In *International Workshop on Robotic Wireless Sensor Networks (IEEE DCOSS '09)*,
Marina Del Rey, California, USA, June 2009.

Connectivity-based Localization in Robot Networks

Tobias Jung, Mazda Ahmadi, Peter Stone
Department of Computer Sciences
The University of Texas at Austin
Email: {tjung,mazda,pstone}@cs.utexas.edu

***Abstract*—Consider a small team of autononomous robots, each equipped with a radio, that are deployed in an ad-hoc fashion and whose goal it is to act as signal relay nodes to form a temporary, adaptive, and highly robust communication network. To perform this type of self-optimization and self-healing, relative localization (i.e. knowing direction and distance to every other robot in the network) is necessary. In a sense, the problem is similar to the one studied in ad-hoc sensor networks. The key differences are that (1) anchor nodes with known locations are not available; that (2) the connectivity graph is very sparse, because of a comparatively small number of nodes involved; and that (3) the communication nodes are actually mobile robots such that apart from location we also have to estimate the directions to other nodes (which can not be obtained from a single time slice). To solve this problem, we propose a global approach that exploits the mobility of the robots to obtain multiple connectivity measurements over a small time window. Together with the odometry of individual robots, we then try to estimate underlying locations that best explain the observerd connectivity data by minimizing a suitable stress function. Through simulation of a concrete real-world scenario we show that our approach performs reasonably well with as few as ten robots. We examine its performance both under outdoor and indoor conditions (i.e. uniform and non-uniform signal propagation). In addition, we also consider the case where we are able to observe the distance between connected tobots, which further improves accuracy substantially.**

I. Introduction

While radio communication generally works well in open (line of sight — LOS) environments, impeding features such as walls or other obstacles hinder the propagation of radio signals in indoor (non-LOS) environments such that direct communication often becomes impossible. However, having communication when operating in these environments is highly desirable for many real-world situations (for example, establishing and maintaining communication throughout disaster-stricken urban areas to coordinate rescue and emergency operations). A solution we explore in this paper is to employ a team of intelligent communication robots that are scattered through the environmnent and whose task it is to autonomously create and sustain a temporary communication network. To maintain reliable communication over extended periods of time, the robots will have the ability to move so that they can constantly adjust their position and always find the best signal. In particular, as the network is intended to be multi-path and multi-hop, the loss of individual nodes should not stop data from finding its way to its endpoint via alternative node paths. Thus, if a single robot fails or is destroyed, the rest of the network will have to reposition to cover-up the resulting black spot. In this context, having relative localization (i.e. knowing direction and distance to every other robot in the network is necessary, or at least very useful, for performing the desired self-optimization and self-healing behavior [1].

For this paper, we describe a history-based approach that jointly estimates relative positions of the robots by combining individual odometry with the global connectivity graph over a small time window. More specifically, we will consider the localization problem under the following constraints (which stem from our particular application):

- the robots are deployed ad-hoc, scattered throughout the environment and unaware of their initial global location and heading. The goal is to estimate their relative position in the network (i.e. distance and direction to all other robots).
- as there are no stationary beacon nodes with known locations, the relative locations of the robots can only be jointly determined from connectivity.
- the total number of robots is small, such that any technique relying on dense coverage will fail.
- the robots are mobile[1], but we do not assume control over their movement. However, the robots are equipped with fairly accurate odometry sensors.
- the robots may have to operate under non-LOS conditions, such that signal strength measurements may become unreliable and cannot be used to directly infer the underlying distance.
- the robots (small, inexpensive units) lack any sensor device that would otherwise enable sophisticated mapping of the environment.

II. Related work

The localization problem for a single mobile robot has been extensively studied in the past. Mobile robot localization usually works by assuming a prior map and then trying to determine the robot's position with respect to that map by integrating motion and sensor data over time using Bayesian filtering. More advanced algorithms try to learn a map and solve the localization problem simultaneously (SLAM). For a summary of these methods, see [10]. However, chiefly because of two reasons, these methods are not directly applicable to our situation: (1) the robots are deployed in uncharted territory and lack sensors for mapping the environment; and

[1]It is a crucial assumption we make throughout this paper that the robots actually change their positions such that their connectivity changes over time. Only this way will we obtain sufficient data to solve an otherwise seriously underdetermined reconstruction problem.

(2) these methods do not adequately address the problem of simultaneously localizing multiple robots.

