passive localization is possible by "sniffing" wireless traffic; and
- WiFi signals do not require line-of-sight. As such, they can be used even in the presence of obstructions that would block laser range-finders or cameras.

In this paper, we utilize an existing 802.11 wireless infrastructure to estimate the location of an autonomous vehicle within an industrial environment. Although WiFi localization has been used extensively in the literature for person tracking [2] and location-aware computing [3], it is unclear whether a WiFi localization system alone will ever provide the accuracy required for autonomous vehicles operating in industrial environments. However, as a secondary localization system, WiFi signal strength information can be used to generate a coarse global position estimate which can be used to bootstrap the primary system.

Although GPS can sometimes provide this secondary estimate, coverage is often unavailable inside buildings or when the view of the sky is obstructed, and incorrect estimates can occur due to multipathing. Durrant-Whyte et al. have determined that the physics of GPS, combined with insufficient coverage in Australia, prevent it from being reliable enough to be used as a stand-alone navigation sensor in industrial scenarios [4].

We have designed the WiFi position estimator for use in one of several scenarios. Firstly, it can be used as an initial estimate during the initialization of a primary localization system. This can speed up early convergence and reduce the possibility of multiple hypotheses due to environment symmetries. Secondly, the WiFi localizer can be polled when the primary localizer's confidence drops below a threshold, indicating that it may be lost. In this case, the WiFi localizer can again provide a position estimate to re-seed the local position estimator. Lastly, the estimate from the WiFi localizer can be used as an independent estimate to detect possible anomalies occurring in other localization systems.

Our approach to WiFi localization is based on the work of Ferris, Hähnel, and Fox [5] and is tailored to industrial environments. The two main components of the WiFi localization system are a WiFi map generated using Gaussian processes, and a localizer which uses Bayesian filtering. The resulting system is analyzed against ground truth measure-

Fig. 1. The autonomous Hot Metal Carrier (HMC)

ments to determine the expected accuracy. Finally, the system is integrated with an existing laser-beacon localizer to provide an estimate used during initialization or after a failure in the primary localizer.

Our paper continues as follows: Section II discusses related WiFi-based localization research, and Section III briefly describes Gaussian process regression. Sections IV and V discuss the mapping and localization phases, respectively. We present experimental results in Section VI, and finally close with a discussion of our future plans and some conclusions.

II. Related Work

Due to the relatively coarse accuracy of most existing WiFi position estimation systems, many have been designed with smart environments and context-aware computing in mind. Two broad categories of localization systems using WiFi signal strength have emerged (algorithms using other features such as time or angle of arrival are not covered here for brevity):

- **Modeling**, in which an explicit model of the expected signal strength is determined.
- **Mapping**, where labeled training data is gathered to generate a map which is used during localization.

Although systems based on modeling methods have shown to be effective at determining a user's location [3], [6], they often rely on extensive knowledge of the environment (such as Access Point positions or the location and number of walls) to predict the signal strength [7], [8]. In one approach, the parameters of the proposed signal strength attenuation model are empirically determined, and this information is used to triangulate the user's position around a university campus [3].

Mapping methods introduce a calibration step, where labeled data is gathered prior to the system's operation. Various properties of the received signal can be stored for use during localization, including raw signal strength [2], histograms [9], and Gaussians [1]. For example, one approach exhaustively compares the input signal strength against the training data for all locations to determine a set of nearest neighbors, which are interpolated to generate a position estimate [2].

Given enough training measurements and even coverage, the mapping-based methods are generally quite accurate, and have the advantage of using empirical data which may be hard to model using analytical models. However, some of the calibration methods presented require an extensive data collection step. In one approach, WiFi data is gathered every few meters in four different orientations [2].

We wish to take advantage of the benefits that the mapping methods provide without a need for an intensive data collection process. Fortunately, Gaussian process-based mapping methods provide a solution to this problem (as well as many others). Several signal strength-based localization systems that utilize Gaussian processes have already been developed, and this method has proved to be very well-suited for the problem domain [5], [10]. Gaussian processes (GP) and their advantages are introduced in the next section.

