Rapid Mesh Network Setup for Indoor RF Tracking

Mark Hedley
CSIRO ICT Centre
Sydney, Australia
Mark.Hedley@csiro.au

Thuraiappah Sathyan
CSIRO ICT Centre
Sydney, Australia
Saji.Sathyan@csiro.au

***Abstract*— In this paper we present a practical system for indoor tracking that is low-cost, quick and simple to setup, and accurate. The system consists of three types of nodes, GPS referenced anchor nodes, static nodes placed at fixed locations within a building, and mobile tags that are attached to objects to be tracked. The network of anchor and static nodes must be setup to be globally rigid which allows a unique determination of the location of the static nodes using cooperative localization. The static and anchor nodes are then used as the reference for tracking the mobile tags. We describe a technique to identify static nodes that cannot be uniquely solved so that they can be augmented or removed, and then present a fast and distributed algorithm for determining the locations of the rigid static nodes. We also present an algorithm based on particle filters for tracking the tags that utilizes the knowledge of the range error distribution. Finally we present results obtained using our WASP (Wireless Ad-hoc System for Positioning) platform in an indoor office environment.**

Keywords – RF Tracking, Mesh Network, Particle Filter, Cooperative Localization, Rigid Graph

I. Introduction

Technology for outdoor navigation and tracking is now ubiquitously available, and this has lead to an explosive growth in recent years in location based services, particularly via smart phones. While some of the technologies have been for decidedly trivial consumer applications, the same technology is being harnessed at low cost for applications such as enhancing security by tracking staff and visitors in secure areas. The two key underlying technologies are satellite based navigation (particularly GPS) and fingerprint analysis of the received signal strength (RSS) of radio signals from 802.11 access points. These are naturally complementary, with RSS becoming more accurate in dense urban environments where GPS becomes less accurate. Unfortunately neither of these technologies performs well within buildings. No matter how sensitive a satellite receiver is the coverage within buildings will be highly limited and the accuracy is badly degraded. RSS-based techniques work reasonably well outdoors due to the detailed signal strength maps measured by "war driving", however there is no equivalent information available within buildings. Further the rapid spatial variation of RSS within buildings, due to the building and contents, requires high density sampling to be useful. In addition higher accuracy is expected indoors – it is possible to navigate a road or find a shop with errors of several metres, indoors similar errors make navigating to a particular room difficult.

In this paper we present a practical system for accurate indoor tracking that features rapid deployment. While the most common indoor systems use RSS fingerprint techniques, accuracy is typically no better than several metres, and the requirement for signal strength maps precludes rapid or simple setup. A better alternative is directly measuring range based on measurement of the time of arrival (TOA). We have previously presented such a system called WASP (Wireless Ad-hoc System for Positioning) [1] for accurate indoor and outdoor range measurement, which is the basis for the system described in this paper.

The deployment of a local positioning system (LPS) requires the installation of an infrastructure of reference nodes (often called anchor nodes or base stations). The complexity of installing the infrastructure, including determining the location of each anchor node, has often been seen as an impediment to the use of a LPS, particularly for temporary deployments for security activities. A key contribution of this paper is a new method and algorithms for the deployment of the infrastructure that is fast, simple, and is suitable for staff without high skill levels in network deployment. Rapid deployment is achieved by eliminating the requirement to have signal strength surveys, or to survey the location of the reference nodes within the building.

The system is designed for tracking within large buildings (such that direct tracking from outside the building is not possible), and will naturally also permit tracking in large outdoor areas. It consists of three types of nodes:

- *Anchor Node*: These nodes determine their own location using GPS and are placed outdoors.
- *Static Node*: These nodes are placed in fixed (initially unknown) locations, typically within buildings or in GPS denied areas. They will be subsequently located and form the reference for mobile tags.
- *Mobile Tag*: These are attached to the objects being tracked. They dynamically determine their location using the fixed infrastructure of anchor and static nodes.

