Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Object Tracking by Scattered Directional Sensors

Kurt Plarre and P. R. Kumar

Abstract— **We address the problem of how to track objects moving at constant velocity using only directional sensors in a wireless sensor network. We model the field of vision of each directional sensor as a line, with the measured data being the times at which sensors detect objects crossing their lines. The network is initially deployed by scattering sensors, and the locations and directions in which the sensor point are also unknown a priori, in addition to the trajectories of the objects.**

The estimation problem involves the solution of a highly non-convex optimization problem. However we develop a three phase algorithm to solve the problem. It first chooses a coordinate basis adapted to the motions of the first two objects. Then it localizes the sensors with respect to that basis, and refines it as new objects arrive. Finally it transforms the basis if the GPS positions of six of the deployed nodes are known.

I. Introduction

Sensor networks integrate many tasks that were traditionally separated: Sensing, computation, communication, and actuation. They present a rich and challenging environment for the design of application algorithms, and it is accordingly useful to address *classes* of such problems.

In [1] we have studied the problem of detecting a static heat source with temperature sensors, which is a representative of problems using *omnidirectional* sensors to localize a stationary object.

In this paper we study a somewhat opposite problem which is representative of scenarios where highly *directional* sensors are deployed to track moving objects. A certain region is monitored by a network of directional sensors whose positions and orientations themselves are initially unknown; see Figure 1. The region is crossed sporadically by objects assumed to be moving at constant velocity at least within a bounded domain of interest. The task of the network is to estimate the trajectory of each object.

We model the "field of vision" of sensors as lines. A sensing node detects an object when it crosses its corresponding line. For each object, the data available to the sensors are thus the times at which sensors detect the object. The locations of the sensors and the directions in which the sensors point are unknown a priori. The sensor directions are also estimated as part of the problem. Section II presents a more detailed description of the setup.

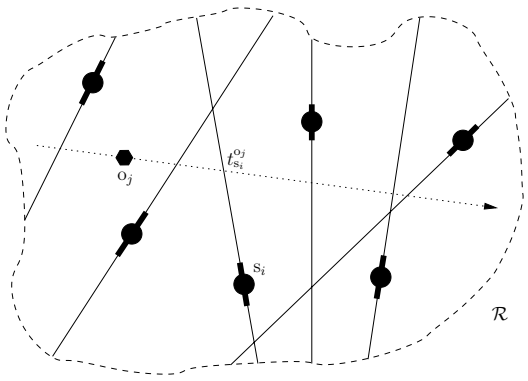

Fig. 1. Problem setup, and naming conventions.

The above described estimation problem involves the minimization of a non-convex function. We devise a three phase algorithm to solve this problem. In the first phase, the motions of the first four objects, and the sensor directions of six sensors are estimated. This problem can be solved in closed form (Section IV-A). During the second phase, the parameter estimates are updated, every time a new object is detected. The third phase is optional and consists of transforming the estimated parameters from an "internal" representation to the "real world" coordinate system, if the locations of six sensors are known somehow, (e.g., through GPS), or two trajectories are known.

To prolong the lifetime of the network, we make use of the "sleep state," a low power state in which sensors consume a small fraction of the power when "awake." After the first four objects have been detected, most sensors go to sleep. Only a few "sentinel nodes" remain awake. When a sentinel sensor detects a new object, it wakes up all other nodes (Section IV-B).

A survey on sensor networks can be found in [2]. Object tracking and sensor localization are important problems in sensor networks and have received much attention in the literature (e.g., [3], [4]). Optimization problems in sensor networks have also been studied (e.g., [1], [5]). The use of information provided by detected objects to improve the accuracy of localization schemes has also been proposed, although in a different context. In [4] connectivity information and information provided by objects detected by omnidirectional sensors are used to determine, for each sensor, a region in which it is located.

This material is based upon work partially supported by NSF under Contract Nos. NSF ANI 02-21357 and CCR-0325716, USARO under Contract Nos. DAAD19-00-1-0466 and DAAD19-01010-465, DARPA/AFOSR under Contract No. F49620-02-1-0325, DARPA under Contact Nos. N00014-0-1-1-0576 and F33615-0-1-C-1905, and AFOSR under Contract No. F49620-02-1-0217.

Kurt Plarre is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W. Main St, Urbana, Illinois, 61801, USA. plarre@uiuc.edu

P. R. Kumar is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W. Main St, Urbana, Illinois, 61801, USA. prkumar@uiuc.edu

0-7803-9568-9/05/$20.00 ©2005 IEEE

II. Problem description

A network of directional sensors $\{\mathrm{s}_1, \mathrm{s}_2, \ldots \mathrm{s}_m\}$ monitors a certain region $\mathcal{R}$ of the plane. Sporadically, objects moving at constant velocity cross $\mathcal{R}$. The size of each object is assumed negligible. A sensor detects an object when it crosses its "line of sight," which is simply modeled as a line, although half-lines and finite line segments can be handled by the algorithm without modification. See Figure 1.

Sensors are initially deployed by scattering them in $\mathcal{R}$. Their locations and the directions of their sensors are unknown a priori. We do not assume any distribution on the locations of the sensors, or the directions of their lines of sight, except for certain non-degeneracy conditions.

The goal of the system is to determine the orientations of the sensors, and the motions of all the objects, in some coordinate system. Later on, if at least six sensors know their exact locations (for example, using GPS), or two trajectories are known, then it is additionally possible to obtain the absolute positions with respect to the externally specified coordinate system.

As mentioned above, there are degenerate situations in which the proposed algorithm cannot be used, for example, if all sensor lines are perfectly parallel. We disregard such situations, and design the algorithm for a generic "good" case. Also, only one object can be handled by the algorithm, at a time, which is the case when objects arrive infrequently. Alternatively, one could assume that there is some separate mechanism, say the signature of an object, that allows sensors to associate crossing times with objects.

Note that the assumption that objects move in straight lines is not a limitation, since time samples can be ordered and split into segments, and the method applied to each segment, thus approximating the object trajectory by a piecewise affine function. This however requires that each time segment cross a certain number of sensors.