Map-free localization is a common objective in the context of WiFi localization. Here, given a number of stationary objects (i.e. WiFi access points), the goal is to estimate the location of a mobile device just based on signal strength readings. In [4] this was done for an indoor environment by first learning signal strength maps for the individual access points from labeled ground truth data using GP regression, and subsequently using Bayesian filtering to estimate the location of the device. In [3] this approach was extended to work without labeled data, using GPLVM, a recent method for non-linear GP-based dimensionality reduction. However, in order to produce reasonable results, dense and overlapping coverage, i.e. a large number of access points, was required. Another example of applying dimensionality reduction for map-free localization is presented in [11]. There, the objective was to determine the location of a number of stationary objects, using first a mobile robot to obtain temporally related (and therefore similar) measurements of some spatial relationship (such as visibility, distance, direction).Then dimensionality reduction was applied to a history of these high-dimensional measurements to produce the low-dimensional locations of the objects.

Localization in ad-hoc sensor networks probably comes closest to what we want to achieve, being both map-free and specifically tailored to simultaneously localize multiple objects from a pairwise similarity measure (radio communication). The methods exploit the fact that each node in a communication network constrains the possible locations of every other node by virtue of having restricted communication range (which usually means every node has contact only with nodes in its neighborhood). A detailed survey of this area is provided by [2]. However, many of the existing technologies rely on large numbers of carefully placed anchor nodes whose location must be known in advance. In addition, many localization methods require more information than just connectivity, and use distance or angle measurements from the anchor nodes to apply multilateration or triangulation techniques to find coordinates of the unknown nodes [6]. A method that does not rely on anchor nodes, and works even when only connectivity is available, is MDS-MAP [8]. However, connectivity-only MDS-MAP requires dense networks with many nodes (on the order of hundreds) and a high degree of connectivity (average number of neighbors). In our case, the communication network consists of a comparatively small number of nodes (in our experiments we consider only 10 nodes) with the average number of neighbors being about 0-2; the reconstruction problem will thus be seriously underconstrained in the absence of movement. Finally, localization in sensor networks usually assumes that the communication nodes are stationary. The novelty/contribution of this paper is to explicitly consider (and exploit) the case where the nodes are in fact autonomous mobile robots and the induced connectivity graph changes over the time (as the robots move around, edges break up and new edges are formed).

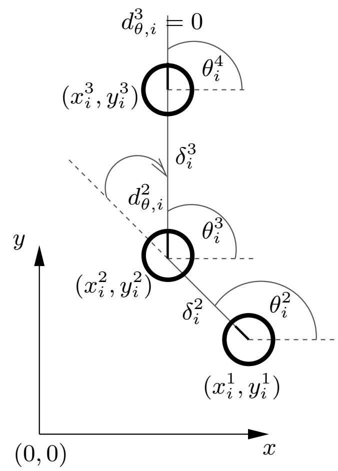


Fig. 1. Motion of robot i over a period of $T = 3$ time steps, shown in a global coordinate system. The heading direction (bearing) is into the direction of the positive x-axis. For every time step $t = 2, \ldots, T$, the robot can observe the relative change in direction $d^t_{\theta,i} = d_\theta(i,t)$, and the amount of distance traveled $\delta^t_i = \delta(i,t)$. In general, the initial pose $(x^1_i, y^1_i, \theta^2_i)$ will be unobservable.

III. Detailed description of our approach

Our approach is a history-based global approach that tries to jointly estimate relative positioning of robots: first, we combine the individual observation and motion histories of all robots in the network. We then estimate locations such that the induced connectivity graph is most consistent with the observed connectivity graph (for all time steps in the history). In the following, we will consider N robots moving in a 2D world for a number of T time steps.

A. *Individual motion*

Let $x(i,t) = x^t_i$ denote the x-coordinate, $y(i,t) = y^t_i$ the y-coordinate, and $\theta(i,t+1) = \theta^{t+1}_i$ the heading of robot i at time t. As illustrated in Figure 1, the coordinates are given with respect to a global coordinate system, and the heading is given with respect to the positive x-axis (note the use of $t+1$ to indicate heading at time t).

Given only the coordinates $(x^1_i, y^1_i), \ldots, (x^T_i, y^T_i)$, we can compute the associated (absolute) heading for all but the last time step. First, introduce the distance robot i moved between two successive time steps along each of the coordinate directions:

$$d_x(i,t) := x(i,t) - x(i,t-1), \quad t = 2 \ldots T$$
$$d_y(i,t) := y(i,t) - y(i,t-1), \quad t = 2 \ldots T.$$

We then have

$$\theta(i,t) = \operatorname{atan2}\big(d_y(i,t), d_x(i,t)\big), \quad t = 2 \ldots T,$$

which gives us the heading (for robot i) at time steps $t = 1 \ldots T-1$ (see Figure 1).

