

Automatic Online Localization of Nodes in an Active Sensor Network

Alex Brooks, Stefan Williams, and Alexei Makarenko
ARC Centre of Excellence for Autonomous Systems
School of Aerospace, Mechanical, and Mechatronic Engineering
University of Sydney
NSW Australia
{a.brooks,s.williams,a.makarenko}@acfr.usyd.edu.au

Abstract—Localization of nodes within a sensing network is a fundamental requirement for many applications. This paper proposes a method by which sensors self-localize based on their uncertain observations of other nodes in the network, using both Monte Carlo and Kalman Filtering techniques.

The proposed methods are demonstrated in a laboratory environment where stationary camera nodes self-localized in real-time by observing Pioneer robots moving about within their field of view. The robots take observations of surveyed beacons in the environment and provide estimates of their poses to the rest of the network.

I. INTRODUCTION

Sensor networks are an active area of research with a large number of potential applications [1] [2]. The basic approach is to distribute a set of small, cheap sensors within an environment. The sensors extract and process information in a distributed manner and share their resulting estimate of the state of the world using Distributed Data Fusion Techniques [3]. An active sensor network also consists of actuated sensors, possibly in the form of mobile robots, which can direct themselves towards areas that interest the network. For the remainder of this paper an element of a sensor network, active or otherwise, will be referred to as a node.

Localization is a fundamental problem for active sensor networks, both for stationary and mobile nodes. A possible localization method for stationary sensors is to manually localize them a priori, for instance with a tape measure, however this is tedious and extremely inflexible. Furthermore, to localize several sensors in the same frame of reference is difficult unless the frame of reference is obvious. Consider instead a system which can handle not only the removal of sensors, for example through damage or power loss, but one in which sensors can also be added at arbitrary positions and at arbitrary times, automatically coming on-line, taking some time to self-localize in the frame of reference of the rest of the system, and then adding information back into that system. The ultimate goal of such a system would be to have a localization method so robust that one could throw a sensing node through a window into an unseen room, or perhaps drop it from a plane, and have it self-localize and come on-line automatically. This paper presents an approach to realizing that goal.

A great deal of attention in the literature has been devoted to the subject of mobile robot localization, and in many domains there exist systems which perform this task very successfully [4] [5].

The mobile robot localization problem has sometimes been divided into two categories: the tracking problem, in which an initial pose for the robot is known, and the global localization problem, in which the robot must localize from scratch [6]. In the case where large numbers of sensors are to be distributed in a fairly care-free manner, providing initial estimates of each sensor's pose would be an undesirable burden, therefore the more difficult problem of global localization must be solved.

Thrun et. al. have shown effective solutions to the global localization problem in the context of a mobile robot in an environment for which it has a map [6], using particle-filtering techniques. The task of localizing immobile nodes is slightly different, however, in that a static map is less useful. Unless each point on the map can be uniquely identified by nearby features and the node's sensors are capable of extracting those features from a single vantage point, the node cannot unambiguously determine its pose from the map. However given a dynamic map, including knowledge of how the map evolves over time, an immobile sensing node can use similar particle-filtering based techniques to localize itself.

The approach presented here is to use a network of nodes, at least one of which is well-localized, to localize one-another by using the well-localized nodes as the features of a dynamic map. The requirements for this system are:

- 1) Nodes that are not yet localized must be able to sense the well-localized nodes in some capacity. Possible sensing modalities include vision, laser, acoustic, and Ethernet signal strength [7].
- 2) Well-localized nodes must be able to communicate their pose estimates regularly to nodes in need of localization.
- 3) There must be relative motion between nodes that are not localized and nodes that are.

Essentially, mobile robots capable of remaining localized are used to make the task of localization easy for a large number of cheap nodes. Once the cheap nodes are localized the localization capabilities of the mobile robot become redundant as the sensing network will be able to localize new robots that

are introduced into the system.

The remainder of this paper is organized as follows: Section II describes the techniques used to perform the localization. Section III describes how the system is implemented in a laboratory setting in which stationary cameras localize themselves based on observations of one or more mobile robots. The robots are performing beacon-based localization with SICK lasers. Section IV outlines the results of this implementation, Section V provides some discussion of this system and Section VI concludes.