System deployment consists of installing anchor and static nodes, determining the location of the static nodes, and then using this fixed infrastructure for tracking mobile tags. Section II describes deployment of the fixed infrastructure, Section III presents an algorithm for locating the static nodes, Section IV presents an algorithm for tracking the mobile tags, and Section V presents experimental results obtained using WASP.

978-1-4673-0387-3/12/$31.00 ©2012 IEEE

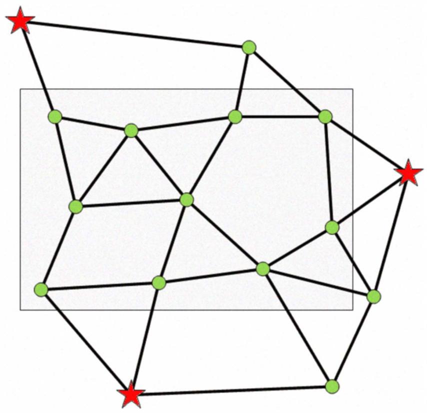

Figure 1. Deployed network of fixed nodes, with grey rectangle representing a building. Red stars are anchor nodes (GPS reference), greed dots are static nodes, and lines are communication links over which range is measured.

II. INFRASTRUCTURE DEPLOYMENT

In this section we will explain how the fixed infrastructure of anchor and static nodes is deployed. An example of a deployment is illustrated in Figure 1. In particular we will describe how to evaluate if the configuration of nodes permits a unique determination of the static node locations. This can occur in real time as the network is deployed, and be used to guide deployment. Nodes that cannot be uniquely located can be determined solely using the network connectivity, and such nodes can either be culled from the set of static nodes, or further static nodes can be installed so that unique locations can be determined.

In this paper we will assume that two-dimensional tracking is being performed, as we have found that in practice it is often not convenient to deploy the infrastructure so as to obtain good three-dimensional tracking using radio frequency (RF) ranging alone. Nonetheless the method described in this paper can be readily extended to three dimensional tracking.

Anchor nodes rely on the availability of a good GPS fix to determine their location. This generally restricts their installation to outdoor areas. For two-dimensional tracking a minimum of three anchors nodes that are not collinear are required. There is no maximum limit to the number of anchor nodes that can be deployed. The accuracy of static node location obtained in a network can be analyzed through the Cramer-Rao lower bound [2], and this shows that we generally minimize errors by having the anchor nodes uniformly distributed around the periphery of the building or area to cover. This is illustrated in the Figure 1 and provides a simple guideline for installation by unskilled personnel.

The static nodes need to be deployed with sufficient density in the area of interest for tracking such that radio communication is available to at least three static nodes. A simple proxy for this condition during deployment is to ensure that each static node can communicate with at least three other fixed nodes (static or anchor). Where this does not occur additional static nodes can be installed to fill in the gap. This task is no more difficult than the installation of a wireless network and does not require highly specialized personnel.

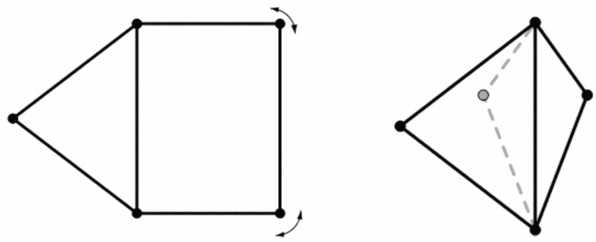

Figure 2. Illustration of (left) finite flexing and (right) partial reflection.

The static nodes must be deployed so that their location can be uniquely determined. The remainder of this section describes how this can be evaluated, and nodes that cannot be uniquely determined can be identified and either removed or augmented with additional static nodes. Uniqueness can be determined using only the connectivity between nodes (i.e., nodes between which range can be measured) and does not depend upon the measured range values. The underlying theory is based on graphs and was presented in [3].