Conversely, if we know the start pose $(x^1_i, y^1_i, \theta^2_i)$, i.e. location and heading at time $t = 1$, and are given both the distances robot i traveled between successive time steps, i.e.

$$\delta(i,t) := \big(d_x(i,t)^2 + d_y(i,t)^2\big)^{1/2}, \quad t = 2 \ldots T, \quad (1)$$

and the relative changes in heading, i.e.

$$d_\theta(i,t) := \theta(i,t) - \theta(i,t-1), \quad t = 2\dots T, \tag{2}$$

we can compute the locations for all time steps $t = 2\dots T$ beyond the first:

$$\theta(i,t) = \theta(i,2) + \sum_{k=3}^{t} d_\theta(i,k) \tag{3}$$

$$x(i,t) = x(i,1) + \sum_{k=2}^{t} \delta(i,k)\cos\bigl(\theta(i,k)\bigr) \tag{4}$$

$$y(i,t) = y(i,1) + \sum_{k=2}^{t} \delta(i,k)\sin\bigl(\theta(i,k)\bigr). \tag{5}$$

Now assume we do not know the initial pose of node, but could observe the history of its motion, i.e. we could observe the δ's from Eq. (1) and d_θ's from Eq. (2). Let $(x_i^{\text{start}}, y_i^{\text{start}}, \varphi_i^{\text{start}})$ be a guess for the true unknown initial pose $(x_i^1, y_i^1, \theta_i^2)$ in a global coordinate frame. Together with the observably odometry data δ and d_θ, every such guess then gives rise to a path $\{(\hat{x}_i^t, \hat{y}_i^t)\}_{i=1,\dots,N}^{t=1,\dots,T}$ via Eqs. (3)-(5):

$$\hat{\theta}(i,t) = \varphi_i^{\text{start}} + \sum_{k=3}^{t} d_\theta(i,k) \tag{6}$$

$$\hat{x}(i,t) = x_i^{\text{start}} + \sum_{k=2}^{t} \delta(i,k)\cos\bigl(\hat{\theta}(i,k)\bigr) \tag{7}$$

$$\hat{y}(i,t) = y_i^{\text{start}} + \sum_{k=2}^{t} \delta(i,k)\sin\bigl(\hat{\theta}(i,k)\bigr). \tag{8}$$

B. Joint connectivity

Whereas motion is handled independently for every individual node, the connectivity graph induced by the locations of all nodes at any time t is a global property of the whole network. Let $\varrho(i,j,t)$ denote the Euclidean distance between node i and node j

$$\varrho(i,j,t) := \bigl[(x_i^t - x_j^t)^2 + (y_i^t - y_j^t)^2\bigr]^{1/2}$$

for $i = 1,\dots,N-1$, $j = i+1,\dots,N$, $t = 1,\dots,T$. Furthermore, let $c(i,j,t)$ denote a binary connectivity variable

$$c(i,j,t) := \begin{cases} 1 & \text{if node } i \text{ and } j \text{ could communicate at time } t \\ 0 & \text{else.} \end{cases}$$

Ideally, under LOS conditions, signal strength falls off uniformly and connectivity is thus directly related with the underlying distance. In this case, we can model connectivity by setting $c(i,j,t) = 1$ if $\varrho(i,j,t) \le R_{\text{max}}$, where R_{max} is the maximum communication distance. Under non-LOS conditions the situation is substantially more difficult; in general, signal strength, and thus connectivity, is then less strongly correlated with distance (e.g. two nodes could be physically close but seperated by a wall that absorbs the signal). In our simulations (see Section 4), we will use a sophisticated ray-casting approach to model physical signal propagation in an indoor environment. However, since our nodes are incapable of determining the topology of their environment (and thus are unaware of walls), we will model the uncertainty using a simple probabilistic relationship between pairwise distance and connectivity.