Let us denote the equation of the line of sight of sensor s_i as

$$\frac{x_{\mathrm{s}_i}}{a_{\mathrm{s}_i}} + \frac{y_{\mathrm{s}_i}}{b_{\mathrm{s}_i}} = 1 \quad \text{or} \quad \bar{a}_{\mathrm{s}_i} x_{\mathrm{s}_i} + \bar{b}_{\mathrm{s}_i} y_{\mathrm{s}_i} = 1\,.$$

The movement of object o_j will be modeled as

$$\begin{aligned} x_{\mathrm{o}_j}(t) &= v^{\mathrm{x}}_{\mathrm{o}_j} t + x^0_{\mathrm{o}_j}\,, \\ y_{\mathrm{o}_j}(t) &= v^{\mathrm{y}}_{\mathrm{o}_j} t + y^0_{\mathrm{o}_j}\,. \end{aligned}$$

The time at which sensor s_i detects object o_j is thus

$$t^{\mathrm{o}_j}_{\mathrm{s}_i} = \frac{1 - \bar{a}_{\mathrm{s}_i} x^0_{\mathrm{o}_j} - \bar{b}_{\mathrm{s}_i} y^0_{\mathrm{o}_j}}{\bar{a}_{\mathrm{s}_i} v^{\mathrm{x}}_{\mathrm{o}_j} + \bar{b}_{\mathrm{s}_i} v^{\mathrm{y}}_{\mathrm{o}_j}} + \nu^{\mathrm{o}_j}_{\mathrm{s}_i}\,,$$

where $\nu^{\mathrm{o}_j}_{\mathrm{s}_i}$ is assumed to be zero mean noise, with $\nu^{\mathrm{o}_j}_{\mathrm{s}_i}$ independent of $\nu^{\mathrm{o}_l}_{\mathrm{s}_k}$ for $(\mathrm{s}_i, \mathrm{o}_j) \neq (\mathrm{s}_k, \mathrm{o}_l)$.

If we associate with each $t^{\mathrm{o}_j}_{\mathrm{s}_i}$ the variable

$$\tau^{\mathrm{o}_j}_{\mathrm{s}_i} := (\bar{a}_{\mathrm{s}_i} v^{\mathrm{x}}_{\mathrm{o}_j} + \bar{b}_{\mathrm{s}_i} v^{\mathrm{y}}_{\mathrm{o}_j}) t^{\mathrm{o}_j}_{\mathrm{s}_i} + (\bar{a}_{\mathrm{s}_i} x^0_{\mathrm{o}_j} + \bar{b}_{\mathrm{s}_i} y^0_{\mathrm{o}_j}) - 1\,,$$

then the estimation of the object motion and sensor direction parameters can be posed as one of minimizing the cost function

$$J = \sum_{i,j} \left(\tau^{\mathrm{o}_j}_{\mathrm{s}_i}\right)^2\,, \tag{1}$$

where, for simplicity, the arguments of J, which are the unknown parameters, are not shown explicitly.

To show the difficulty of minimizing (1), we give the expanded form of J, for three sensors and two objects:

$$\begin{aligned} J = {} & \left[\left(\bar{a}_{\mathrm{s}_1} v^{\mathrm{x}}_{\mathrm{o}_1} + \bar{b}_{\mathrm{s}_1} v^{\mathrm{y}}_{\mathrm{o}_1}\right) t^{\mathrm{o}_1}_{\mathrm{s}_1} + \left(\bar{a}_{\mathrm{s}_1} x^0_{\mathrm{o}_1} + \bar{b}_{\mathrm{s}_1} y^0_{\mathrm{o}_1}\right) - 1\right]^2 + \\ & \left[\left(\bar{a}_{\mathrm{s}_1} v^{\mathrm{x}}_{\mathrm{o}_2} + \bar{b}_{\mathrm{s}_1} v^{\mathrm{y}}_{\mathrm{o}_2}\right) t^{\mathrm{o}_2}_{\mathrm{s}_1} + \left(\bar{a}_{\mathrm{s}_1} x^0_{\mathrm{o}_2} + \bar{b}_{\mathrm{s}_1} y^0_{\mathrm{o}_2}\right) - 1\right]^2 + \\ & \left[\left(\bar{a}_{\mathrm{s}_2} v^{\mathrm{x}}_{\mathrm{o}_1} + \bar{b}_{\mathrm{s}_2} v^{\mathrm{y}}_{\mathrm{o}_1}\right) t^{\mathrm{o}_1}_{\mathrm{s}_2} + \left(\bar{a}_{\mathrm{s}_2} x^0_{\mathrm{o}_1} + \bar{b}_{\mathrm{s}_2} y^0_{\mathrm{o}_1}\right) - 1\right]^2 + \\ & \left[\left(\bar{a}_{\mathrm{s}_2} v^{\mathrm{x}}_{\mathrm{o}_2} + \bar{b}_{\mathrm{s}_2} v^{\mathrm{y}}_{\mathrm{o}_2}\right) t^{\mathrm{o}_2}_{\mathrm{s}_2} + \left(\bar{a}_{\mathrm{s}_2} x^0_{\mathrm{o}_2} + \bar{b}_{\mathrm{s}_2} y^0_{\mathrm{o}_2}\right) - 1\right]^2 + \\ & \left[\left(\bar{a}_{\mathrm{s}_3} v^{\mathrm{x}}_{\mathrm{o}_1} + \bar{b}_{\mathrm{s}_3} v^{\mathrm{y}}_{\mathrm{o}_1}\right) t^{\mathrm{o}_1}_{\mathrm{s}_3} + \left(\bar{a}_{\mathrm{s}_3} x^0_{\mathrm{o}_1} + \bar{b}_{\mathrm{s}_3} y^0_{\mathrm{o}_1}\right) - 1\right]^2 + \\ & \left[\left(\bar{a}_{\mathrm{s}_3} v^{\mathrm{x}}_{\mathrm{o}_2} + \bar{b}_{\mathrm{s}_3} v^{\mathrm{y}}_{\mathrm{o}_2}\right) t^{\mathrm{o}_2}_{\mathrm{s}_3} + \left(\bar{a}_{\mathrm{s}_3} x^0_{\mathrm{o}_2} + \bar{b}_{\mathrm{s}_3} y^0_{\mathrm{o}_2}\right) - 1\right]^2\,. \end{aligned} \tag{2}$$

Note that this is a non-convex function of

$$\{(\bar{a}_{\mathrm{s}_i}, \bar{b}_{\mathrm{s}_i}, v^{\mathrm{x}}_{\mathrm{o}_j}, x^0_{\mathrm{o}_j}, v^{\mathrm{y}}_{\mathrm{o}_j}, y^0_{\mathrm{o}_j});\ 1 \le i \le 3,\ 1 \le j \le 2\}\,,$$

the sensor and object parameters.

