

# Indoor People Tracking Based on Dynamic Weighted MultiDimensional Scaling\*

Jose Maria Cabero  
Robotiker-Tecnalia Technology Centre  
Zamudio, Bizkaia, 48170  
Spain  
jmcabero@robotiker.es

Aritz Sanchez  
Robotiker-Tecnalia Technology Centre  
Zamudio, Bizkaia, 48170  
Spain  
aslekue@robotiker.es

Fernando De la Torre  
Carnegie Mellon University  
Pittsburgh, PA 15232  
USA  
ftorre@cs.cmu.edu

Iñigo Arizaga  
Robotiker-Tecnalia Technology Centre  
Zamudio, Bizkaia, 48170  
Spain  
inyigo@robotiker.es

## ABSTRACT

Accurate location of people in indoor environments is a key aspect of many applications such as resource management or security. In this paper, we explore the use of short-range radio technologies to track people indoors. The network consists of two kind of radio nodes: static nodes (anchors) and mobile nodes (people). From a set of sparse connectivity matrices (people vs. people and people vs. anchors) at each time instant and people's dynamics, we infer people's trajectories. To combine connectivity and dynamic information, we propose an extension of Multidimensional Scaling (MDS), Dynamic Weighted MDS (DWMDs), that finds an embedding of people's trajectories (x and y coordinates of people through time). DWMDs has proven to be more accurate and effective, especially for low connectivity degree networks (i.e. sparse networks), compared to existing location algorithms. Extensive simulations show the effectiveness and robustness of the proposed algorithm.

## Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Wireless communication*; G.1.6 [Numerical Analysis]: Gradient methods

## General Terms

Algorithm, Performance

\*The Project is funded by the AmIGUNE Programme (<http://www.amigune.org>). AmIGUNE is a long-term collaborative Research Programme among the leading Technology Centres and Universities of the Basque Country, with Robotiker-Tecnalia acting as Coordinator.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MSWIM '07, October 22–26, 2007, Chania, Crete Island, Greece.  
Copyright 2007 ACM 978-1-59593-851-0/07/0010 ...\$5.00.

## Keywords

Indoor Tracking, People's Dynamics, RSS, Connectivity Data, Short-Range Radio Technologies

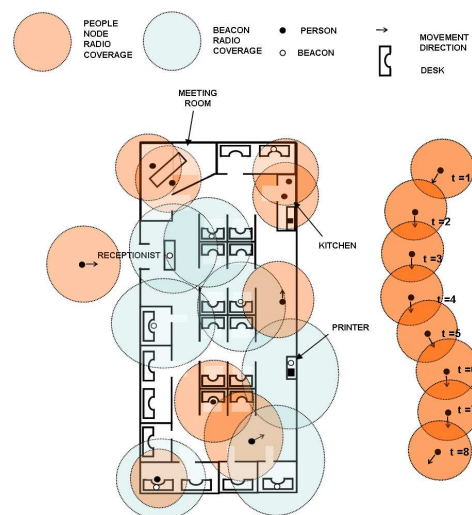


Figure 1: Left: tracking scenario in an office environment at one time snapshot. Right: trajectory of one person through time

## 1. INTRODUCTION

Tracking and locating people indoors and outdoors is an important problem in many applications, such as security, monitoring elderly people, or resource management. Outdoor tracking can be successfully solved with location technologies such as GPS or types of radio communication, such as 802.11 [9, 2]. However, indoor applications with tracking requirements are still an open research topic, since GPS cannot work indoors and radio communication technologies, mainly based on distance estimations, are prone to high errors because of the complexity of the indoor radio channel.