Assume that we have n fixed nodes in total, of which m are anchor nodes. The location of these nodes is denoted by $\mathbf{p}_i$, where $i = 1, \cdots, m$ corresponds to anchor nodes and $i = m+1, \cdots, n$ corresponds to the initially unknown locations of the static nodes. We define a grounded graph $\mathbf{G}_N$ which has n vertices v_i corresponding to $\mathbf{p}_i$, and d edges e_i between the vertices corresponding to nodes between which range can be measured. This graph has an edge between each pair of anchor nodes as the range is known by virtue of knowing the node locations. The locations of the static nodes can be uniquely determined if $\mathbf{G}_N$ is globally rigid. There are three conditions that need to be checked, and Figure 2 illustrates two of these. The graph on the left has a continuous deformation (hence is not a *rigid framework*), and the graph on the right is not unique as distances are presented for a discrete reflection (shown as grey dotted line). There is a third type of rigidity called *redundant rigidity*, and this requires that the graph remain rigid under the removal of any single edge. As the first condition is a trial consequence of the third condition there are two tests to perform. The graph illustrated in Figure 1, when augmented with edges between anchor nodes to form a grounded graph, satisfies all three constraints.

Computationally efficient algorithms for evaluating the constraints have been previously described. Hendrickson [4] presented an $O(n^2)$ algorithm for validating redundant rigidity. During this process *Laman subgraphs* are identified in $\mathbf{G}_N$, where such subgraphs are *redundantly rigid*. Nodes not in any Laman subgraph are thus readily identified, and it is these nodes that cannot be uniquely positioned.

A two dimensional graph has no partial reflections if it is 3-connected [4]. An $O(m)$ algorithm for dividing a graph into its triconnected components was described in [5]. If more than one component is identified additional nodes are required to form connections between the components.

If $\mathbf{G}_N$ is redundantly rigid and triconnected, then the locations of the static node can be uniquely determined

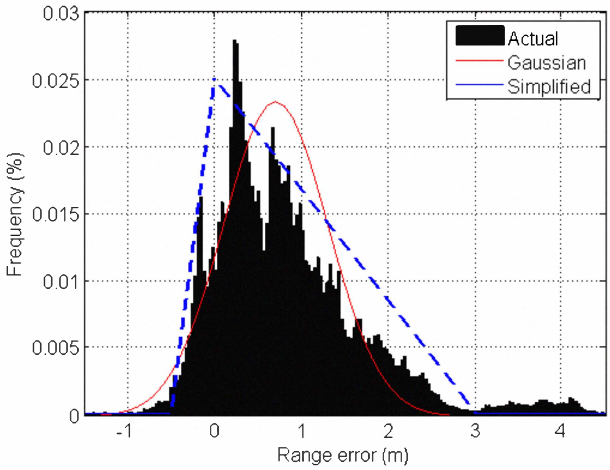

Figure 3. Range error distribution for indoor office environment.

III. STATIC NODE LOCALIZATION

In the previous section we setup the fixed infrastructure to ensure that a unique location can be found for each static node. In this section we present the algorithm to determine the locations of the static nodes $\mathbf{p}_i$, for $i = m+1, \cdots, n$, using the set of range measurements and the known locations of the anchor nodes. This is known as cooperative localization [6], and can be viewed as finding the embedding of the graph $\mathbf{G}_N$ in two-dimensional space that is optimal in some sense.

Previous experimental work has demonstrated [7] that indoor range measurements have an asymmetric distribution (see Figure 3) that is biased to overestimation of range, and that assuming a Gaussian distribution for the range error results is poor localization performance. A common metric to minimize for localization is the sum of the squares of the residuals between measured and calculated ranges. This implicitly assumes a Gaussian distribution. Instead we use the simplified cost metric described in [7] and illustrated as the dashed line in Figure 3 as a likelihood function.