II. LOCALIZATION TECHNIQUES

In the first stage of the localization procedure the probability density of the location of the sensor is represented with a particle filter. A particle filter representation is chosen because non-gaussian and multi-modal distributions are expected. Grid-based methods are rejected because it is anticipated that there will be large areas of space in which the probability of finding the sensor are close to zero. The details of Monte Carlo techniques are not discussed here, instead interested readers are directed to [8], [9] and [10].

The general problem is to discover values for the sensor's pose in terms of x , y and z , plus roll, pitch and yaw for sensors for which direction is meaningful. Upon initialization it is assumed that nothing is known about the sensor's pose except the size of a map (in x , y and z) in which the sensor is guaranteed to lie. This initial ignorance is represented with a uniform distribution over the map's area, which is sampled to produce a set of particles. Constraining the size of the map reduces the number of particles required to approximate the distribution, increasing computational efficiency.

When a sensor observes a robot it computes an estimate of the pose of every robot it knows about at the time of the observation, using the pose estimates that are communicated by the robots. The weighting of each particle is then updated according to the likelihood that the sensor pose represented by that particle is true, given the observation and the robots' pose estimates. This is done by calculating an expected observation of each robot, and comparing these expected observations against the actual observations, given the likelihood function of the sensor and the uncertainty in the robot's pose.

To explain, consider the example of an elevated camera with known height, roll, and pitch observing a perfectly-localized robot on the ground. These three variables are artificially fixed for the purposes of explanation to ensure that the state can easily be displayed in a two-dimensional image. Figure 1 shows the situation after repeated observations (in azimuth and elevation) of a single stationary robot. For each particle a predicted observation can be computed and compared against an actual observation. Particles that are not good at predicting observations are given lower weightings and removed on resampling. The resulting probability distribution becomes a circle, with cameras facing inwards.

When relative motion is introduced, the weights of the particles that can best predict that motion will be increased. Consider again the example of the camera with fixed height,

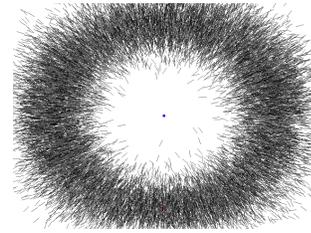


Fig. 1. View from above of particles clustering in a circle around a stationary robot after multiple observations. Particles are represented by short black lines, pointing in the direction of the camera pose they represent. The blue square represents the robot's location. The true location of the sensor is in the bottom part of the ring.

roll, and pitch. Each observation of the robot defines a circle, centered on the robot, in which particles are reinforced. As the robot moves, different observations are produced which define circles that intersect only at the true pose of the sensor. Particles located elsewhere will be removed. The six-dimensional problem where x , y , z , roll, pitch and yaw are all unknown is only different in that more particles are required to represent the distribution.

One advantage of particle filters is their resistance to outliers. A false observation that occurs nowhere near a real robot will cause the weightings of all likely particles to drop substantially. Since the particle weightings are always re-normalized to sum to unity, only the relative weightings of the particles are important. Since unlikely observations don't affect the relative weightings very much, they are largely ignored.

Note that the sensor needn't know which robot it is observing if particles are simply re-weighted according to every possible robot-observation pairing. When particles are re-weighted from an incorrect pairing the weightings will drop drastically, however they will drop by a similar amount for all particles, as for false observations. Particles that can give rise to any observation will have their relative weightings increased, and particles that can give rise to no observations will have their relative weightings decreased. An example is shown in Figure 2, where two stationary robots are observed with the same camera as in the previous example. As soon as either of the robots starts to move differently from the other the ambiguity will be resolved and a unimodal distribution will form.

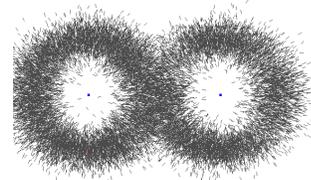


Fig. 2. View from above of particles clustering in a two circles around two stationary robots. The sensor is only observing the left-hand robot.