Short-range communication technologies, due to their cheap cost, are being implemented in almost all personal electronic devices, such as mobile phones, PCs or PDAs. The

widespread use of these devices makes them ideal platforms for location-aware applications. In this paper, we investigate a new indoor tracking system based on short-range radio technologies. Figure 1 left, shows an office environment scenario with two kinds of radio nodes: static nodes (anchors) and mobile nodes (people). Given the set of connectivity matrices between people vs. people and people vs. anchors at different time stamps, an extension of Multidimensional Scaling (MDS) [4], Dynamic Weighted MDS (DWMDS), is proposed to infer the unknown people’s trajectories (see figure 1 right). To resolve ambiguities that occur when having a very sparse set of measurements, our approach finds solutions consistent with a previously learned mobility pattern of people in the network.

The rest of the paper is organized as follows: section 2 reviews previous work and section 3 formulates DWMDS, the tracking algorithm. The optimization procedure is explained in section 4 and the corresponding experiments and comparison with other location algorithms are reported in section 5. Section 6 presents the conclusions and future research.

## 2. PREVIOUS WORK

Most popular methods to locate people indoors are based on measurements of radio signals, such as Time of Arrival (ToA) [15], Angle of Arrival (AoA) [10], Time Difference of Arrival (TDoA) [13] and Received Signal Strength (RSS). The first three measurements need costly customized hardware, whereas RSS is the most attractive technique because of the variety of personal radio communication devices that cheaply and by default implement it.

Most of the algorithms based on RSS use trilateration [1], multilateration [11] or similar methods [19, 8] to make an estimation of the distance between the tracked object and some known anchors. However, the complexity of the radio channel indoors and, consequently, its difficulty to be properly modeled, makes these methods very prone to errors [5]. Most reliable RSS-based methods do not try to estimate distances directly from RSS, instead a RSS map of the scene is previously built [12, 8, 3, 1]. However, this methodology is extremely dependent on the environment and any significant change to the topology implies a costly re-calibration.

In the context of sensor networks, classical MDS has been used to locate static sensor nodes in dense wireless sensor networks [7, 18, 17, 16]. All of them are applied to static networks with high connectivity. When the networks are sparser, the errors increase exponentially. The best results are reported by [16] using classical MDS (MDS-MAP(P)) as a starting point and an optional subsequent optimization phase based on least squares minimization (MDS-MAP(P,R)). DWMDS performance is compared to MDS-MAP(P) and MDS-MAP(P,R) in section 5.3.

## 3. TRACKING AS A LOW DIMENSIONAL EMBEDDING PROBLEM

In this section we pose the problem of tracking people as finding a two dimensional embedding using the connectivity information of people and anchors. Section 3.1 discusses the benefits of using connectivity rather than distance estimations. Section 3.2 describes the energy functions of the extended MDS algorithm.

### 3.1 Connectivity versus distance

Most of the current localization techniques using RSS are based on distance estimations between the tracked node and anchors with known positions, using the free space attenuation formula [14]. Nevertheless, such a straightforward approach does not account for the extremely complex nature of the electromagnetic field indoors, which mainly present three physical effects: reflection, refraction and diffraction. These effects make the simple characterization of the wireless channel indoors a weak model for practical applications, which leads to big errors in the estimation of distances (usually, the bigger the radius of coverage is (R), the bigger the expected error in the estimation of the distance will be).

The inability to accurately model the radio channel indoors requires looking for new approaches that do not rely on distance estimations. Our tracking algorithm relies on connectivity information, short-range technologies and people’s dynamics. The connectivity information at one time instant between two nodes in the network is recorded according to the following procedure (see figure 2): if a node  $i$  is inside the coverage radius of a node  $j$  (and vice versa), the  $ij$  and  $ji$  terms of a connectivity matrix are set to 1 (connected nodes) and 0 otherwise (disconnected nodes).

$i$  and  $j$  can be any node, anchor or person. As section 5 will show, not only do the anchors work actively in the tracking process, but the nodes in motion help too. For instance, considering figure 2: if node A were an anchor and node B and C nodes in motion that at time  $t$  are at those positions, then even though there is not a direct connection between nodes A and C, the node B acts as a bridge between them, and our tracking system will take advantage of it.