There have been a number of algorithms published for cooperative localization in recent years, such as [8] and references contained therein. The problem is that most are not able to use a non-Gaussian range distribution, or are too computationally complex for real-time operation. The approach that we will use for locating the static nodes is to use the DV distance algorithm [9] to obtain an initial estimate of the node locations. This algorithm is simple and fast yet is reasonably robust. We then refine the initial locations using a particle filter based algorithm that we previously described in [7]. In the remainder of this section we provide a brief review of the algorithms.

A. DV-Distance Algorithm

The DV-distance algorithm [9] propagates positions of the anchor nodes through network flooding, which also allows the static nodes to find the multihop distances to the anchors. Each anchor calculates a correction factor based on the multihop and true distances to the other anchors, and floods the network again with the calculated correction factor. A static node uses the correction factor it retained and the multihop distance to the anchors to estimate the range to the anchors. When a static node has range estimates to three or more anchors, it calculates its position using least squares (LS) estimation. This gives an initial estimate for the location of the static node, and the initial estimate for the variance is the residual in the LS calculation.

B. Particle Filter Refinement Algorithm

The refinement algorithm iteratively updates the location of all static nodes until the locations converge. At each iteration the location of each static node is independently updated using the mean and covariance of the neighbors' state at the previous iteration in addition to the measured range values. The particle filter is used to implement a Bayesian estimator, and the location of the node is the mean of the estimated *a posteriori* distribution (also called a minimum mean squared error (MMSE) estimator). This typically only takes a few iterations to converge.

As the nodes are static the state vector, denoted by $\mathbf{x}_i$ for node i, is just the node location. The estimate of the state for the lth particle at iteration k is denoted by $\mathbf{x}_i^l(k)$. Samples at iteration k are drawn from an importance density for which the most common choice is the transition prior $p(\mathbf{x}(k)\,|\,\mathbf{x}(k-1))$, which is intuitive and simple to implement [10]. The weight update is dependent only on the likelihood, $p(\mathbf{z}_i\,|\,\mathbf{x}_i)$, where $\mathbf{z}_i$ is the set of measured ranges to node i. The updated weight of the lth particle is given by [10]

$$w_i^l(k+1) \propto w_i^l(k)\, p(\mathbf{z}_i(k+1)\,|\,\mathbf{x}_i^l(k+1)).$$

After a few iterations the variance of the particles will increase such that there are only a few particles with significant weight. This phenomenon is referred to as sample degeneracy and can lead to divergence of the particle filter [10]. The objective of the resampling step is to remove the particles with insignificant weights and to generate a new particle set by concentrating on the ones with significant weights [11]. At each time step new samples are drawn from the newly proposed particles using the updated weights as probabilities. Since the newly generated particles are independent and identically distributed samples, they are assigned equal weights. A number of resampling techniques have been proposed in the literature. In this paper we use systematic sampling [11].

Not only is there an error in the range measurement, captured by the approximate likelihood function, but there is also an error in the location of neighboring nodes. For anchor nodes the GPS receiver returns an error value, and for static nodes a covariance is updated at each iteration. This uncertainty in the location of neighboring nodes is taken into account as now described. Let u be the unit vector from node i to node j, and $\mathbf{P}_j^k$ be the covariance of the location at iteration k of node j. The variance in the range is given by $\mathbf{u}^T\mathbf{P}_j^k\mathbf{u}$, and the simplified range error model shown in Figure 3 is extended by the square root of this value.

IV. Tracking Mobile Tags

Once deployment is complete and the fixed nodes (anchor and static) have known locations the network is ready to use. The final algorithm described in this section is for tracking the mobile tags. Each tag is independently tracked using only range measurements to the fixed nodes. A Bayesian tracking algorithm is used, and this uses the range error model described in the previous section and in Figure 3. The filter takes as input the measured range values between a mobile tag and fixed nodes. The advantage of this over separately computing a location at each time step and applying a tracking filter to the computed locations is that tracking continues with fewer range measurements. Further, computationally simple gradient descent type algorithms can end up in a local minima and result in high location error due to the non-Gaussian nature of the range measurement noise. As the measurement equation (that relates range to location) and the noise is non-Gaussian, a particle filter is selected for tracking. The algorithm is based on that which was previously presented in [13].