In addition the number of parameters to determine is very large. For example, even for four objects and six sensors, the number of parameters is $4 \times 4 + 6 \times 2 = 28$. Only the global minimum is an acceptable solution, and only an exhaustive search could ensure that one finds it.

To estimate the minimum of (1) we devise a novel two phase algorithm with an optional third phase that corresponds to the final coordinate transformation. We will develop a recursive algorithm by which the data provided by four objects and six sensors is used to determine an initial solution. The data provided by other sensors and objects is recursively incorporated into the algorithm, thus improving the accuracy of the solution. Details are given in Sections IV-A and IV-B. Section V presents simulations of the algorithm for an example application, and Section VI presents an actual implementation.

III. The main ideas

It is evident that finding the global minimum of a non-convex cost function, such as (1), directly, is a very difficult task. We thus divide the process into two phases. In the first phase, we use the data obtained from only m sensors and n objects, where m and n are chosen in such a way that (1) can be set exactly to zero, independent of the noise. Solving the resulting equation provides an initial estimate of the parameters. In the second phase, as new data is incorporated into the problem, the sensor and object parameter estimates are refined, using a local improvement algorithm.

We note that since we do not know the "real world" coordinate system we must choose a "custom" system in which to state the equations and thus localize the sensor rotations, and the motions of the objects. Later on we will use the locations of six sensors, if known, to transform the so obtained parameters to the correct representation. This can be done at any point of the algorithm.

Since we are free to choose our coordinate system, we will choose it in such a way that it simplifies the estimation problem. In fact, if the coordinate system is not carefully chosen, the resulting equations cannot be solved in closed form. We thus have the task of finding the right coordinate system in which to write the equations, and then finding a procedure to solve them.

We choose the "adaptive" coordinate system in the following way:

1) The motion of the first object is used to fix the "horizontal $x-$axis," with its position at time $t = 0$ as the origin, and speed normalized to 1. As we will see in Section IV, this fixes all parameters of o_1 in the custom system.
2) The motion of the second object is used to fix the "vertical $y-$axis," with its speed also normalized to 1. However, since its position at time $t = 0$ is unknown, two parameters corresponding to o_2 will be undetermined (Section IV).

To determine the number of sensors, and object measurements, needed to find the initial estimates, i.e., n and m, we reason in the following way:

1) Each remaining object, o_j, used in the first phase will add four unknown parameters to the problem: $v^{\mathrm{x}}_{\mathrm{o}_j}$, $x^0_{\mathrm{o}_j}$, $v^{\mathrm{y}}_{\mathrm{o}_j}$, and $y^0_{\mathrm{o}_j}$.
2) Each sensor s_i included in this phase will add two unknown parameters to define its "line."
3) On the other hand, the number of data measurements obtained from the detection of the first n objects by m sensors is nm.

Considering that we need at least the same number of data variables as the number of unknown parameters to solve the equations, we need:

$$nm \geq 4(n-2) + 2 + 2m\,,$$

which is satisfied by $m = 6$, and $n = 3$. We thus need at least six sensors and three objects to initialize the system. However we will see in Section IV-A that the resulting equation is quadradic, and we will need the data from a fourth object to resolve the sign of the root.

IV. Algorithm

In this section we present the estimation algorithm. An actual implementation is presented in Section VI.

A. First phase

During this first phase, after deployment, all sensors are awake, waiting for the first four objects. The data collected from these objects is used to form an initial estimate of the object and sensor parameters.

As mentioned above, the first object is used to fix the "horizontal $x-$axis" of the adaptive coordinate system. The point on the plane at which o_1 was at time $t = 0$, is chosen as the origin of the coordinate system. The direction of motion determines the axis, and the scale is given by assuming that the speed of o_1 is 1.

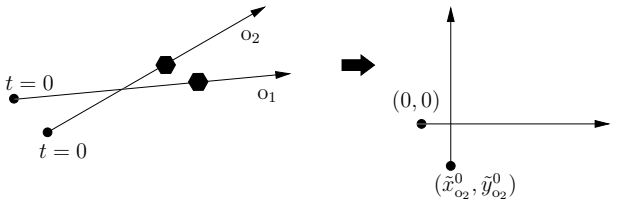

Fig. 2. Adaptive coordinate system obtained from the trajectories of the first two objects.

The second object fixes the vertical axis. The direction of motion of o_2 determines the "vertical $y-$axis," while the scale is given by assuming that its speed is 1. The point at which o_2 was at time $t = 0$ is unknown. We call this point $(\tilde{x}^0_{\mathrm{o}_2}, \tilde{y}^0_{\mathrm{o}_2})$. These parameters are unknown even with respect to the adaptive basis and must be estimated as part of the problem.

In our coordinate system, we know that the line corresponding to sensor s_i passes through the points $(t^{\mathrm{o}_1}_{\mathrm{s}_i}, 0)$ and $(\tilde{x}^0_{\mathrm{o}_2}, \tilde{y}^0_{\mathrm{o}_2} + t^{\mathrm{o}_2}_{\mathrm{s}_i})$. Thus, the equation for s_i in this system is determined as

$$\frac{\tilde{y}_{\mathrm{s}_i}}{\tilde{x}_{\mathrm{s}_i} - t^{\mathrm{o}_1}_{\mathrm{s}_i}} = \frac{\tilde{y}^0_{\mathrm{o}_2} + t^{\mathrm{o}_2}_{\mathrm{s}_i}}{\tilde{x}^0_{\mathrm{o}_2} - t^{\mathrm{o}_1}_{\mathrm{s}_i}}\,. \tag{3}$$

Thus, subject only to $(\tilde{x}^0_{\mathrm{o}_2}, \tilde{y}^0_{\mathrm{o}_2})$ being unknown, each sensor's line is determined.