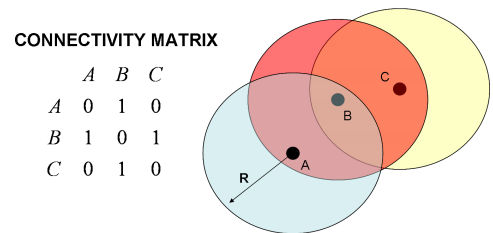


Figure 2: Node B detects A and C. C and A are out of range, so they do not detect each other

### 3.2 Dynamic Weighted MultiDimensional Scaling (DWMDS)

Multidimensional scaling (MDS) [4] is a powerful statistical dimensionality reduction technique for data analysis that has been used extensively in social sciences, engineering or marketing. The starting point of MDS is a matrix consisting of pairwise dissimilarities or distances between data samples. MDS attempts to find an embedding in a metric space, so that the distances in this low-dimensional space correspond to the given dissimilarities between samples in the original space. Let  $\mathbf{y}_1, \dots, \mathbf{y}_n$  be the samples in the original space, and  $\delta_{ij}$  the corresponding distance between sample  $i$  and  $j$ . Let  $\mathbf{x}_1, \dots, \mathbf{x}_n$  be the coordinates of the samples in the embedded space and  $d_{ij}$  the corresponding distance between sample  $i$  and  $j$  in that space. The main goal of MDS is to find an embedding (i.e.  $\mathbf{x}_1, \dots, \mathbf{x}_n$ ) such that  $d_{ij}$  in the low-dimensional space is close to the original distance in the original space  $\delta_{ij}$  in the least square sense. It is not usu-

ally possible that  $d_{ij} = \delta_{ij} \forall i, j$ , and it is common to find a unique solution by averaging the least square error using different normalization errors such as:<sup>1</sup>

$$\Omega_1(\mathbf{X}) = \sum_{i < j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}} \quad \Omega_2(\mathbf{X}) = \frac{\sum_{i < j} (d_{ij} - \delta_{ij})^2}{\sum_{i < j} \delta_{ij}^2}$$

$$\Omega_3(\mathbf{X}) = \sum_{i < j} \left( \frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2$$

A local minimum of the previous error functions w.r.t  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$  is usually found by using standard gradient techniques [6]. In this section, we extend classical MDS by adding dynamic information to constrain people's trajectories. Let us denote the coordinates of the nodes (people and anchors) in the network as:

$$\mathbf{X}^{d,t} = \begin{pmatrix} \mathbf{x}_1^{d,t} \\ \mathbf{x}_2^{d,t} \\ \vdots \\ \mathbf{x}_r^{d,t} \end{pmatrix} \quad \mathbf{X}^s = \begin{pmatrix} \mathbf{x}_1^s \\ \mathbf{x}_2^s \\ \vdots \\ \mathbf{x}_q^s \end{pmatrix}$$

$$\mathbf{x}_i^{d,t} = \{x_{i1}^{d,t}, x_{i2}^{d,t}\} \quad \begin{cases} \mathbf{x}_i^{d,t} \in \mathbb{R}^{1 \times 2} \\ \mathbf{x}_i^s \in \mathbb{R}^{1 \times 2} \\ \mathbf{X}^{d,t} \in \mathbb{R}^{r \times 2} \\ \mathbf{X}^s \in \mathbb{R}^{q \times 2} \end{cases}$$

where  $\mathbf{X}^{d,t}$  corresponds to the coordinates of the mobile nodes at time stamp  $t$ .  $\mathbf{X}^s$  are the time invariant coordinates of the static nodes.  $x_{i1}$  and  $x_{i2}$  are the  $x$  and  $y$  coordinates for the device  $i$ . Assuming we gather data over  $T$  time stamps:

$$\mathbf{X}^d = ( \mathbf{X}^{d,1} \quad \mathbf{X}^{d,2} \quad \dots \quad \mathbf{X}^{d,T} ) \in \mathbb{R}^{r \times 2T}$$