A. Modeling

Let $\mathbf{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]$ denote the state vector consisting of the position and velocity of a mobile tag. The motion dynamics of the tag are usually unpredictable, particularly when carried by a person, hence a multiple model approach is used. In our implementation we assume that the state transition is adequately represented by two nearly constant velocity (NCV) models with different process noise values. The state transition model is then expressed as

$$\mathbf{x}_k = F_k \mathbf{x}_{k-1} + \mathbf{v}_k^{r_k}.$$

F_k is the state transition matrix. For a time between successive samples of T and denoting the two-dimensional identity matrix by I_2 we have

$$F_k = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \otimes I_2$$

$\mathbf{v}_k$ is the model dependent process noise, where r_k is in set {1,2}. It is assumed to be a zero mean white Gaussian sequence with covariance matrix given by

$$Q_k^{r_k} = q_k^{r_k} \begin{pmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{pmatrix} \otimes I_2$$

where q is a model dependent power spectral density of the process noise.

The measured range between the mobile node and anchor node j at time k can be written in terms of the $\mathbf{p}_k$ (the location of the tag at time k), $\mathbf{p}^j$ (the known location of anchor node j) and the measurement noise ω which is modeled as illustrated in Figure 3.

$$z_k^j = \left\| \mathbf{p}_k - \mathbf{p}^j \right\|_2 + \omega_k^j.$$

The augmented state of the target at time t_k is denoted by

$$\mathbf{y}_k = [\mathbf{x}_k \quad r_k].$$

The transition between the models is assumed to obey a first-order Markov chain. The probability matrix governing the transition defined by $[p_{ij}]$ where

$$p_{ij} = P\{r_k = i \mid r_{k-1} = j\}.$$

B. Filter

The optimal Bayesian solution for tracking the state at each time step k given the measured data up to that time is given by the following recursion [14]

$$p(\mathbf{y}_k \mid \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k \mid \mathbf{y}_k)}{p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})} \int p(\mathbf{y}_k \mid \mathbf{y}_{k-1}) p(\mathbf{y}_{k-1} \mid \mathbf{z}_{1:k-1}) d\mathbf{y}_{k-1}.$$

The optimal Bayesian recursion is only conceptual and cannot be determined analytically expect in special cases. The particle filter uses a point mass representation to approximate the optimal Bayesian recursion. This representation consists of a set of random samples and associated weights. Denote by $\{\mathbf{y}_{k-1}^i, w_{k-1}^i\}_{i=1}^N$ the random sample-weight pairs that represent the posterior density at time k-1. With the availability of the latest measurement at time k the particle filter algorithm approximates the posterior density at time k by a new set of random sample-weight pairs $\{\mathbf{y}_k^i, w_k^i\}_{i=1}^N$, through sampling and resampling steps, as described in the previous section.

The estimated location and velocity of each tag at each time step is given by the weighted mean of the particles (i.e. MMSE estimator)

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{N} w_k^i \mathbf{x}_k^i$$

For indoor tracking we use the standard multiple model particle filter (MM-PF) [15]. In this approach the state is augmented with a discrete variable representing the active model and model switching is assumed to be governed by a first-order Markov chain. During sampling the discrete model variable is drawn first and, conditional upon the model, the kinematic state of the model is drawn. Weights are calculated using the simplified range error model shown in Figure 3 and the systematic resampling algorithm is used for resampling.

For outdoor tracking, where the range measurement noise is well approximated by a Gaussian distribution, we can greatly simplify the computational complexity of the standard MM-PF using a Rao-Blackwellization. We presented such an algorithm in [13], and demonstrated its performance and its ability to reduce the computational complexity with WASP data measured in an outdoor environment.