Now we turn to the second object. Reordering, (3) we obtain

$$(\tilde{x}^0_{\mathrm{o}_2} - t^{\mathrm{o}_1}_{\mathrm{s}_i})\tilde{y}_{\mathrm{s}_i} = (\tilde{y}^0_{\mathrm{o}_2} + t^{\mathrm{o}_2}_{\mathrm{s}_i})\tilde{x}_{\mathrm{s}_i} - t^{\mathrm{o}_2}_{\mathrm{s}_i}\tilde{y}^0_{\mathrm{o}_2} - t^{\mathrm{o}_1}_{\mathrm{s}_i}t^{\mathrm{o}_2}_{\mathrm{s}_i}\,. \tag{4}$$

Consider now the third object o_3. Assume that the equation for o_3 in our coordinate system is

$$\tilde{x}_{\mathrm{o}_3}(t) = \tilde{v}^{\mathrm{x}}_{\mathrm{o}_3}t + \tilde{x}^0_{\mathrm{o}_3}\ \ ,\ \ \tilde{y}_{\mathrm{o}_3}(t) = \tilde{v}^{\mathrm{y}}_{\mathrm{o}_3}t + \tilde{y}^0_{\mathrm{o}_3}\,.$$

We know o_3 is detected by sensor s_i at time $t^{\mathrm{o}_3}_{\mathrm{s}_i}$. Combining this information with (4), we obtain

$$\begin{aligned}(\tilde{x}^0_{\mathrm{o}_2} - t^{\mathrm{o}_1}_{\mathrm{s}_i})(\tilde{y}^0_{\mathrm{o}_3} + \tilde{v}^{\mathrm{y}}_{\mathrm{o}_3}t^{\mathrm{o}_3}_{\mathrm{s}_i}) = \\ (\tilde{y}^0_{\mathrm{o}_2} + t^{\mathrm{o}_2}_{\mathrm{s}_i})(\tilde{x}^0_{\mathrm{o}_3} + \tilde{v}^{\mathrm{x}}_{\mathrm{o}_3}t^{\mathrm{o}_3}_{\mathrm{s}_i}) - t^{\mathrm{o}_1}_{\mathrm{s}_i}\tilde{y}^0_{\mathrm{o}_2} - t^{\mathrm{o}_1}_{\mathrm{s}_i}t^{\mathrm{o}_2}_{\mathrm{s}_i}\,.\end{aligned} \tag{5}$$

Let $\mathbf{M}$ be a matrix such that its $i-$th row is

$$\begin{aligned}\mathbf{M}_{i,*} \ := \ & \big[\tilde{x}^0_{\mathrm{o}_2} - t^{\mathrm{o}_1}_{\mathrm{s}_i}, t^{\mathrm{o}_3}_{\mathrm{s}_i}(\tilde{x}^0_{\mathrm{o}_2} - t^{\mathrm{o}_2}_{\mathrm{s}_i}), -(\tilde{y}^0_{\mathrm{o}_2} + t^{\mathrm{o}_2}_{\mathrm{s}_i}), \\ & -t^{\mathrm{o}_3}_{\mathrm{s}_i}(\tilde{y}^0_{\mathrm{o}_2} + t^{\mathrm{o}_2}_{\mathrm{s}_i}), (t^{\mathrm{o}_1}_{\mathrm{s}_i}\tilde{y}^0_{\mathrm{o}_2} + t^{\mathrm{o}_1}_{\mathrm{s}_i}t^{\mathrm{o}_2}_{\mathrm{s}_i})\big]\,.\end{aligned} \tag{6}$$

Likewise, let $\mathbf{v} := [\tilde{y}^0_{\mathrm{o}_3}, \tilde{v}^{\mathrm{y}}_{\mathrm{o}_3}, \tilde{x}^0_{\mathrm{o}_3}, \tilde{v}^{\mathrm{x}}_{\mathrm{o}_3}, 1]^{\mathrm{T}}$. Then, from (5), we can write the linear system $\mathbf{M}\mathbf{v} = 0$. For this system to have a nontrivial solution, $\mathbf{M}$ must be column rank deficient. Let us rewrite $\mathbf{M}$ in term of its columns. For this, let us first define

$$\begin{aligned}\bar{e} \ &:= \ [1, 1, \ldots, 1]^{\mathrm{T}}\,, \\ \bar{T}_{\mathrm{o}_1} \ &:= \ [t^{\mathrm{o}_1}_{\mathrm{s}_1}, t^{\mathrm{o}_1}_{\mathrm{s}_2}, \ldots, t^{\mathrm{o}_1}_{\mathrm{s}_m}]^{\mathrm{T}}\,, \\ \bar{T}_{\mathrm{o}_2} \ &:= \ [t^{\mathrm{o}_2}_{\mathrm{s}_1}, t^{\mathrm{o}_2}_{\mathrm{s}_2}, \ldots, t^{\mathrm{o}_2}_{\mathrm{s}_m}]^{\mathrm{T}}\,, \\ \bar{T}_{\mathrm{o}_3} \ &:= \ [t^{\mathrm{o}_3}_{\mathrm{s}_1}, t^{\mathrm{o}_3}_{\mathrm{s}_2}, \ldots, t^{\mathrm{o}_3}_{\mathrm{s}_m}]^{\mathrm{T}}\,, \\ \bar{T}^{\mathrm{o}_2}_{\mathrm{o}_1} \ &:= \ [t^{\mathrm{o}_1}_{\mathrm{s}_1}t^{\mathrm{o}_2}_{\mathrm{s}_1}, t^{\mathrm{o}_1}_{\mathrm{s}_2}t^{\mathrm{o}_2}_{\mathrm{s}_2}, \ldots, t^{\mathrm{o}_1}_{\mathrm{s}_m}t^{\mathrm{o}_2}_{\mathrm{s}_m}]^{\mathrm{T}}\,, \\ \bar{T}^{\mathrm{o}_3}_{\mathrm{o}_1} \ &:= \ [t^{\mathrm{o}_1}_{\mathrm{s}_1}t^{\mathrm{o}_3}_{\mathrm{s}_1}, t^{\mathrm{o}_1}_{\mathrm{s}_2}t^{\mathrm{o}_3}_{\mathrm{s}_2}, \ldots, t^{\mathrm{o}_1}_{\mathrm{s}_m}t^{\mathrm{o}_3}_{\mathrm{s}_m}]^{\mathrm{T}}\,, \\ \bar{T}^{\mathrm{o}_3}_{\mathrm{o}_2} \ &:= \ [t^{\mathrm{o}_2}_{\mathrm{s}_1}t^{\mathrm{o}_3}_{\mathrm{s}_1}, t^{\mathrm{o}_2}_{\mathrm{s}_2}t^{\mathrm{o}_3}_{\mathrm{s}_2}, \ldots, t^{\mathrm{o}_2}_{\mathrm{s}_m}t^{\mathrm{o}_3}_{\mathrm{s}_m}]^{\mathrm{T}}\,.\end{aligned}$$