For the localization of the static nodes in the network, the resampling process is complicated somewhat by the fact that there is no process noise associated with the states to be estimated. In standard robot localization problems new samples are drawn from the previously computed sample set, where the probability of drawing each sample is proportional to its weight [5]. Since the previous sample set is not uniformly weighted, some particles will be sampled more frequently than others. This results in sample impoverishment, with many particles representing identical poses. This sample impoverishment is obviated to some degree because newly drawn samples are progressed through the motion model, with its associated process noise, which spreads the particles out. However in the absence of process noise the sample impoverishment becomes extremely severe because identical particles can never split up. The undesirable effect of this is that probability distributions can only be represented by a weighted sum of the initial particles, since particles can never move.

Liu and West discuss methods for overcoming the problem of estimating fixed parameters using Monte Carlo methods [11]. One approach is to add artificial process noise after every observation. This has the effect of jiggling the particles around to explore the density function better. The drawback of this approach is that it is a poor approximation for parameters which are really fixed, such as the sensor's pose. Every time artificial noise is added the estimate becomes more dispersed, increasing convergence times by throwing away precious information at every observation.

Another approach is to use kernel smoothing when resampling. The set of weighted particles approximate an underlying probability distribution which can be recovered by convolving the set of particles with a kernel [12]. Resampling from this distribution avoids the sample impoverishment problem. This process can be achieved practically by resampling in the usual manner, adding a random perturbation in pose drawn from the kernel function to each resampled particle. Since convolution is a loss of information, this approach suffers from the same problems as the previous approach, namely that the distribution spreads out too much. In fact the two approaches are equivalent if resampling is performed at every time step.

Liu and West also discuss a method for dealing with this loss of information by biasing the artificial process noise added to each particle in the direction of the mean, such that the covariance doesn't change during the resampling process. While this sounds like a good approach for a unimodal distribution, no such guarantees can be made for the problem at hand.

Instead, the approach proposed in this paper is to recognize that the distribution starts to look fairly gaussian after it converges to a unimodal distribution and becomes fairly compact. At this point the distribution is modelled as a gaussian, using the covariance matrix calculated from the weighted particle distribution, and further updates are performed using an extended Kalman filter. With no further loss in information from resampling the sensor's pose can be estimated with excellent precision.

The lower bound on the uncertainty with which a sensor's pose can be estimated is achievable in the scenario where linear observations are made of a single robot whose position is known exactly. In this case the state of the system at time k is a column vector $x(k)$ describing the pose of the sensor.

Since the system is linear and perfect knowledge of the robot's position is assumed, the sensor can subtract the robot's position to take noisy observations of its own pose directly. This gives rise to the observation equation

$$z(k) = \mathbf{H}x(k) + w(k) \quad (1)$$

where w_k is an uncorrelated vector of white observation errors with mean zero and variance \mathbf{R} .

The lower bound on the uncertainty in the sensor's pose can be found by using the information form of the Kalman filter to examine the evolution of the state covariance matrix. The update equation may be written as

$$\mathbf{P}(k | k)^{-1} = \mathbf{P}(k | k - 1)^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \quad (2)$$

Under the assumption of perfect knowledge of the robot's location and a static sensor,

$$\mathbf{P}(k | k - 1)^{-1} = \mathbf{P}(k - 1 | k - 1)^{-1} \quad (3)$$

Applying Equations 2 and 3 successively for k observations, assuming no initial information about the sensor's pose, results in

$$\mathbf{P}(k | k)^{-1} = k \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \quad (4)$$

Inverting 4 gives the covariance matrix

$$\mathbf{P}(k | k) = \frac{\mathbf{H}^{-1} \mathbf{R} \mathbf{H}^{-T}}{k} \quad (5)$$

This gives a lower bound on the sensor's uncertainty

$$\lim_{k \rightarrow \infty} \mathbf{P} = 0$$

This proof only serves to show the best possible precision that can be obtained. In the more general case where the robot is not perfectly localized there will be a non-zero lower bound on the uncertainty in the sensor's position.