III. Gaussian Process Regression

Gaussian processes offer many advantages that make them suited for a localization system that utilizes WiFi signal strength [5]. Firstly, they are non-parametric, so a model that can correctly fit the data is not required [11]. Because GPs place a prior over the distribution of functions, many highly non-linear models can emerge from GP regression [11]. Secondly, they are continuous. Training data does not need to be gathered at regularly spaced intervals, nor does the environment necessarily need to be discretized during localization. Training data can come from arbitrary points, and predictions can be generated for any point in the environment. Furthermore, the predictions will use a maximal amount of the training information, as opposed to a small number neighbors [11]. Thirdly, GPs correctly handle uncertainty in both the process and the estimation. This is especially useful because WiFi signal strength measurements are very noisy due to various phenomena such as diffraction, scattering, reflection, and absorption [8].

A Gaussian process essentially defines a probability distribution over *functions*. We wish to generate a function $f(\mathbf{x}_*)$ that makes predictions for all possible inputs $\mathbf{x}_*$. We use a training data set $\mathcal{D} = \{(\mathbf{x}_i, y_i) \,|i = 1, \ldots, n\}$ consisting of n observations in $\mathbb{R}^d$ drawn from a noisy process $y_i = f(\mathbf{x}_i) + \varepsilon$, where ε is additive Gaussian noise with zero mean and variance σ_n^2. For notational simplicity, the inputs of the training set are grouped into a $d \times n$ matrix $\mathbf{X}$, and the observations y_i are grouped into a vector $\mathbf{y}$.

To generate $f(\mathbf{x}_*)$, GPs rely on a covariance function kernel $k(\mathbf{x}_p, \mathbf{x}_q)$ that specifies how the values at different points are correlated. Generally, points with inputs $\mathbf{x}$ that are close to each other are likely to have similar target values y. The user has many choices for this kernel (see [11] for examples), and we have chosen the popular squared exponential kernel:

$$k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x}_p - \mathbf{x}_q)^{\mathrm{T}}\mathbf{M}(\mathbf{x}_p - \mathbf{x}_q)\right), \quad (1)$$

where $\mathbf{M}$ is a matrix whose diagonal elements are set to the respective length scales (ℓ_i): $\mathbf{M} = \mathrm{diag}(\boldsymbol{\ell})^{-2}$. The hyperparameters σ_f^2 and $\boldsymbol{\ell}$ are the signal variance and characteristic length scales, respectively.

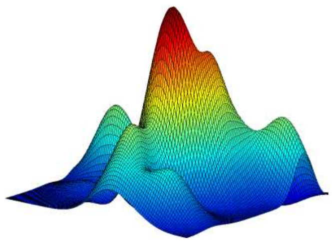


Fig. 2. Signal strength mean prediction – eq. (4b).

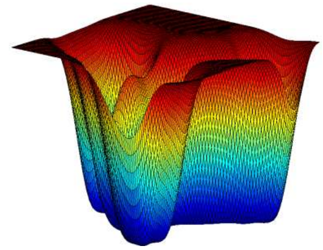


Fig. 3. Signal strength variance prediction – eq. (4c).

Because y is actually a noisy observation of the true value $f(\mathbf{x})$, we must add a term to account for observation noise in the covariance function:

$$\operatorname{cov}(y_p, y_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq} \tag{2}$$

where δ_{pq} is 1 if $p = q$ and 0 otherwise. Note that the covariance between the outputs is written as a function of the inputs, emphasizing the non-parametric nature of Gaussian process regression.

Again for notational convenience, we rewrite the covariances as matrices and vectors, such that K is a matrix of the covariances evaluated at all pairs of training points, and $\mathbf{k}_*$ is a vector of the covariances between $\mathbf{x}_*$ and the n training inputs. This compact form allows us to rewrite equation (2) into:

$$\operatorname{cov}(\mathbf{y}) = K + \sigma_n^2 I \tag{3}$$

We can then generate the posterior distribution over functions for arbitrary points $\mathbf{x}_*$ given the training data $\mathbf{X}$ and $\mathbf{y}$:

$$p(f(\mathbf{x}_*) \mid \mathbf{x}_*, \mathbf{X}, \mathbf{y}) \sim \mathcal{N}\left(\mu_{\mathbf{x}_*}, \sigma_{\mathbf{x}_*}^2\right), \text{with} \tag{4a}$$

$$\mu_{\mathbf{x}_*} = \mathbf{k}_*^{\mathrm{T}} \left(K + \sigma_n^2 I\right)^{-1} \mathbf{y} \tag{4b}$$

$$\sigma_{\mathbf{x}_*}^2 = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^{\mathrm{T}} \left(K + \sigma_n^2 I\right)^{-1} \mathbf{k}_* \tag{4c}$$

This distribution provides us with the necessary likelihood model that will be used for localization in Section V.