C. Objective

Regardless of what underlying physical process gave rise to the measured signal strength, assume we could observe the joint connectivity of the network together with the individual motion of nodes over a history of T time steps. Figure 2 depicts this situation. Our goal is now to collectively find initial poses for all nodes, i.e. determine vector

$$\vec{x} := (x_1^{\text{start}}, y_1^{\text{start}}, \varphi_1^{\text{start}}, \dots, x_N^{\text{start}}, y_N^{\text{start}}, \varphi_N^{\text{start}}) \in \mathbb{R}^{3N}, \tag{9}$$

such that when we apply the odometry to expand the paths over time, the connectivity graph induced by the underlying estimated locations best agrees with the observed connectivity graph induced by the unknown true locations. Let $\hat{\varrho}_{\vec{x}}(i,j,t)$ denote the distance between estimated locations for i and j at time t for a particular $\vec{x}$ from Eq. (9):

$$\hat{\varrho}_{\vec{x}}(i,j,t) := \bigl[(\hat{x}_i^t - \hat{x}_j^t)^2 + (\hat{y}_i^t - \hat{y}_j^t)^2\bigr]^{1/2},$$

where $\{(\hat{x}_i^t, \hat{y}_i^t)\}$ is obtained via Eqs. (6)-(8).

In the following we will consider two different scenarios: in the first scenario, we assume, as described above, that we can only observe whether or not two nodes were connected at any given time in the history. In a second scenario, we consider the case where range information is available for nodes that can communicate (for example, by assuming that the nodes are equipped with time difference of arrival (TDoA) hardware).

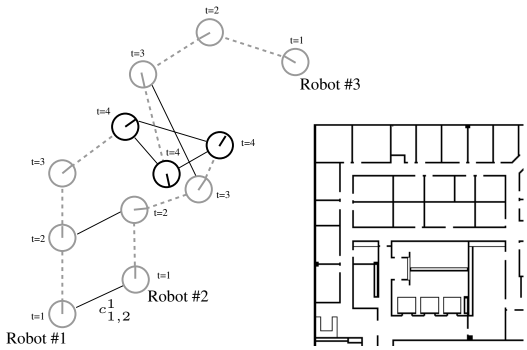
In both scenarios, to determine the best fit between connectivity graphs derived from a particular guess $\vec{x}$ and the observed (true) connectivity, we will first define a suitable error function and then minimize it with respect to the vector of initial poses $\vec{x}$.

Doing this for the second scenario is pretty straightforward: if we have access to the pairwise distances ϱ for connected nodes, we can directly consider the error function

$$E(\vec{x}) = \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\bigl[\hat{\varrho}_{\vec{x}}(i,j,t) - \varrho(i,j,t)\bigr]^2 c(i,j,t), \tag{10}$$

where $\hat{\varrho}_{\vec{x}}$ denotes the pairwise distance for locations expanded from $\vec{x}$, and minimize it with respect to $\vec{x}$. Note that this approach only makes use of information for nodes that are connected, i.e. ignores the nodes that are not connected. Still, as we will see in Section 4, this will give us very good results, since we are integrating information over multiple time steps.

The first scenario is more challenging because we have fewer and less reliable information from which we can infer locations. Here we assume a simple (topology-free) probabilistic model for observing connectivity between nodes at any given time given underlying locations, and find the maximum likelihood solution for $\vec{x}$ over all the full history of



	Pairwise connectivity $c(i,j,t)$:			Pairwise distance $\varrho(i,j,t)$:		
	(1,2)	(1,3)	(2,3)	(1,2)	(1,3)	(2,3)
$t=1$	1	0	0	2.1m	-1	-1
$t=2$	1	0	0	2.2m	-1	-1
$t=3$	0	0	1	-1	-1	3.4m
$t=4$	1	1	1	1.5m	1.2m	1.6m

	Individual odometry $d_\theta(i,t), \delta(i,t)$					
	Robot #1		Robot #2		Robot #3	
	$d_\theta(1,t)$	$\delta(1,t)$	$d_\theta(2,t)$	$\delta(2,t)$	$d_\theta(3,t)$	$\delta(3,t)$
t=2	0deg	0.2m	70deg	0.25m	-40deg	0.3m
t=3	20deg	0.15m	-30deg	0.15m	-80deg	0.2m
t=4	0deg	0.2m	5deg	0.1m	0deg	0.3m

Fig. 2. Graphical illustration of our objective. Left: As the individual robots move and change their location, so does the underlying connectivity graph change over time. Graph sketches the movement of 3 robots over 4 time steps; lines mark connectivity. Center: A sketch of the indoor office environment used in the experiments. Right: Measurements we have from which to reconstruct the positioning of the robots with respect to each other. An entry of '-1' in the pairwise distance matrix denotes missing data (i.e. robots could not communicate).