With these definitions we can write $\mathbf{M}$ as

$$\begin{aligned}\mathbf{M} &= \big[\tilde{x}^0_{\mathrm{o}_2}\bar{e} - \bar{T}_{\mathrm{o}_1},\, \tilde{x}^0_{\mathrm{o}_2}\bar{T}_{\mathrm{o}_3} - \bar{T}^{\mathrm{o}_3}_{\mathrm{o}_1},\, -\tilde{y}^0_{\mathrm{o}_2}\bar{e} - \bar{T}_{\mathrm{o}_2},\\ &\quad -\tilde{y}^0_{\mathrm{o}_2}\bar{T}_{\mathrm{o}_3} - \bar{T}^{\mathrm{o}_3}_{\mathrm{o}_2},\, \tilde{y}^0_{\mathrm{o}_2}\bar{T}_{\mathrm{o}_1} + \bar{T}^{\mathrm{o}_2}_{\mathrm{o}_1}\big] \qquad (7)\end{aligned}$$

Since $\mathbf{M}$ is column rank deficient, there exist real numbers α_1, α_2, α_3, α_4, α_5, such that

$$\begin{aligned}&\alpha_1(\tilde{x}^0_{\mathrm{o}_2}\bar{e} - \bar{T}_{\mathrm{o}_1}) + \alpha_2(\tilde{x}^0_{\mathrm{o}_2}\bar{T}_{\mathrm{o}_3} - \bar{T}^{\mathrm{o}_3}_{\mathrm{o}_1}) +\\ &\alpha_3(-\tilde{y}^0_{\mathrm{o}_2}\bar{e} - \bar{T}_{\mathrm{o}_2}) + \alpha_4(-\tilde{y}^0_{\mathrm{o}_2}\bar{T}_{\mathrm{o}_3} - \bar{T}^{\mathrm{o}_3}_{\mathrm{o}_2}) +\\ &\qquad \alpha_5(\tilde{y}^0_{\mathrm{o}_2}\bar{T}_{\mathrm{o}_1} + \bar{T}^{\mathrm{o}_2}_{\mathrm{o}_1}) \;=\; 0\,. \qquad (8)\end{aligned}$$

Collecting terms, and defining

$$\begin{aligned}\bar{\mathbf{M}} &:= \big[\bar{e}, \bar{T}_{\mathrm{o}_1}, \bar{T}_{\mathrm{o}_3}, \bar{T}_{\mathrm{o}_2}, \bar{T}^{\mathrm{o}_3}_{\mathrm{o}_1}, \bar{T}^{\mathrm{o}_3}_{\mathrm{o}_2}, \bar{T}^{\mathrm{o}_2}_{\mathrm{o}_1}\big]\,,\\ \bar{\mathbf{v}} &:= \big[\alpha_1\tilde{x}^0_{\mathrm{o}_2} - \alpha_3\tilde{y}^0_{\mathrm{o}_2}, \alpha_5\tilde{y}^0_{\mathrm{o}_2} - \alpha_1, \alpha_2\tilde{x}^0_{\mathrm{o}_2} - \alpha_4\tilde{y}^0_{\mathrm{o}_2},\\ &\quad -\alpha_3, -\alpha_2, -\alpha_2, -\alpha_4, \alpha_5\big]^{\mathrm{T}}\,,\end{aligned}$$

we can rewrite (8) as $\bar{\mathbf{M}}\bar{\mathbf{v}} = 0$. Let $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^{\mathrm{T}}$ be the solution to this equation, with $\alpha_5 := 1$. Then

$$\begin{aligned}\alpha_1\tilde{x}^0_{\mathrm{o}_2} - \alpha_3\tilde{y}^0_{\mathrm{o}_2} &= \theta_1\,, & -\alpha_3 &= \theta_4\,,\\ \alpha_5\tilde{y}^0_{\mathrm{o}_2} - \alpha_1 &= \theta_2\,, & -\alpha_2 &= \theta_5\,,\\ \alpha_2\tilde{x}^0_{\mathrm{o}_2} - \alpha_4\tilde{y}^0_{\mathrm{o}_2} &= \theta_3\,, & -\alpha_4 &= \theta_6\,,\end{aligned}$$

Solving this nonlinear system one obtains

$$\tilde{x}^0_{\mathrm{o}_2} = \frac{\frac{\alpha_5\theta_3 + \alpha_4\theta_2 + \alpha_2\alpha_3}{2\alpha_2\alpha_5} \pm \sqrt{\frac{(\alpha_5\theta_3 + \alpha_4\theta_2 + \alpha_2\alpha_3)^2 + 4\alpha_2\alpha_5(\alpha_4\theta_1 - \alpha_3\theta_3)}{2\alpha_2\alpha_5}}}{\qquad}. \qquad (9)$$

To resolve the sign in (9) we make use of the data provided by the fourth object o_4. We simply choose the sign that best explains the detection times $t^{\mathrm{o}_4}_{\mathrm{s}_i}$.

Once the value of $\tilde{x}^0_{\mathrm{o}_2}$ is known, remaining parameters can be easily computed.

B. Second phase

Once the parameters for the first four objects and six sensors have been estimated, most sensors go to sleep. A few sentinel sensors stay awake and sensing. When a sentinel sensor detects an object, it wakes up the complete sensor network. All sensors then wait for the object and register the time at which they detect it. It is important to note that we allow for the possibility that some sensors will not detect a given object, since they may wake up too late. This is illustrated in Figure 3.

Each sensor has at most one detection time for the new object. To form an estimate of the trajectory of this object, at least four measurements are necessary. To gather this information, sensors share their measurements (if they have one), and collect measurements from other nodes. The obtained data are used to refine the estimates of all parameters.