Because the covariance function is such an integral part of GP regression, the $d + 2$ free hyperparameters (σ_f^2, σ_n^2, and $\boldsymbol{\ell} = \{\ell_{1\ldots d}\}$) greatly affect the characteristics of the predicted signal. Fortunately, these hyperparameters can be learned by maximizing the log marginal likelihood of the observations conditioned on the hyperparameters [11].

IV. Gaussian Processes for WiFi Map Creation

To create a WiFi map, we must first collect our training sample $\mathcal{D}$, consisting of signal strength measurements (y) associated with positions in the environment ($\mathbf{x}$). These training points need not be grid aligned or sampled in any specific way, which enables us to gather the data required to train a map during normal vehicle operation. However, any remote areas that will be of interest during localization should be covered (at least partially); a simple user check of the training data is sufficient to determine adequate map coverage.

The map is learned off-line. Although the MAC address of each Access Point (AP) is known, its location is not used. The parameters of the Gaussian process are learned, and signal strength estimates are generated for a uniformly-spaced grid of points in the environment. This process is repeated once for each access point (GP parameters are re-learned for each AP to account for differences between them), generating as many maps as there are detected access points in the environment. Although this map-making process is relatively slow (a few hours on a standard desktop computer), it is performed off-line and only done once. An example WiFi map for one access point is shown in Fig. 2 (mean), and Fig. 3 (variance). As expected, the mean generates a prediction for signal strength, and the variance is lower in areas where more training data was available.

V. Sequential Monte Carlo Location Estimation

The localization system uses a Sequential Monte Carlo Localization algorithm [12] consisting of 1000 particles which are initially distributed uniformly at random over the entire environment. Only particle positions are tracked, as signal strength from an omnidirectional antenna does not provide bearing information. During localization, incoming WiFi packets change the likelihood of all particles. For each particle, the predicted signal strength mean and variance are extracted from the generated map associated with the corresponding access point. These are used to calculate the measurement likelihood:

$$p(z_t | \mathbf{x}_*) \propto \frac{1}{\sqrt{2\pi\sigma_{\mathbf{x}_*}^2}} \exp\left(-\frac{(z_t - \mu_{\mathbf{x}_*})^2}{2\sigma_{\mathbf{x}_*}^2}\right), \tag{5}$$

where z_t is the received signal strength at time t, and $\mu_{\mathbf{x}_*}$ and $\sigma_{\mathbf{x}_*}^2$ are the mean and variance at position $\mathbf{x}_*$ predicted by equation (4).

In this analysis, we have chosen not to use vehicle motion information under the assumption that for our generic WiFi localization system, this information may not be available. Therefore, in order to keep our WiFi-based system completely independent, the motion model is instead replaced by random particle motion. Periodically, each particle is randomly perturbed, effectively spreading the particle cloud in all directions. Although some of the particles are moving in the opposite direction to the vehicle's actual motion, in

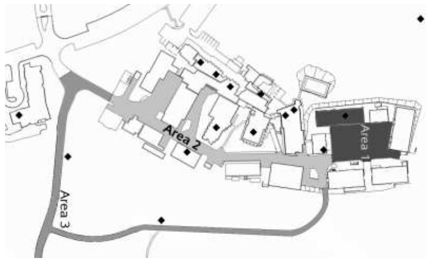


Fig. 4. Map of the test site, spanning approx. 250m by 400m. Access points locations are shown with black diamonds.

practice, WiFi packets are received often enough that the likelihood model combined with particle resampling correctly track the vehicle location.

The system also needs to be robust to the kidnapped robot problem. Although it is unlikely that anyone will try to kidnap a 20-ton industrial vehicle, there can be cases where WiFi information may not be available for a period of time, or the WiFi location estimate could be incorrect. For this reason, a small number of particles are randomly dispersed in the environment instead of being propagated. These particles are usually discarded during the next evaluation step, but they are critical for the robustness of the WiFi position estimation system.

VI. Experimental Results

We have conducted experiments to evaluate the accuracy of the WiFi-based localization system alone, and we have also integrated it with an existing localization system so that it can be used during initialization or for failure recovery. We present two experiments in this section. The first is a demonstration of the WiFi localizer accuracy. After generating a map using training data and GP regression, the WiFi localization system is tested and compared against ground truth data obtained from a highly precise laser-beacon localizer. In the second experiment, we purposefully create faults in the laser localizer and autonomously use the global WiFi position estimate as a seed for the laser localizer to re-initialize itself.