III. IMPLEMENTATION

The system described above has been implemented in our laboratory for stationary cameras, attached to the rafters, observing mobile robots. The robots are Pioneers with SICK lasers, which are well-localized based on a set of laser-reflective beacons. A variant of SLAM [14] is used to generate a map of the laser-reflective features and, once the map has converged, the robots localize within this map using a beacon-based localizer. Sensors learn of robots' estimated positions by listening to network traffic that is communicated on the Active Sensor Network described in [15].

Camera locations are initialised with uniform distributions over x , y , z and yaw, while pitch and roll can easily be measured with a cheap tilt sensor. Figure 3 shows an initial distribution of 25,000 particles.

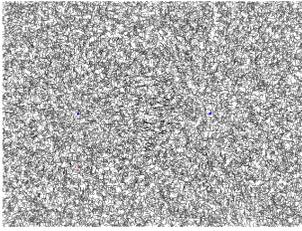


Fig. 3. View from above of the initial distribution of 25,000 particles. The particles are scattered randomly in x , y , z and yaw. Each particle represents a possible sensor pose.

Observations of robots are made using a background subtraction algorithm which identifies moving objects. This algorithm is only capable of determining the direction from the camera to the robot; no information about the robot's heading is extracted. From each image in which motion is detected an azimuth and elevation to the moving object are obtained. The likelihood function used to re-weight particles is a bivariate gaussian over azimuth and elevation plus a small constant offset:

$$L = e^{(-\frac{1}{2}\sigma_{az}^2 - \frac{1}{2}\sigma_{el}^2)} + 0.1$$

where σ_{az} and σ_{el} are the number of standard deviations by which the observation is in error, in azimuth and elevation respectively. The offset is added so that the relative weightings over extremely unlikely particles are approximately equal, to help ignore false observations.

Re-sampling is performed only when the number of effective particles, N_{eff} , drops below 0.5, where

$$N_{eff} = \frac{1}{\sum_i (w_k^i)^2}$$

and w_k^i is the weighting of particle i at time-step k . Infrequent sampling is desirable both in terms of increased computational efficiency and reduced sample impoverishment.

As discussed earlier, kernel-smoothing is required before re-sampling to avoid extreme sample impoverishment in the absence of process noise. As suggested in [16] a product kernel is used. This involves smoothing the data once per dimension, using the same univariate kernel but with a different bandwidth in each dimension. A gaussian kernel is used with a bandwidth h_i dependant on the estimated variance of the data in each dimension, as

$$h_i = \sigma_i n^{-1/5}$$

where σ_i is the standard deviation in the i 'th dimension as estimated from the data and n is the number of particles. [12].

When the particle distribution collapses to the point that the variances in x , y and z are below a threshold the full covariance matrix is calculated, doubled to account for errors in approximation by a gaussian, and used to seed a Kalman filter-based estimate of the camera location. Frames from a typical run, showing some instantaneous observations and the associated states of the estimator, are shown in Figure 4.

IV. EXPERIMENTAL RESULTS

To assess the accuracy of this technique the location of the camera was measured within the map that had been built by the mobile robot.

In addition, the predictive power of the filter was tested. Given an estimate of the robot's location (received over the network) and an estimate of the camera's location (the best estimate from the filter), an observation of the robot in image-space can be predicted at each time step. Comparing these predictions with the real observations produces an innovation sequence. A similar innovation sequence can be generated, using the measured camera location rather than the location estimated by the filter.

An analysis was made of a typical run including some false observations from humans. The raw innovations for the azimuth and elevation predictions are shown in Figure 5, using both the measured and estimated sensor location. It can be seen that the estimated sensor location is inaccurate initially, but converges to the right area after about the first thirty frames. Figure 6 shows the same graphs from the 500th observation onwards, so that finer detail can be seen. It can be seen from these graphs that there is a small time-correlation in the innovation sequence. This is most likely due to inaccuracies in picking the robot's location from its bounding box, which is somewhat dependant on the robot's elevation in the image frame. Any time lag in the estimates of the robot's location will produce correlations in the innovations also. Fixing this should improve accuracy.

