Poster Abstract: Anchor-Free Distributed Localization in Sensor Networks

Nissanka B. Priyantha, Hari Balakrishnan, Erik Demaine, and Seth Teller
MIT Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
{bodhi, hari}@csail.mit.edu, edemaine@mit.edu, seth@csail.mit.edu

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems; F.2.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*Nonnumerical Algorithms and Problems*

General Terms

Algorithms, Performance.

Keywords

Sensor networks, Localization.

1. INTRODUCTION

Physical location is an important attribute of a sensor's data stream in a large number of sensor network applications. In addition, geographic information, for instance in the form of node coordinates in some common coordinate system, is a useful primitive in routing protocols such as geographic routing, information dissemination protocols such as directed diffusion using location attributes, and sensor query processing systems.

We present a method to facilitate large-scale deployment of location-aware sensor networks. We show that large networks of location-aware sensors can be made cooperatively self-configuring, that is, that each sensor can run an algorithm locally, interacting only with neighboring nodes, such that after a number of iterations all sensors will have reached a consensus about their coordinates in some coordinate system. By doing this in an automated manner, large-scale sensor networks can eliminate the cumbersome and unscalable process of manually configuring sensor nodes with their location.

In non-urban outdoor settings, nodes may obtain location information using an existing infrastructure such as GPS. However, GPS receivers may be too expensive, too large, too power-intensive for the desired application, or simply unavailable. One solution to this problem is an alternative location infrastructure such as Cricket that works in places that GPS does not. Another solution to these problems is to equip sensors with hardware capable of estimating distances to nearby nodes, and to have the sensors themselves self-configure into a consistent coordinate system.

Cricket-like infrastructures and autonomously operating sensor networks both need a solution to the following problem: *Given a set of nodes with unknown position coordinates, and a mechanism by which a node can estimate its distance to a few nearby (neighbor) nodes, determine the position coordinates of every node via local node-to-node communication.* Our solution to this problem is fully decentralized: all nodes start from a random initial coordinate assignment and use only local distance estimates to converge to a coordinate assignment that is consistent with the distance estimates by exchanging only local information. The resulting coordinate assignment has translation and orientation degrees of freedom, but is correctly scaled. A post-process could incorporate absolute position information into three or four nodes to remove the translation and orientation degrees of freedom.

2. POSSIBLE APPROACHES

Anchor-based v. anchor-free algorithms. Some previous work on this problem assumes that a non-negligible fraction of nodes in the network are *anchor nodes* that already know their location [1, 2, 4, 5]. In contrast, we pursued an anchor-free approach for three reasons. First, establishing anchors is a manual deployment task, and may be cumbersome. Second, the numerical stability of anchor-based approaches is questionable, since they give more weight to anchor position estimates, and errors in those estimates will have undue effect on the global solution. Finally, anchor-based approaches may not scale well, since to combat the instability described above, a large number of anchors may be required to configure an unbounded working area.

In contrast, anchor-free algorithms use local distance information to attempt to determine node coordinates when no nodes have pre-configured positions. Of course, any such coordinate system will not be unique and can be embedded into another global coordinate space in infinitely many ways, depending on global translation, rotation, and possibly flipping. This limitation is fundamental to the problem specification, and is not a limitation of the algorithm.

Incremental v. concurrent algorithms. These algorithms usually start with a core of three or four nodes with assigned coordinates. Then they repeatedly add appropriate nodes to this set by calculating the node's coordinates using the measured distances to previous nodes with already computed coordinates. These coordinate calculations are based on either simple trigonometric equations or some local optimization scheme.

SenSys'03, November 5–7, 2003, Los Angeles, California, USA.
ACM 1-58113-707-9/03/0011.

A drawback of incremental algorithms is that they propagate measurement errors, resulting in poor overall coordinate assignments. Some incremental approaches apply a later global optimization phase to balance such error, but it remains difficult to jump out of local minima introduced by the local optimization in the incremental phase.

Concurrent algorithms. In these algorithms, all the nodes calculate and refine their coordinate information in parallel. Some of these algorithms use an iterative optimization scheme that reduces the difference between measured distances and the calculated distances based on current coordinate estimates.

3. AFL: ANCHOR-FREE LOCALIZATION

Our contribution is an algorithm called *AFL* (Anchor-Free Localization), a *concurrent* and *anchor-free* solution to the above problem. We show that this combination has significant advantages over several previous approaches. The AFL algorithm consists of two phases. The first phase produces a fold-free graph embedding which "looks similar" to the original embedding. The second phase uses a mass-spring based optimization to correct and balance localized errors.