observations. As observation model we utilise a Gaussian cdf

$$p(c^t_{ij}=1\,|\,\hat{x}^t_i,\hat{y}^t_i,\hat{x}^t_j,\hat{y}^t_j) := 1-\Phi_{\mu,\sigma^2}\big(\hat{\varrho}_{\vec{x}}(i,j,t)\big)$$
$$p(c^t_{ij}=0\,|\,\hat{x}^t_i,\hat{y}^t_i,\hat{x}^t_j,\hat{y}^t_j) := \Phi_{\mu,\sigma^2}\big(\hat{\varrho}_{\vec{x}}(i,j,t)\big)$$

incorporating our belief that the probability of i and j being connected will be high when i and j are close. Here, as usual, $\Phi_{\mu,\sigma^2}(\cdot)$ is defined as

$$\Phi_{\mu,\sigma^2}(\cdot) = \frac{1}{2}\left(1+\operatorname{erf}\left\{\frac{\cdot-\mu}{\sqrt{2\sigma^2}}\right\}\right)$$

with parameters μ being the mean, and σ^2 being the variance of the underlying Gaussian density. We then minimize the error function

$$E(\vec{x}) = \sum_{t=1}^{T}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\Big\{c^t_{ij}\log\Big(1-\Phi_{\mu,\sigma^2}\big(\hat{\varrho}_{\vec{x}}(i,j,t)\big)\Big)+ (1-c^t_{ij})\log\Big(\Phi_{\mu,\sigma^2}\big(\hat{\varrho}_{\vec{x}}(i,j,t)\big)\Big)\Big\} \tag{11}$$

which is the associated negative loglikelihood. The first term in the sum corresponds to the connected nodes, and is large when two nodes were observed as being connected, but their estimated locations are far apart. Conversely, the second term corresponds to the non-connected nodes, and is large when two nodes were observed as being not connected, but their estimated locations are close. Admittedly, this is a rather broad approach that completely ignores the underlying topology, but it is the best we can do given the little information we have.

IV. Experimental results

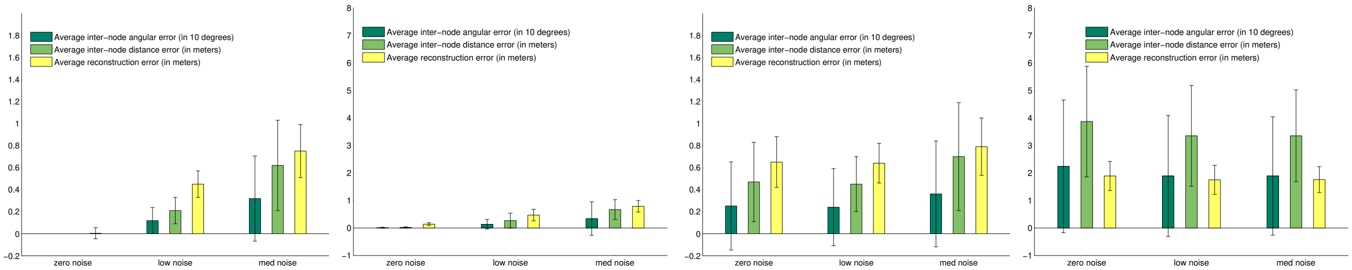
As test environment we simulate an indoor office space, a small sketch of which is shown in Figure 2b. We use CybelePro, a proprietary simulator [X], to model the motion of robots in the environment under real-world conditions. For the following experiments we consider a team of $N = 10$ robots. At the start of the simulation the robots were artificially placed such that they were well spread out and located in different parts of the environment. Under this setup, the nodes did not form a fully connected network initially (some nodes were deliberately placed outside the communication range of all other nodes), but were able to establish a fully connected mesh after performing some exploratory movemnet. This setup simulates the initial phase of our intended application: scattered nodes explore their immediate surroundings and search for other nodes until a fully connected network is established.

We consider two scenarios for the propagation of radio signals. The first one corresponds to an open environment and models received signal strength with radial symmetric attentuation: two nodes were allowed to communicate when their distance was below 9m, irrespective of impeding walls and other obstacles. In the second scenario we actively investigate indoor localization, where the underlying topology (see Figure 2b) strongly effects the propagation of the signal. This was implemented by precomputing a signal strength map for the whole office area: first, the space was evenly divided into a grid of cells. Then the received signal strength between every pair of cells was calculated by modeling the signal propagation and attentuation by integrating the effect of obstacles along a ray cast between the cells [9]. Note that for the purpose of localization the robots themselves were unaware of underlying topology.