After the switch to a Kalman filter the normalized innovations can be plotted. These are shown in Figure 7, showing that the filter is performing correctly. Very large innovations are produced by humans in the field of view at various times in the sequence. The large innovations allow these false observations to be ignored.

A total of four trials were performed, producing four estimates of the sensor's location in (x, y, z) space, from which a mean location was calculated. The standard deviation of the Euclidean distance of the estimates from the mean was 0.7cm, showing that the estimates consistently converged to a common location. The distance between the mean estimated location and the measured location was 14.5cm. This is an encouraging result, considering that the camera itself is about 11cm long.

Another possible reason for this consistent difference between the measured and estimated locations is the inaccuracy inherent in representing a physical robot as a point target. Some point on the image of the robot must be chosen as its centre, which may consistently be different from the point which the robot reports as its centre. Any consistent observation bias will be accounted for by the localizer with a position offset. This feature is desirable if the sensor is to be used to take observations of similar targets, however it is undesirable if the sensor's position estimate is to be used directly, for example by a repair robot that has to find the sensor.

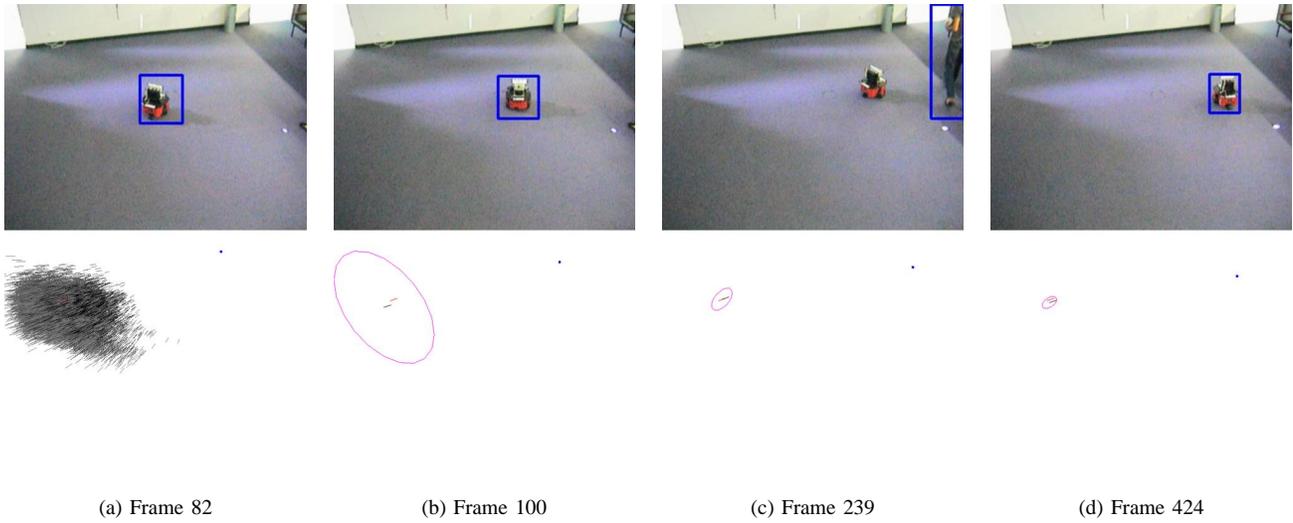
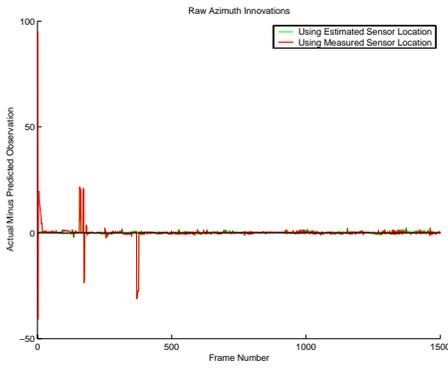
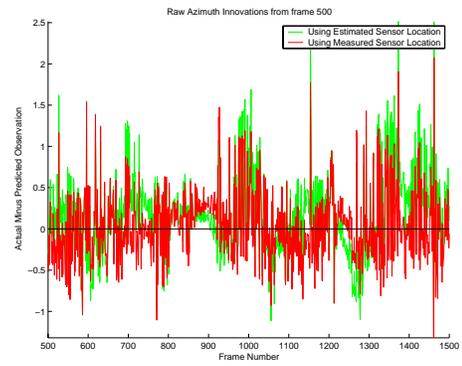