A. Experimental Setup

The test area is an industrial work site approximately 250m by 400m (shown in Fig. 4) which is part of the Queensland Center for Advanced Technologies (QCAT). Two types of buildings are situated on site: multi-storied offices and large industrial sheds. There is also an area without buildings surrounded by vegetation. Industrial vehicles such as forklifts and trucks operate on the roads around the site, and there is also considerable pedestrian traffic.

A wireless network consisting of many different access points provides continuous network access to QCAT staff and mobile robots over the entire site. The majority of the access points are located indoors for good coverage in the offices (although those APs can still be detected outdoors), and there are also several access points located outdoors at strategic locations. Most access points are omnidirectional, while some are directional to cover narrower regions such as along roadways.

In both experiments, the site has been partitioned into three areas with distinct features based on environment type and WiFi coverage.

- **Area 1** consists of a large open area surrounded by industrial sheds, one of which the vehicle can drive inside. Several access points are in this region, including several outdoor omnidirectional access points.
- **Area 2** consists of a road with buildings on both sides. Although many access points are in this area, most are indoors and do not offer a wide coverage area. GPS reception is poor in this area due to a very narrow view of the sky and multipathing from buildings.
- **Area 3** consists of a road surrounded by vegetation. Very few wireless access points are in this area, but there are several directional access points that cover the road only. GPS reception in this area is generally reliable.

The primary test vehicle is the Hot Metal Carrier (HMC), a 20-ton forklift-type vehicle (shown in Fig. 1) that has been automated by the Autonomous Systems Lab at CSIRO [13]. This type of vehicle is used in aluminum smelters, where it transports molten aluminum in a large metal crucible. This crucible weighs two tons and can transport eight tons of molten aluminum at 700°C. Because the vehicles are performing this task repeatedly, they are a prime target for automation. However, because a potentially non-localized vehicle presents obvious safety risks, the reliability of the localization system is of critical importance.

A laser beacon-based localization system for the HMC was previously developed and found accurate to well within a meter [13]. It utilizes reflective beacons that are installed in the environment at surveyed locations. The laser-beacon localizer does not search the global space to find possible locations, and it requires an initial estimate of its location upon start-up (which may occur inside or outside a shed). It may also require re-initialization if it becomes lost. The WiFi localizer is ideally-suited to aid the laser system in these situations as it does not share any common failure modes.

WiFi signal strength is acquired using a laptop equipped with a PCMCIA Netgear WiFi card. The card has been modified to use an external omnidirectional antenna which is mounted on the top of the HMC. Kismet [14], used in its passive mode, sniffs wireless packets' MAC addresses and signal strengths, passing this information to the localizer. DDX, a distributed shared memory architecture, is used to share information between the various software systems on the vehicle [15].

In our experiments, the vehicle was driven manually to gather WiFi data for the mapping and localization data sets. Although both cover the same general regions (roads and sheds), the two data sets were taken many days apart and at different times of the day. Additionally, the exact path taken

TABLE I
WIFI LOCALIZATION ACCURACY

Area	Localizer Error (m)		Median num. of visible APs	Distance traveled (m)	Time (sec)
	Mean	Median			
1	7.4	5.7	3	113	137
2	7.8	5.4	5	220	230
3	22.3	13.5	3	338	270

was slightly different, as the vehicle could have been driving on the opposite side of the road or in the opposite direction that the training data was collected. However, because of the continuous nature of the GP regression, this does not greatly affect us.

B. WiFi Location Estimation Accuracy

For each of the three areas defined in Section VI-A, WiFi data was collected using the HMC. The accuracy of the localizer is examined in each of the three regions, and is summarized in Table I. The WiFi localizer estimate is taken as the mean of the particles, and ground truth is obtained using the laser localizer. The WiFi localizer error is computed at a rate of approximately 60Hz using the Euclidean distance as the error metric. The approximate distance traveled is reported (taken from wheel odometry), as well as the time of each run. During each of the three data runs, the vehicle was stopped once or twice.

The mean and median of the position errors vary in each region. This is due to the nature and distribution of the access points available in each of the three regions. For example, Area 1 contains several omnidirectional access points located on top of buildings. A histogram of the errors in Area 1 is shown in Fig. 5. Area 2 has a large number of access points, but many are within buildings, greatly reducing their range. Area 3 has the greatest error of the three regions. We suspect this is because the access points in this region are spaced far apart and directional. However, because Area 3 has few buildings and other obstructions, GPS is readily available. In the event that a secondary position estimate is required, GPS can be used instead of the WiFi-based estimate.