A. Range-based localization

For each of these two scenarios, our primary interest is in obtaining relative localization using distance-based measurements. We therefore assume that whenever two robots can communicate, we can observe the underlying distance. To solve the localization problem, we collected odometry of every individual robot, together with joint connectivity data and associated distances, over a time horizon of 60 seconds[2], giving us $\delta(i,t), d_\theta(i,t), c(i,j,t), \varrho(i,j,t)$. Assembling the data across the nodes and time steps can be done by having every node broadcast its local odometry and connectivity data (at this point we do not consider the cost of communication). Since the underlying minimization problem in Eq. (10) does not admit a closed form solution, we employ the efficient and Hessian-free scaled conjugate gradients algorithm [5] as

[2]The simulation is updated once every 100msecs. To reduce communication, we only considered measurements at the rate of 1Hz. We subsampled the data and adjusted the odometry correspondingly.



(a) Range-based (left: outdoor, right: indoor)

(b) Connectivity-based (left: outdoor, right: indoor)

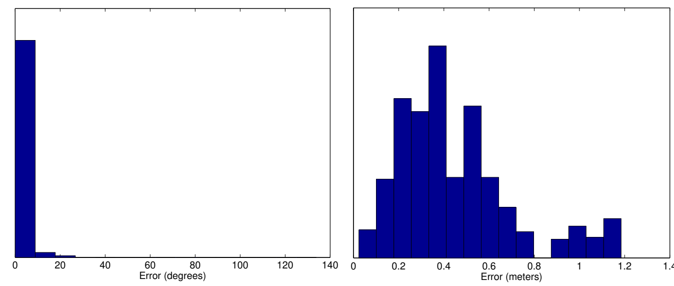
Fig. 3. Results for range-based and connectivity-based estimation of the relative positioning. The error is shown in terms of relative angular error (measured as angle between any two robots i and j, with respect to the heading of i), relative distance error, and reconstruction error (given as L2 norm between ground truth and reconstruction after coordinate system registration). Each bar shows the average over all robots and time steps, with the error bars corresponding to one standard deviation. Note the different scales of the y-axis. Figure 4 and 5 show in more detail the results for connectivity-based outdoor localization under low noise.

an iterative gradient-based solver. The computation of the gradient is described in Appendix A. Since the problem is nonconvex, we have to deal with multiple local minima. To ensure a high-quality solution, we ran the solver multiple times, each time restarting with a random initial value for $\vec{x}$. In our case we used 100 restarts and ran the optimizer for a maximum of 200 iterations. Note that in principle, because all the nodes in the network share the same data, this computation can be carried out in a distributed way.

Figure 3a shows the result for both scenarios using the same data. Since our goal is relative localization, we determine the quality of the reconstruction by taking the angular error between any two robots (i.e. the angle under which node i sees node j with respect to its own heading) and their relative distance. The plot shows the corresponding errors averaged over all pairs of robots and time steps (errorbars are given in one standard deviation). In addition, we also consider the effect of making noisy measurements: all of the observations (i.e. both odometry terms and the range) were corrupted by white noise with standard deviation of 1% (denoted as 'low') or 10% (denoted as 'medium').

B. Connectivity-based localization

In a second series of experiments, we repeated all of the above, this time assuming localization from connectivity alone. In this case, the relative locations were estimated by solving Eq. (11). The computation of the gradient is given in Appendix B. As we can see from the results in Figure 3b, the reconstruction error for both the outdoor and the indoor scenario is, unsurprisingly, substantially higher for connectivity-based localization than it is for range-based localization: in the outdoor case the angular error on the average is about 2-4 degrees, whereas with range-based localization it was less than 1 degree (with zero noise). For the case of 'low noise', Figure 4 shows in more detail how the error is distributed over its values. To give a visual impression of the quality of reconstruction, Figure 5 shows both the estimated and true locations, after fitting the relative coordinates to the true coordinates via Procrustes analysis [7]. Still, as most applications only require approximate localization, an error of this order of magnitude will usually be acceptable, considering that we can achieve this result from connectivity alone. Also note that our method for connectivity-based localization is far less susceptible to noise. On the other hand, under indoor conditions range-based localization still continues to work with negligible performance loss, whereas connectivity-based localization breaks down and produces an angular error of about 20 degrees on the average.