For each s_k, and object o_l, let $\mathcal{O}^{\mathrm{s}_i}_{\mathrm{s}_k} := \{l | \mathrm{s}_i \text{ knows } t^{\mathrm{o}_l}_{\mathrm{s}_k}\}$, and $\mathcal{S}^{\mathrm{s}_i}_{\mathrm{o}_l} := \{k | \mathrm{s}_i \text{ knows } t^{\mathrm{o}_l}_{\mathrm{s}_k}\}$. The cost at s_i is then given by

$$\begin{aligned}J_{\mathrm{s}_i} &= \sum_k \sum_{l\in\mathcal{O}^{\mathrm{s}_i}_{\mathrm{s}_k}} \big[\bar{a}^{\mathrm{s}_i}_{\mathrm{s}_k} v^{\mathrm{x},\mathrm{s}_i}_{\mathrm{o}_l} t^{\mathrm{o}_l}_{\mathrm{s}_k} + \bar{a}^{\mathrm{s}_i}_{\mathrm{s}_k} x^{0,\mathrm{s}_i}_{\mathrm{o}_l} +\\ &\qquad \bar{b}^{\mathrm{s}_i}_{\mathrm{s}_k} v^{\mathrm{y},\mathrm{s}_i}_{\mathrm{o}_l} t^{\mathrm{o}_l}_{\mathrm{s}_k} + \bar{b}^{\mathrm{s}_i}_{\mathrm{s}_k} y^{0,\mathrm{s}_i}_{\mathrm{o}_l} - 1\big]^2\,. \qquad (10)\end{aligned}$$

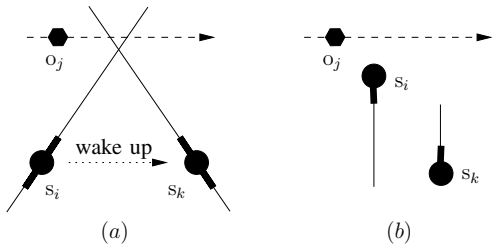

Fig. 3. Some objects are not detected by all sensors: (a) s_k wakes up too late to detect o_j, (b) s_i only covers a half-line, while s_k has a limited range.

where $\bar{a}^{\mathrm{s}_i}_{\mathrm{s}_k}$, $\bar{b}^{\mathrm{s}_i}_{\mathrm{s}_k}$, $v^{\mathrm{x},\mathrm{s}_i}_{\mathrm{o}_l}$, $x^{0,\mathrm{s}_i}_{\mathrm{o}_l}$, $v^{\mathrm{y},\mathrm{s}_i}_{\mathrm{o}_l}$, and $y^{0,\mathrm{s}_i}_{\mathrm{o}_l}$ are the estimated parameters at s_i. We use a block coordinate descent method (see [6]) to minimize J_{s_i}. Sensor s_i performs one step of Newton's algorithm for the parameters of each sensor and object for which it has enough data.

For example, for each sensor s_k, for which it has enough data, s_i computes

$$\begin{aligned}A^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l} &:= v^{\mathrm{x},\mathrm{s}_i}_{\mathrm{o}_l} t^{\mathrm{o}_l}_{\mathrm{s}_k} + x^{0,\mathrm{s}_i}_{\mathrm{o}_l}\,, & D^{\mathrm{s}_i}_{\mathrm{o}_k,\mathrm{o}_l} &:= \bar{a}^{\mathrm{s}_i}_{\mathrm{s}_k} t^{\mathrm{o}_l}_{\mathrm{s}_k}\,,\\ B^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l} &:= v^{\mathrm{y},\mathrm{s}_i}_{\mathrm{o}_l} t^{\mathrm{o}_l}_{\mathrm{s}_k} + y^{0,\mathrm{s}_i}_{\mathrm{o}_l}\,, & E^{\mathrm{s}_i}_{\mathrm{o}_k,\mathrm{o}_l} &:= \bar{b}^{\mathrm{s}_i}_{\mathrm{s}_k}\,,\\ C^{\mathrm{s}_i}_{\mathrm{o}_k,\mathrm{o}_l} &:= \bar{a}^{\mathrm{s}_i}_{\mathrm{s}_k}\,, & F^{\mathrm{s}_i}_{\mathrm{o}_k,\mathrm{o}_l} &:= \bar{b}^{\mathrm{s}_i}_{\mathrm{s}_k} t^{\mathrm{o}_l}_{\mathrm{s}_k}\,.\end{aligned}$$

With these definitions we can rewrite (10) as

$$\begin{aligned}J_{\mathrm{s}_i} &= \sum_k \sum_{l\in\mathcal{O}^{\mathrm{s}_i}_{\mathrm{s}_k}} \big[A^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l}\bar{a}^{\mathrm{s}_i}_{\mathrm{s}_k} + B^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l}\bar{b}^{\mathrm{s}_i}_{\mathrm{s}_k} - 1\big]^2\\ &= \sum_l \sum_{k\in\mathcal{S}^{\mathrm{s}_i}_{\mathrm{o}_l}} \big[D^{\mathrm{s}_i}_{\mathrm{o}_k,\mathrm{o}_l} v^{\mathrm{x},\mathrm{s}_i}_{\mathrm{o}_l} + C^{\mathrm{s}_i}_{\mathrm{o}_k,\mathrm{o}_l} x^{0,\mathrm{s}_i}_{\mathrm{o}_l} +\\ &\qquad F^{\mathrm{s}_i}_{\mathrm{o}_k,\mathrm{o}_l} v^{\mathrm{y},\mathrm{s}_i}_{\mathrm{o}_l} + E^{\mathrm{s}_i}_{\mathrm{o}_k,\mathrm{o}_l} y^{0,\mathrm{s}_i}_{\mathrm{o}_l} - 1\big]^2\,. \qquad (11)\end{aligned}$$

Let $\mathbf{v}^{\mathrm{s}_i}_{\mathrm{s}_k} := \big[\bar{a}^{\mathrm{s}_i}_{\mathrm{s}_k}, \bar{b}^{\mathrm{s}_i}_{\mathrm{s}_k}\big]^{\mathrm{T}}$. The gradient of J_{s_i} with respect to $\mathbf{v}^{\mathrm{s}_i}_{\mathrm{s}_k}$ is

$$\begin{aligned}\mathbf{g}^{\mathrm{s}_i}_{\mathrm{s}_k} := \Bigg[&\sum_{l\in\mathcal{O}^{\mathrm{s}_i}_{\mathrm{s}_k}} (A^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l}\bar{a}^{\mathrm{s}_i}_{\mathrm{s}_k} + B^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l}\bar{b}^{\mathrm{s}_i}_{\mathrm{s}_k} - 1)A^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l},\\ &\sum_{l\in\mathcal{O}^{\mathrm{s}_i}_{\mathrm{s}_k}} (A^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l}\bar{a}^{\mathrm{s}_i}_{\mathrm{s}_k} + B^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l}\bar{b}^{\mathrm{s}_i}_{\mathrm{s}_k} - 1)B^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l}\Bigg]^{\mathrm{T}}\,,\end{aligned}$$

and the Hessian is given by

$$\mathbf{H}^{\mathrm{s}_i}_{\mathrm{s}_k} := \begin{bmatrix} \sum_{l\in\mathcal{O}^{\mathrm{s}_i}_{\mathrm{s}_k}} (A^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l})^2 & \sum_{l\in\mathcal{O}^{\mathrm{s}_i}_{\mathrm{s}_k}} A^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l} B^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l} \\ \sum_{l\in\mathcal{O}^{\mathrm{s}_i}_{\mathrm{s}_k}} B^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l} A^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l} & \sum_{l\in\mathcal{O}^{\mathrm{s}_i}_{\mathrm{s}_k}} (B^{\mathrm{s}_i}_{\mathrm{s}_k,\mathrm{o}_l})^2 \end{bmatrix}.$$