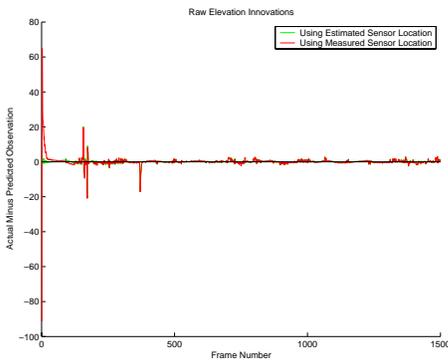
Fig. 4. Frames from a typical run, showing the current observation and the state as (a) the particle filter starts to converge, (b) a Kalman filter is adopted, (c) false observations are ignored and (d) the sensor becomes fairly well localized. After adoption of a Kalman filter, three-sigma uncertainty ellipses are shown in magenta. The measured sensor location is shown in red.



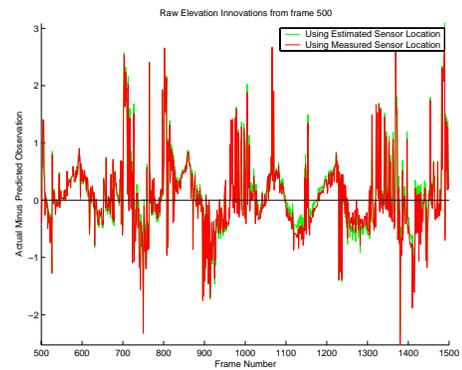
(a) Azimuth Innovations



(a) Azimuth Innovations



(b) Elevation Innovations



(b) Elevation Innovations

Fig. 5. Raw innovations for both (a) azimuth and (b) elevation, showing the difference in degrees between the expected and actual observations, for both the measured and estimated sensor locations.

Fig. 6. Raw innovations for both (a) azimuth and (b) elevation. This is the same data as shown in Figure 5, but only data after the 500th frame is shown.

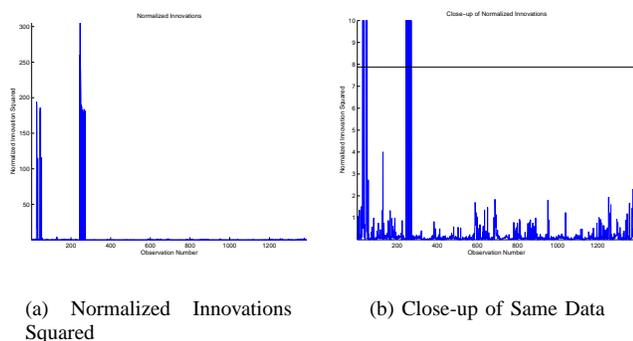


Fig. 7. Normalised innovations squared, after the model has switched to a Kalman filter. The black line is the confidence interval used to gate observations. This corresponds to the conservative 99.5% confidence interval. Humans were present in the field of view in frames 156-161, 170-175 and 369-385. The false observations from these humans produced the very large innovations around these times. The expected value of the normalised innovation is equal to the number of degrees of freedom in the observation, 2 in our case. This shows that the expected observation noise is a little conservative.

V. DISCUSSION

There are several schemes described in the literature for the localization of sensing nodes [17] [18]. The system discussed in this paper has several desirable qualities:

1) Rather than localizing from a specific signal sensed in a specific modality, networks of heterogeneous sensors can be localized simultaneously. Consider a network consisting of seismic, magnetic, vision-based and acoustic sensors, which is designed to track large vehicles. The sensors could be localized by driving one or more well-localized vehicles through the area and having each observe the vehicle in its preferred modality.