(a) Spread of angular error

(b) Spread of distance error

Fig. 4. Error for connectivity-based outdoor localization with low noise.

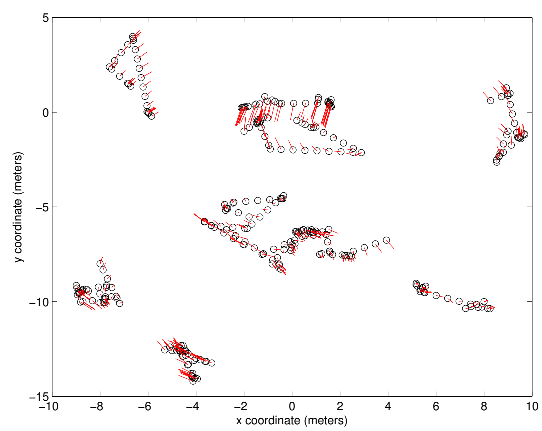


Fig. 5. Ground truth versus reconstruction (after performing registration). Plot shows individual paths of 10 robots over a time horizon of 60 seconds (one measurement every 1 second). The true locations are shown in black, the associated reconstructions are shown at the end of the red lines. The length of the red lines therefore indicates the reconstruction error. (Note that some paths contain fewer than 60 distinct points, in that case the robot remained stationary for some time.)

APPENDIX A: GRADIENT FOR RANGE-BASED

The gradient of Eq. (10) is computed as follows: the partial derivative $\partial_{x^{(\nu)}}$ for the ν-th component of $\vec{x}$, where $x^{(\nu)}$ is one of $x_\nu^{\text{start}}, y_\nu^{\text{start}}, \varphi_\nu^{\text{start}}$ for $\nu = 1 \ldots N$, is obtained from

$$\partial_{x^{(\nu)}} E = \sum_{t=1}^{T} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[\hat{\varrho}_{\vec{x}}(i,j,t) - \varrho(i,j,t)\right]^2 c(i,j,t) \cdot \\ \cdot \left[\partial_{x^{(\nu)}} \hat{\varrho}_{\vec{x}}(i,j,t)\right]. \quad (12)$$

Since only those terms in the sum where either index i or index j is equal to ν are unequal from zero, we can rewrite the summation in the following way:

$$\partial_{x^{(\nu)}} E = \sum_{t=1}^{T} \Bigg\{ \sum_{i=1}^{\nu-1} \left[\hat{\varrho}_{\vec{x}}(i,\nu,t) - \varrho(i,\nu,t)\right]^2 c(i,\nu,t) \cdot \\ \cdot \left[\partial_{x^{(\nu)}} \hat{\varrho}_{\vec{x}}(i,\nu,t)\right] + \\ \sum_{i=\nu+1}^{N} \left[\hat{\varrho}_{\vec{x}}(\nu,i,t) - \varrho(\nu,i,t)\right]^2 c(\nu,i,t) \cdot \\ \cdot \left[\partial_{x^{(\nu)}} \hat{\varrho}_{\vec{x}}(\nu,i,t)\right] \Bigg\} \quad (13)$$

To compute the derivative of the pairwise distances $\hat{\varrho}_{\vec{x}}$, we have to consider the following 6 different cases. First, consider the derivative with respect to the initial x-coordinate of the ν-th robot. Repeated application of the chain rule gives:

$$\partial_{x_\nu^{\text{start}}} \hat{\varrho}_{\vec{x}}(i,\nu,t) = -\left[\hat{\varrho}_{\vec{x}}(i,\nu,t)\right]^{-1} \hat{d}_x(i,\nu,t), \quad (14)$$

where $\hat{d}_x(i,\nu,t) := \hat{x}(i,t) - \hat{x}(\nu,t)$. Second, doing the same for the y-coordinate, we obtain:

$$\partial_{y_\nu^{\text{start}}} \hat{\varrho}_{\vec{x}}(i,\nu,t) = -\left[\hat{\varrho}_{\vec{x}}(i,\nu,t)\right]^{-1} \hat{d}_y(i,\nu,t), \quad (15)$$

where $\hat{d}_y(i,\nu,t) := \hat{y}(i,t) - \hat{y}(\nu,t)$. Third, for the derivative with respect to the initial heading angle $\varphi_\nu^{\text{start}}$, we obtain (repeated application of the chain rule):