The number of visible access points is tracked by determining how many unique APs in the map have been detected (sent a packet) at least once in the past 3 seconds. This changes over time and is correlated with the properties of the different regions. Fig. 6 shows a histogram of the number of visible APs during a run in Area 2. In most cases, the localizer is more accurate when more access points are visible. Note that this is not due to triangulation (which would be used in a modeling technique), but the fact that the likelihood models complement each other. In practice, an accurate location estimate can be achieved using two or three access points.

C. System integration

The global WiFi position estimator has been integrated with the laser-beacon localizer on the HMC. In the event of

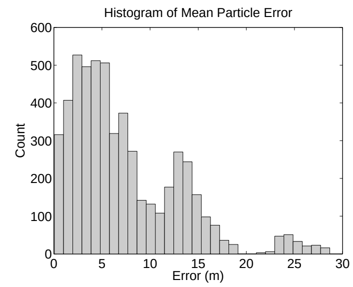


Fig. 5. Histogram of the mean error in Area 1

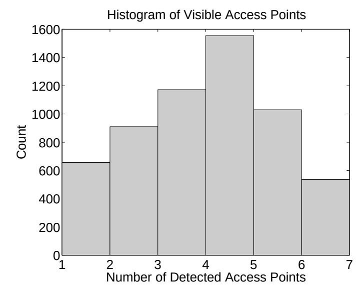
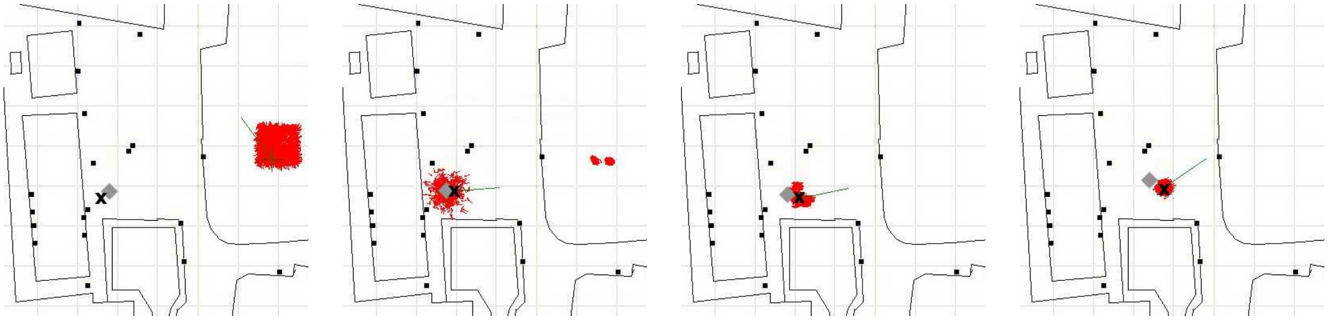


Fig. 6. Histogram of the number of simultaneously visible distinct access points in Area 2 (using a three-second window)

a localizer failure (which can be triggered manually during testing), the laser localizer uses the global estimate from the WiFi system as a seed for an initial estimate region in which to search. The laser-localizer particles are uniformly distributed around the seed point (vehicle headings are randomly assigned since no heading information is provided by the WiFi system), where the size of the region is specified by the WiFi system's localization uncertainty. This uncertainty is inversely proportional to the product of the number of visible access points and the accepted packet rate.

In the next experiment, the laser-beacon localizer was set to an erroneous location and the WiFi localizer was used to bootstrap it back to the approximately correct area where it could re-localize. A time sequence of the event is shown in Fig. 7. The laser localizer also uses a particle filter which is shown in red in the figure. Initially, a failure is induced by moving the particles to a constrained region (the square) in Fig. 7(a). In Fig. 7(b), bootstrapping has occurred and particles are moved to a new seed area distributed around the WiFi position estimate. In Figs. 7(c) and 7(d), the vehicle has moved and the laser localizer has locked onto the correct vehicle location.