2) Rather than localizing a radio receiver which is attached to a camera, this system localizes the actual sensor which is intended for the sensing activity. This should provide better accuracy.

3) Systems that rely on measuring the Time Difference Of Arrival at various sensors require that the sensor density be such that a localizing signal be detected by multiple sensors, and require that the TDOAs from several sensors be transmitted to a central location for processing. In the system presented in this paper each sensor localizes independently.

4) While the robots that are being observed must transmit position estimates, the nodes that are to be localized are entirely passive. There is no need for them to transmit any special codes or localizing signals.

VI. CONCLUSION AND FUTURE WORK

This paper has described a system in which elements of a network are localized based on observations of other elements of the network. The implementation of this system has been described and analyzed, validating its performance.

We are working on integrating this localization scheme with an Active Sensor Network we are developing which will operate continuously, maintaining and updating distributed models of the environment such as occupancy grids and the

locations and activities of people. We are also investigating control strategies for active nodes. When a new node is added in an unknown location, active nodes should consider the benefit of acting in a way that is more likely to help the new node become localized, in comparison to the benefits of fulfilling its other tasks, and behave accordingly.

VII. ACKNOWLEDGEMENTS

This work is supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

REFERENCES

- [1] Estrin, Girod, Pottie, and Srivastava, "Instrumenting the world with wireless sensor networks," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001), Salt Lake City, Utah, May 2001*, 2001, pp. 2033–2036.
- [2] H. Qi, S. S. Iyengar, and K. Chakrabarty, "Distributed sensor networks - a review of recent research," *Journal of the Franklin Institute*, pp. 655–668, 2001.
- [3] H. Durrant-Whyte and M. Stevens, "Data fusion in decentralized sensing networks," in *4th International Conference on Information Fusion*, 2001, pp. 302–307.
- [4] S. Williams, G. Dissanayake, and H. Durrant-Whyte, "Field deployment of the simultaneous localisation and mapping algorithm," in *15th IFAC World Congress on Automatic Control*, 2002.
- [5] Dellaert, Fox, Burgard, and Thrun, "Monte carlo localization for mobile robots," in *International Conference on Robotics and Automation*, vol. 2, 1999, pp. 1322–8.
- [6] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.
- [7] A. Howard, S. Siddiqi, and G. Sukhatme, "An experimental study of localization using wireless ethernet," in *International Conference on Field and Service Robotics*, 2003.
- [8] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan, "An introduction to mcmc for machine learning," *Machine Learning*, pp. 5–43, 2003.
- [9] S. Thrun, "Particle filters in robotics," in *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, 2002.
- [10] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [11] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*. New York, J. F. G. D. F. A. Doucet and N. J. Gordon, Eds. Springer-Verlag, New York, 2000, pp. 197–223.
- [12] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.
- [13] M. Dissanayake, P. Newman, H. Durrant-Whyte, S. Clark, and M. Csorba, "An experimental and theoretical investigation into simultaneous localisation and map building," in *6th International Symposium on Experimental Robotics*, 1999, pp. 171 – 180.
- [14] S. Williams, G. Dissanayake, and H. Durrant-Whyte, "An efficient approach to the simultaneous localisation and mapping problem," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002, pp. 406–411.
- [15] A. Makarenko, E. Nettleton, B. Grocholsky, S. Sukkarieh, and H. Durrant-Whyte, "Building a decentralized active sensor network," in *The 11th International Conference on Advanced Robotics*, 2003.
- [16] D. Scott, *Multivariate Density Estimation: Theory, Practice and Visualization*. John Wiley and Sons, Inc., 1992.
- [17] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low cost outdoor localization for very small devices," University of Southern California, Tech. Rep. 5, 2000.
- [18] R. Moses, D. Krishnamurthy, and R. Patterson, "A self-localization method for wireless sensor networks," *EURASIP Journal on Applied Signal Processing*, 2002.