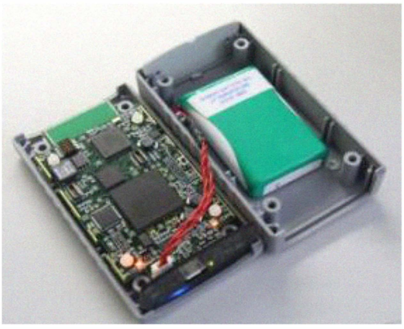

Figure 4. WASP mobile tag.

V. EXPERIMENTAL RESULTS

A. Experimental Apparatus

The WASP platform [1] was developed for accurate localization and high-rate data communications using low-cost hardware in difficult radio propagation environments (e.g. strong multipath interference). The system was also designed to be suitable for rapid deployment in unknown environments, for a range of applications including tracking athletes and fire-fighters. A picture of a WASP node is shown in Figure 4.

In the presence of significant multipath interference the TOA accuracy is limited by the bandwidth of the signal. In order to maximize accuracy WASP signaling uses the entire 125 MHz bandwidth available in the 5.8 GHz frequency band allocated for industrial, scientific and medical (ISM) purposes. Although the width of the impulse response (null to null) is 16 ns, the super-resolution TOA algorithm that we have developed [12] is capable of accuracy down to just half a nanosecond. We implemented a simple custom TDMA (time division multiple access) protocol to suit the limited computational performance of the hardware. The physical layer uses orthogonal frequency division multiplexing (OFDM), with data rates of 4 Mbps and 8 Mbps The time slots in the TDMA MAC protocol have 2.5 ms duration, and a typical sports system has ten anchor nodes and up to thirty mobile nodes, taking a total of forty slots. The update rate is ten locations per second.

B. Experimental Data

Figure 5 shows the network layout used in one of the experiments conducted to evaluate the performance of the proposed algorithm. There were five anchor nodes and 28 static nodes. There were another seven mobile tags, which were carried by six people. The update rate of WASP during the experiment was 5 Hz for each node and the experiment lasted nearly 200 s.

The true locations of all static nodes were determined through a survey and hence, the ranges between them were calculated and compared to the measured ranges. Figure 3 shows the actual range error histogram along with a best fitting Gaussian distribution and a simplified triangular likelihood function. The best fitting Gaussian has a standard deviation of 0.6 m. To present the shape of the high probability region clearly, the tail of the distribution (probability mass of 2%) is not shown in the figure.

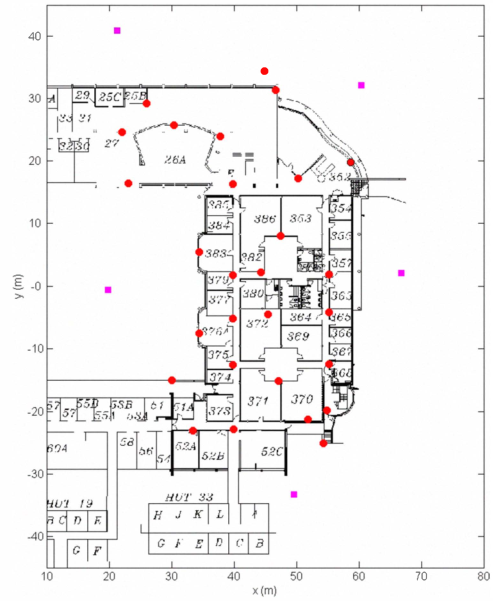

Figure 5. Node locations for experimental data in large office building. Squares are anchor nodes and circles are static nodes.

C. Uniqueness of Solution

The data consisted of range measurements between nodes at a number of time steps. A single measured distance matrix was formed which was the median range measured between each pair of nodes, or zero if there were no measurements between a given node pair. A connection matrix was also formed which contained unity if a median range measurement was available, and zero otherwise. Programs were written in Matlab to implement the algorithms described in Section II for evaluating triconnectivity (no partial reflections) and rigidity. The executing time was only 0.11 seconds for the 33-node grounded graph, with the result that the graph is redundantly rigid and triconnected, hence is expected to have a unique solution.