This simple example demonstrates how the bootstrapping process works. We have conducted many successful trials of this technique onboard the HMC which removes the constraint of having an *a priori* location estimate for seeding the laser-beacon localizer, or having to wait for the system to converge if the particles are initialized over the entire environment. In environments with high symmetry, convergence may not occur without a global position estimate. Fast and accurate con-



(a) Localizer failure (triggered) (b) WiFi estimate merged (c) Bootstrapping in progress (d) Laser localizer converged

Fig. 7. Time sequence of a re-initialization in Area 1. The HMC (marked with an 'X') is moving towards the right of the compound. The WiFi localizer estimate is shown as the large diamond, and the laser-beacon localizer particles are shown in red. The grid spacing is 10m.

vergence is necessary for autonomous vehicles operating in industrial environments, since they should not move without knowledge of their position.

Although the laser localizer generally converges very quickly, there are cases where the bootstrapping can fail. These are usually due to the inherent limitations of the laser-beacon system, such as when a small number of beacons are detected, or when the environment is symmetric with respect to orientation (for example, a road with very regular beacon placements). If the localizer fails to converge, a new WiFi estimate can be used for re-initialization once again.

VII. Conclusions and Future Work

We have demonstrated a mapping-based WiFi position estimation system that can be used either independently or as an extra layer for another higher-resolution localizer. Initially, the mapping phase uses a set of labeled training data and Gaussian process regression to create WiFi maps. The localization phase uses these maps as the sensor likelihood model for the particle filter (with signal strength as the only sensor input).

The WiFi system has been demonstrated on an autonomous Hot Metal Carrier in an industrial environment. The system has been integrated with the existing localization infrastructure so that it can provide a global estimate during (re-)initialization, or resolve ambiguities due to environment symmetries. Furthermore, the results of the two localization solutions can be cross-checked to increase redundancy. The lack of common failure modes in the two systems increases the fault tolerance of the overall system.

Future work on the localization system will focus on refining the integration of the WiFi localizer and the laser localizer, so as to provide as much redundancy as possible. Additionally, we will investigate methods to continuously adjust the map during localization to offset long-term changes in the wireless network. This would make the WiFi localization system robust to changes in building and wireless infrastructure.

Acknowledgments

The authors gratefully acknowledge Stephen Nuske, Paul Flick, and Polly Alexander for their help collecting data. Cédric Pradalier helped develop parts of the software infrastructure, and Frederic Moster created the Kismet interface software. The authors also wish to thank Robert Zlot, Stephen Nuske, and the anonymous reviewers for their comments.

References

[1] A. Howard, S. Siddiqi, and G. S. Sukhatme, "An experimental study of localization using wireless ethernet," in *Proc. of International Conf. on Field and Service Robotics*, July 2003.

[2] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. of IEEE INFOCOM*, 2000.

[3] A. Smailagic and D. Kogan, "Location sensing and privacy in a context-aware computing environment," *Wireless Communications, IEEE*, vol. 9, no. 5, pp. 10–17, October 2002.

[4] H. Durrant-Whyte, D. Pagac, B. Rogers, M. Stevens, and G. Nelmes, "An autonomous straddle carrier for movement of shipping containers," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 3, pp. 14–23, 2007.

[5] B. Ferris, D. Hähnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proc. of Robotics: Science and Systems*, August 2006.

[6] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo, "Self-Mapping in 802.11 Location Systems," in *Proc. of UbiComp 2005: Ubiquitous Computing*, September 2005.

[7] M. Hassan-Ali and K. Pahlavan, "A new statistical model for site-specific indoor radio propagation prediction based on geometric optics and geometric probability," *IEEE Trans. Wireless Commun.*, vol. 1, no. 1, pp. 112–124, January 2002.

[8] W. Braun and U. Dersch, "A physical mobile radio channel model," *IEEE Trans. Veh. Technol.*, vol. 40, no. 2, pp. 472–482, 1991.

[9] M. Youssef, A. Agrawala, A. Shankar, and S. Noh, "A Probabilistic Clustering-Based Indoor Location Determination System," UMD Computer Science Department, Tech. Rep. CS-TR-4350, 2002.

[10] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann, "GPPS: A Gaussian Process Positioning System for Cellular Networks," in *Advances in Neural Information Processing Systems*, 2003.

[11] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[12] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, "Particle filters for mobile robot localization," *Sequential Monte Carlo Methods in Practice*, 2000.

[13] A. Tews, C. Pradalier, and J. Roberts, "Autonomous hot metal carrier," in *Proc. of International Conf. on Robotics and Automation*, April 2007.

[14] Kismet wireless. http://www.kismetwireless.net.

[15] P. Corke, P. Sikka, J. Roberts, and E. Duff, "DDX: A distributed software architecture for robotic systems," in *Proc. of the Australian Conf. on Robotics and Automation*, December 2004.