Applying Newton's method to (10) with respect to $\bar{a}_{\mathrm{s}_k}$ and $\bar{b}_{\mathrm{s}_k}$, we obtain the recursion

$$\mathbf{v}^{\mathrm{s}_i}_{\mathrm{s}_k} \leftarrow \mathbf{v}^{\mathrm{s}_i}_{\mathrm{s}_k} - \big(\mathbf{H}^{\mathrm{s}_i}_{\mathrm{s}_k}\big)^{-1}\mathbf{g}^{\mathrm{s}_i}_{\mathrm{s}_k}\,.$$

Similar expressions are obtained, if Newton's method is applied to (10), with respect to $v^{\mathrm{x},\mathrm{s}_i}_{\mathrm{o}_l}$, $x^{0,\mathrm{s}_i}_{\mathrm{o}_l}$, $v^{\mathrm{y},\mathrm{s}_i}_{\mathrm{o}_l}$, and $y^{0,\mathrm{s}_i}_{\mathrm{o}_l}$.

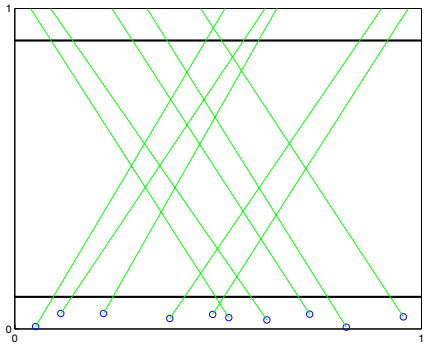

Fig. 4. Setup for simulations. Sensors are shown as circles along the bottom of the fi gure; their directions are shown by lines. The dark parallel horizontal lines indicate the boundaries of the region of interest. Objects move from left to right but not along parallel trajectories.

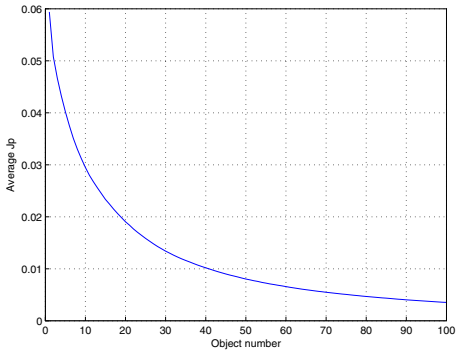

Fig. 5. Average estimation error (Jp), as a function of the number of detected objects, for 100 different runs of the algorithm.

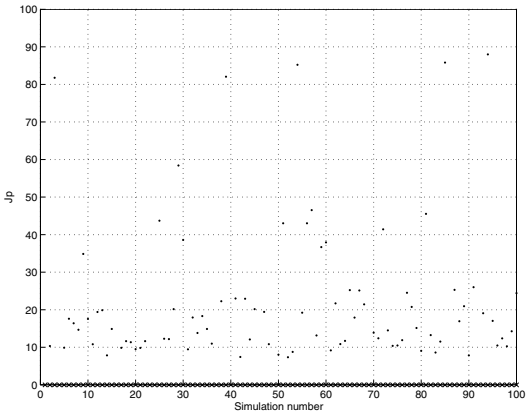

Fig. 6. Error in parameter estimates given by the adaptive basis algorithm (crosses), and a randomly restarted local improvement algorithm (dots).

V. Simulations

We have conducted simulations, followed by an implementation. In this section, we present the results of our preliminary simulation studies. In the next section, we describe the results of the experimentation.

Figure 4 shows the setup for the simulations. A section of a passage (say, a road, bridge, or tunnel) is monitored by a collection of m sensors located along one side of the passage. The length of the section is L, and its width is W. In the simulations, for simplicity $L = W = 1$. Sensors are located regularly, except for noise in their positions, and the angles of their lines of sight are approximately 63 degrees. The exact angles of the sensors must be recovered from the measurements, as part of the problem. We have purposely avoided situations in which sensors are "close to vertical" or "close to horizontal," since such situations produce numerical problems. The measurement errors are uniformly distributed in $[-0.01, 0.01]$. Objects enter the section from the left and exit it from the right. The speed of the objects is chosen uniformly and independently in the range $[0.01, 0.1]$, while their trajectories are fixed by choosing random entry and exit points. To ensure that the two first trajectories are not parallel, they are fixed: the first trajectory entering and exiting at the bottom, and the second trajectory entering at the bottom and exiting at the top (thus maximizing the angle between them).

The estimation of the sensor and object parameters is done by minimizing the quadratic cost function (1), although the quality of the resulting estimates is assessed by the cost defined by

$$\begin{aligned}(2\bar{m}+4\bar{n})J_p \;:=\; & \sum_{i=1}^{\bar{m}}\left[(\hat{a}_{\mathrm{s}_i}-a_{\mathrm{s}_i})^2+(\hat{b}_{\mathrm{s}_i}-b_{\mathrm{s}_i})^2\right]+\\ & \sum_{j=1}^{\bar{n}}\Big[(\hat{v}^{\mathrm{x}}_{\mathrm{o}_j}-v^{\mathrm{x}}_{\mathrm{o}_j})^2+(\hat{x}^0_{\mathrm{o}_j}-x^0_{\mathrm{o}_j})^2+\\ & \qquad (\hat{v}^{\mathrm{y}}_{\mathrm{o}_j}-v^{\mathrm{y}}_{\mathrm{o}_j})^2+(\hat{y}^0_{\mathrm{o}_j}-y^0_{\mathrm{o}_j})^2\Big]\,,\end{aligned}$$

where $\bar{m}$, and $\bar{n}$ are the number of sensors and objects, respectively. The behavior of J_p for the first 100 objects (after the passage of the initial four objects necessary to initialize the algorithm), for 100 different runs of the algorithm is shown in Figure 5. The curve shown corresponds to an average over the 100 runs of the simulation. It is clear from Figure 5 that the quality of the estimation improves with the number of detected objects, which is desirable. It is important to mention the importance of the refining step to improve the performance of the algorithm when measurements are noisy.