D. Location of Static Nodes

The algorithm described in Section III to determine the location of the static nodes was implemented in Matlab. The DV distance algorithm was used to find an initial location using the median range matrix, followed by five iterations of the particle filter refinement algorithm. This number of iterations was found to be sufficient for the algorithm to converge and there was little improvement with further iterations. The algorithm used a single NCV model with power spectral density $0.1\text{m}^2/\text{s}^3$ and 1000 particles. Weight calculations were based on the simplified range error distribution model shown in Figure 3, and the sample covariance of the static nodes were used to adjust this distribution to account for the uncertainty in the neighboring nodes.

The true locations of the static nodes were known as they were surveyed, hence the error in the computed location of the static nodes could be computed. The cumulative distribution function of the error is plotted in Figure 6. From this it is seen that the median positioning error is 0.62 m, and 90% of static nodes have a positioning error of less than 1.25 m.

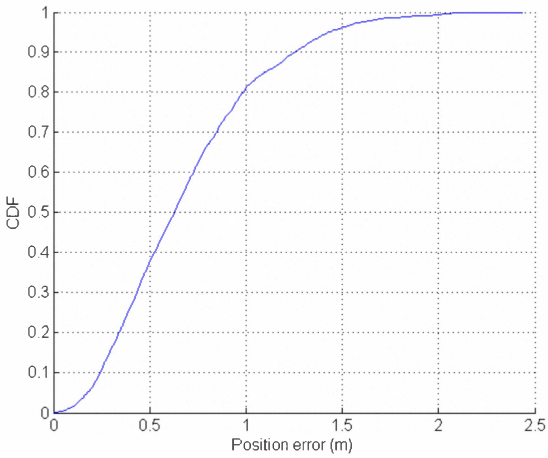

Figure 6. Cumulative density function of position error of static nodes.

E. Tracking of Mobile Tags

As the data was obtained in an indoor environment, the multiple model particle filter described in Section IV was implemented in Matlab for tracking the mobile tags.The MMPF consisted of 2000 particles. Two NCV models were used to characterize the motion of the mobile node. The power spectral density of the two models were set to 0.1 and 0.5 m^2/s^3. The simplified range error model shown in Figure 3 was used to update the weights of the particles. Figure 7 shows the computed paths of the mobile tags. As there was no independent tracking system the positioning system cannot be quantified, but qualitatively the paths are a good representation of the true paths taken.

VI. CONCLUSION

In this paper a system for indoor tracking that is accurate and can be rapidly deployed has been described. Experimental results obtained using the WASP platform in an office environment have been used to verify the algorithms and show that high accuracy can be obtained using only five external GPS referenced anchor nodes.

ACKNOWLEDGMENT

We wish to thank the members of the wireless tracking team at CSIRO for the development of the WASP platform and for experimental work to obtain the data used in this paper. In particular we wish to thank Alex Grancea, Phil Ho, David Humphrey, Mark Johnson, Joseph Pathikulangara and Alija Kajan.