The first phase of the algorithm proceeds by electing five reference nodes. The algorithm first elects five reference nodes. First it selects a node n_1 at the periphary of the graph. Next it selects node n_2 which is maximum hop count away from n_1. Next, the node at a maximum hop count away and at eqidistant, from both n_1 and n_2 is selcted as n_3. Node n_4 is the node at maximum hopcount from n_3 and at equidistant from n_1 and n_2. Finally node n_5 is selected to be the one at eqidistant from all n_1, n_2, n_3, and n_4. All these nodes are selected using a straightforward variant of distributed leader election.

Then, for each node n_i, use the hop-counts $h_{1,i}$, $h_{2,i}$, $h_{3,i}$, $h_{4,i}$,and $h_{5,i}$ from the chosen reference nodes to approximate the polar coordinates (ρ_i, θ_i). Here, R is the maximum radio range.

$$\begin{aligned} \rho_i &= h_{5,i} \times R \\ \theta_i &= \tan^{-1}\left(\frac{h_{1,i} - h_{2,i}}{h_{3,i} - h_{4,i}}\right) \end{aligned}$$

The second phase of the AFL algorithm performs a local optimization of the current estimated coordinates of each node in parallel. Using the current estimated position, each node n_i calculates the estimated distance $\hat{d}_{i,j}$ to each neighbor n_j. It also knows the *measured* distance $r_{i,j}$ to each neighbor n_j.

Let $\hat{v}_{i,j}$ represent the unit vector in the direction from $\hat{p}_i$ to $\hat{p}_j$. The error between the estimated and the measured distances is represented by a force $\vec{F}_{i,j}$ in the direction $\hat{v}_{i,j}$. This force is defined as

$$\vec{F}_{i,j} = \hat{v}_{i,j}(\hat{d}_{i,j} - r_{i,j}). \tag{1}$$

The resultant force on the node i is given by

$$\vec{F}_i = \sum_{i,j} \vec{F}_{i,j}.$$

Each node i update its estimated coordinates by "moving" in the direction of the resultant force. The new estimated coordinates are selected such that it reduces the energy of the node and the movement is less than $|\vec{F}_i|/(2m_i)$, where m_i is the number number of neighbors.

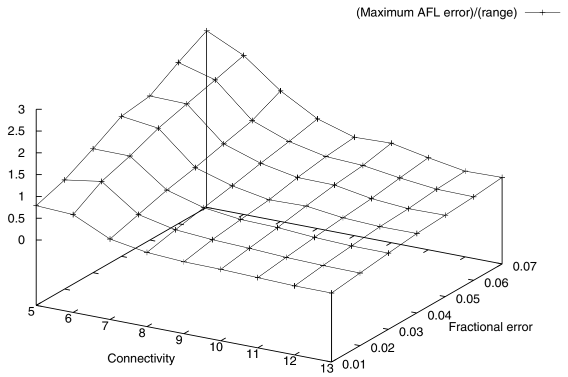

Figure 1: Maximum error between any two unconnected nodes as a fraction of the range.

We simulated the AFL algorithm on a 250-node graph and we ran 50 simulations to obtain each point on the graph. More extensive simulation results are described in [3].

Figure 1 shows the *maximum* error between any two nodes after running the AFL algorithm on a 250-node graph. Each point on the grap represents 50 simulations on different graphs. When the graph undergoes some physical deformation, this is identical to some points in the graph moving with respect to other points. Hence the maximum error between any two points corresponds to the maximum deformation the graph has undergone. Figure 1 shows the superior performance of AFL under ranging errors, since the maximum distance error between any two points is small most of the time. In most cases the absolute position error is smaller than the radio range, showing a degree of robustness to error that is significantly better than in previously published schemes.

4. REFERENCES

[1] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less Low Cost Outdoor Localization For Very Small Devices. Technical Report 00-729, Computer Science Department, University of Southern California, Apr. 2000.

[2] L. Doherty, K. Pister, and L. Ghaoui. Convex position estimation in wireless sensor networks. In *Proc. IEEE INFOCOM*, April 2001.

[3] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Anchor-Free Distributed Localization in Sensor Networks. Technical Report MIT-LCS-TR-892, MIT Lab for Computer Science, Apr. 2003.

[4] C. Savarese, J. Rabaey, and J. Beutel. Locationing in Distributed Ad-Hoc Wireless Sensor Networks. In *Proc. of ICASSP*, pages 2037–2040, Salt Lake City, UT, May 2001.

[5] A. Savvides, C. Han, and M. Srivastava. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In *Proc. 7th ACM MOBICOM Conf.*, pages 166–179, July 2001.