To illustrate the importance of the first phase of the algorithm, we compare in Figure 6 the error in the parameter estimates, J_p, for the first six sensors and four objects, given by the adaptive basis algorithm (crosses), versus that of a randomly restarted local improvement algorithm (dots). In each simulation, the local improvement algorithm was restarted at 100 different points, and the best parameter estimates chosen, as the ones minimizing (1). The random initialization points were obtained in the same fashion as the actual parameters of sensors and objects. No noise in the data was considered. It can be seen in Figure 6 that the local improvement algorithm is unable to find the optimum parameter estimates, in contrast to the adaptive basis algorithm.

VI. Implementation

The algorithm was next implemented on Berkeley mica2 [7] motes provided with light sensors, and lasers that were pointed at the sensors. Sensors were installed on one side of a track and lasers on the other side. A toy car was run through this sensor field, acting as "object." The track was 16 feet long and 8 feet wide. The speed of the car was approximately constant, equal to 1.41 feet per second. A picture of the setup can be seen in Figure 7.

As the car runs along the track, it interrupted the lasers. The interruption times were recorded by the sensors. Six sensors $\{s_1, ..., s_6\}$ were used. Each sensor transmitted the recorded time to a seventh sensor s_7, which performed the computations. After four runs, s_7 estimated the angles of the sensors, and the entry and exit points of the last run. After that, for every new run, s_7 updated the estimated parameters and computed the entry and exit points of the latest run. Figure 8 shows the estimated trajectories for four runs in an experiment with a total of 32 runs.

Note that the algorithm is able to estimate the trajectories with reasonable accuracy, and is numerically stable, even after a large number of objects has passed.

To perform the coordinate transformation, the first two trajectories were fixed. The first car entered at 0 and exited at 0, while the second car entered at 0 and exited at 8. This improved the accuracy of the estimation.

VII. Concluding remarks

We have studied the application of directional sensors to the detection and tracking of moving objects. The estimation of the object trajectory is a non-convex optimization problem. To overcome the difficulty of local minima, as well as the large number of parameters, a two phase, adaptive-basis optimization algorithm was designed.

The use of the adaptive-basis algorithm in the first phase of the algorithm is of key importance, since large errors in the estimated sensor directions, produced by the use of a suboptimal method in the first phase, cannot be corrected by the algorithm in the second phase.

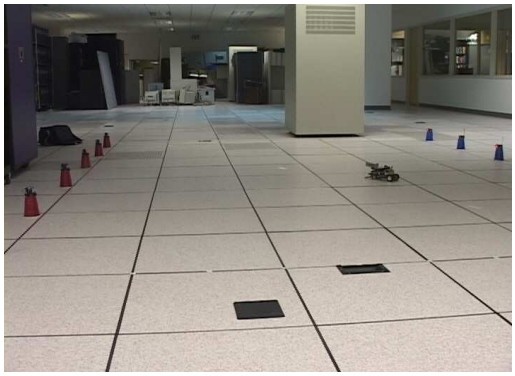

Fig. 7. Picture of testbed. Sensors can be seen on the right, while lasers are on the left. The car that was used as "object" is in the center.

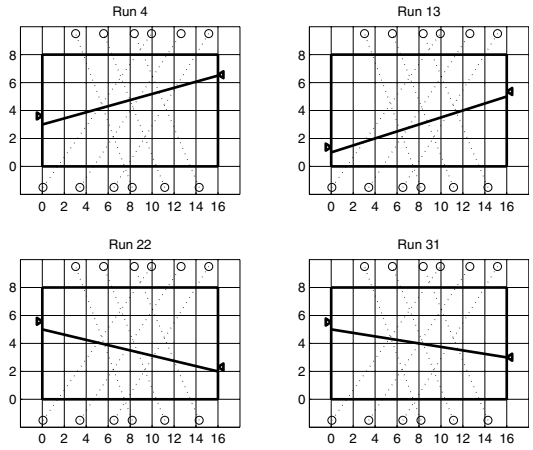

Fig. 8. Runs 4, 13, 22, and 31 from an experiment with a total of 32 runs. Top circles are lasers, bottom circles are sensors. Sensor lines are shown with dotted lines. Note that the sensor lines shown were estimated from the data. The domain is a rectangle marked with a thick borderline. The actual trajectory is shown as a left-to-right thick line. Estimated entry and exit points are indicated with triangles.

We have presented simulations comparing the performance of the adaptive-basis algorithm to that of a randomly initialized local improvement algorithm.

We have also presented the results of an actual implementation using Berkeley motes and lasers.

VIII. Acknowledgments

We thank Roberto Solis Robles for his implementation of this algorithm on the mica2 motes, and for ensuring the proper time synchronization of the sensors.

References

[1] Kurt Plarre and P. R. Kumar, "Increasingly correct message passing algorithms for heat source detection in sensor networks," in *1st IEEE Int. Conf. on Sensor and Ad hoc Communications and Networks (SECON 2004)*, Santa Clara, CA, October 2004, pp. 470–479.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazone*, vol. 40, no. 8, pp. 102–114, 2002.

[3] Juan Liu, Maurice Chu, Jie Liu, Jim Reich, and Feng Zhao, "Distributed state representation for tracking problems in sensor networks," in *3rd Int. Conf. on Information Processing in Sensor Networks (IPSN 2004)*, Berkeley, CA, April 2004, pp. 234–242.

[4] A Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem, "Distributed online localization in sensor-networks using a moving target," in *3rd Int. Conf. on Information Processing in Sensor Networks (IPSN 2004)*, Berkeley, CA, April 2004, pp. 61–70.

[5] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *3rd Int. Conf. on Information Processing in Sensor Networks (IPSN 2004)*, Berkeley, CA, April 2004, pp. 20–27.

[6] Dimitri P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, Mass., USA, 1995.

[7] Online reference: http://www.xbow.com.