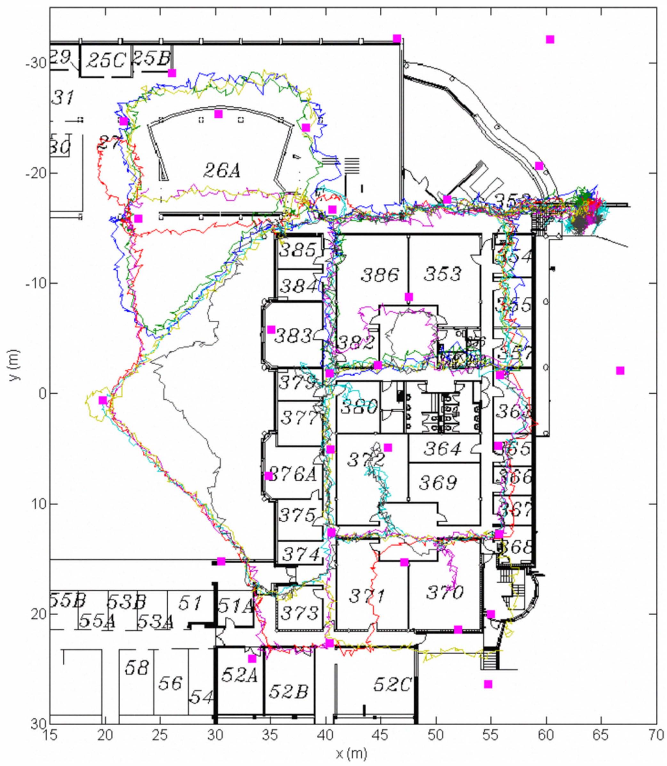

Figure 7. Plot of paths of seven mobile tags. Rectangles are fixed nodes.

REFERENCES

[1] T. Sathyan, D. Humphrey and M. Hedley, "A System and Algorithms for Accurate Radio Localization using Low-cost Hardware," *IEEE Trans. Society, Man and Cybernetics – Part C,* vol. 41, no. 2, pp. 211-222, March 2011.

[2] E.G. Larsson, "Cramer-Rao bound analysis of distributed positioning in sensor networks," *IEEE Signal Processing Letters,* vol. 11, no. 3, pp. 334 – 337, March 2004.

[3] J.A. Aspnes, T. Eren, D.K. Goldenberg, A.S. Morse, W. Whiteley, B.D.O Anderson and N. Belhumerur, "A Theory of Network Localization," *IEEE Trans. Mobile Computing*, vol. 5, no. 12, pp. 1663-1677, Dec. 2006.

[4] B. Hendrickson, "Conditions for Unique Graph Realizations," *SIAM J. Comput.*, 21(1), pp.65-84, Feb. 1992.

[5] J.E. Hopcroft and R.E. Tarjan, "Dividing a graph into triconnected components," *SIAM J. Comput.* 2(3), pp. 135-158, 1973.

[6] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero, R.L. Moses and N.S. Correal, "Locating the Nodes," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54-69, Jul. 2005.

[7] T. Sathyan and M. Hedley, "A particle filtering algorithm for cooperative tracking of nodes in wireless networks," *Proc. PIMRC 2011*, Toronto, Sept. 2011, pp. 1279-1283.

[8] H. Wymeersch, J. Lien and M.Z. Win, "Cooperative Localization in Wireless Networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427-450, Feb. 2009.

[9] D. Niculescu and Badri Nath, "DV Based Positioning in Ad Hoc Networks," *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267-280, 2003.

[10] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filter for online/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[11] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice.* New York, NY: Springer, 2001.

[12] D. Humphrey and M. Hedley, "Super-resolution Time of Arrival for Indoor Localization," *IEEE International Conference on Communications*, pp. 3286–3290, Beijing, China, May 2008.

[13] T. Sathyan and M. Hedley, "Efficient Particle Filtering for Tracking Maneuvering Objects," IEEE/ION PLANS 2010, California, May 2010.

[14] M.S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for online nolinear/non-Gaussian Bayesian tracking," *IEEE Trans. Sig. Proc.*, vol. 50, no. 2, pp. 174-188, Feb. 2002.

[15] S. McGinnity and G. Irwin, "Multiple model bootstrap filter for maneuvering target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 3, pp. 1006–1012, Jul. 2000.

[16] R. van der Merwe, N. de Freitas, A. Doucet, and E. A. Wan, "The unscented particle filter," Cambridge University Engineering Department, Cambridge, UK, Tech. Rep. CUED/FINFENG/TR380, Aug. 2000.