$$\partial_{\varphi_\nu^{\text{start}}} \hat{\varrho}_{\vec{x}}(i,\nu,t) = \left[\hat{\varrho}_{\vec{x}}(i,\nu,t)\right]^{-1} \cdot \\ \cdot \left[\hat{d}_x(i,\nu,t)\hat{\beta}_{\vec{x}}(\nu,t) - \hat{d}_y(i,\nu,t)\hat{\alpha}_{\vec{x}}(\nu,t)\right], \quad (16)$$

where

$$\hat{\alpha}_{\vec{x}}(\nu,t) := \sum_{k=2}^{t} \delta(\nu,k) \cos\left(\varphi_\nu^{\text{start}} + \sum_{l=2}^{k} d_\theta(\nu,l)\right)$$
$$\hat{\beta}_{\vec{x}}(\nu,t) := \sum_{k=2}^{t} \delta(\nu,k) \sin\left(\varphi_\nu^{\text{start}} + \sum_{l=2}^{k} d_\theta(\nu,l)\right).$$

The remaining three cases correspond to the derivatives with respect to the first index in $\hat{\varrho}_{\vec{x}}(\nu,i,t)$, here we just have to invert the sign of Eqs. (14)-(16). Note that for the first time step $t=1$ the derivative with respect to $\varphi_\nu^{\text{start}}$ is zero, i.e.

$$\partial_{\varphi_\nu^{\text{start}}} \hat{\varrho}_{\vec{x}}(\nu,i,1) = \partial_{\varphi_\nu^{\text{start}}} \hat{\varrho}_{\vec{x}}(i,\nu,1) = 0.$$

APPENDIX B: GRADIENT FOR CONNECTIVITY-BASED

Likewise, to compute the gradient of Eq. (11), we start from

$$\partial_{x^{(\nu)}} E = \sum_{t=1}^{T} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \Phi'_{\mu,\sigma^2}\left(\hat{\varrho}_{\vec{x}}(i,j,t)\right) \left[\partial_{x^{(\nu)}} \hat{\varrho}_{\vec{x}}(i,j,t)\right] \cdot \\ \cdot \left(\frac{1 - c(i,j,t)}{\Phi_{\mu,\sigma^2}\left(\hat{\varrho}_{\vec{x}}(i,j,t)\right)} - \frac{c(i,j,t)}{1 - \Phi_{\mu,\sigma^2}\left(\hat{\varrho}_{\vec{x}}(i,j,t)\right)} \right),$$

where Φ' is derivative of the Gaussian cdf, i.e.

$$\Phi'_{\mu,\sigma^2}(\cdot) = \exp\left\{ -\frac{(\cdot - \mu)^2}{2\sigma^2} \right\} \cdot \frac{1}{\sqrt{2\pi\sigma^2}}.$$

Rewriting this sum similar to Eq. (13), we only need to plug in the previously computed partial derivates from Eqs. (14)-(16).

ACKNOWLEDGMENTS

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CNS-0615104), DARPA (FA8750-05-2-0283 and FA8650-08-C-7812), the Federal Highway Administration (DTFH61-07-H-00030), and General Motors. This particular research was enabled by cooperation and technical support from Vikram Manikonda and his team at IAI.

REFERENCES

[1] M. Ahmadi and P. Stone. Keeping in touch: Maintaining biconnected structure by homogeneous robots. In *Proc. of 21st National Conference on Artificial Intelligence*, pages 580–85, 2006.

[2] J. Bachrach and C. Taylor. Localization in sensor networks. In Ivan Stojmenovic, editor, *Handbook of Sensor Networks*. Wiley, 2005.

[3] B. Ferris, D. Fox, and N. Lawrence. WiFi SLAM using Gaussian process latent variable models. In *Proc. of Int. Joint Conf. on Artificial Intelligence*, 2007.

[4] B. Ferris, D. Hähnel, and D. Fox. Gaussian Processes for signal strength-based location estimation. In *Proc. of Robotics Science and Systems*, 2006.

[5] M. Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533, 1993.

[6] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *IPSN*, 2003.

[7] G. A. F. Seber. *Multivariate Observations*. Wiley, 1984.

[8] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from mere connectivity. In *MobiHoc' 03*, 2003.

[9] D. A. Shell and M. J. Mataric. High-fidelity radio signal strength modeling for robotic simulation. In *in submission IEEE IROS*, 2009.

[10] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.

[11] T. Yairi. Map building without localization by dimensionality reduction techniques. In *Proc. of 24th International Conference on Machine Learning (ICML)*, 2007.