The input to the proposed DWMDs algorithm is a set of binary matrices with connectivity information between all nodes in the network, one at each time stamp. To avoid ill posed problems and recover smooth trajectories, we add a smoothing dynamic penalty term that penalizes the movement of mobile nodes in two consecutive time stamps, according to their dynamics. The DWMDs expression minimizes:

$$\Omega_{DWMDs}(\mathbf{X}^{ALL}) = \sum_{t=1}^T \sum_{i < j} \frac{w_{ij}^t (d_{ij}^t - \delta_{ij}^t)^2}{\delta_{ij}^t} \quad (1)$$

$$+ \sum_{t=2}^T \sum_{i=1}^r \left\| \alpha_i \left( \mathbf{x}_i^{d,t} - \lambda_i^{d,t-1} \cdot \mathbf{x}_i^{d,t-1} \right) \right\|_F^2$$

where at any time stamp

$$\sum_{i < j} \frac{w_{ij} (d_{ij} - \delta_{ij})^2}{\delta_{ij}} = \sum_{i < j} \frac{w_{ij}^{dd} (d_{ij}^{dd} - \delta_{ij}^{dd})^2}{\delta_{ij}^{dd}}$$

$$+ \sum_{i < j} \frac{w_{ij}^{sd} (d_{ij}^{sd} - \delta_{ij}^{sd})^2}{\delta_{ij}^{sd}} + \sum_{i < j} \frac{w_{ij}^{ss} (d_{ij}^{ss} - \delta_{ij}^{ss})^2}{\delta_{ij}^{ss}} \quad \delta_{ij} \in (0, R]$$

$\mathbf{X}^{ALL}$  comprises the coordinates of the trajectories  $\mathbf{X}^d$  and the position of the anchors  $\mathbf{X}^s$ .  $w_{ij}^t$  is a weight that stresses the difference between the distance of the nodes  $i$  and  $j$  ( $d_{ij}^t$ , which depends on  $\mathbf{x}_i$  and  $\mathbf{x}_j$ ) and the corresponding connectivity value  $\delta_{ij}^t$  at time  $t$ .  $\delta_{ij}^t$  can take any value from zero to  $R$  (coverage radius) when there is connectivity; otherwise, this term is not taken into account in the error equation. The  $\alpha_i$  is a tradeoff parameter to equilibrate the contribution of the static and dynamic terms.  $\lambda_i^{d,t-1}$  is a vector to adjust each tracked person's trajectory dependent on their dynamics. This article works with synthetic trajectories, so we assume that the dynamics of each mobile node is known. In real testbeds, there will be a learning stage where the dynamics of each person in the network is modeled. Before proceeding with details of the algorithm, we reformulate the problem into a matrix formulation for mathematical convenience. Equation 1 can be rewritten as:

$$\Omega_{DWMDs}(\mathbf{X}^{ALL}) = \quad (2)$$

$$+ \left\| \mathbf{M}^B \circ (\mathbf{D}^B (\mathbf{X}^{ALL}) - \Delta^B) \right\|_F^2$$

$$+ \left\| \mathbf{A} (\mathbf{X}^d - (\mathbf{X}^d \circ \mathbf{\Lambda}) \mathbf{P}) \mathbf{I}_{mT_{first}} \right\|_F^2$$

where, for simplicity in the rest of the article, the first term comprises the three terms showed above regarding the mobile vs. mobile nodes, static vs. mobile nodes and static vs. static nodes relations. In the static term:

$$\mathbf{M}^B = \frac{\mathbf{T}^B \circ (\mathbf{W}^B)^{\circ \frac{1}{2}}}{(\Delta^B)^{\circ \frac{1}{2}}} \in \mathbb{R}^{(r+q)T \times (r+q)T}$$

<sup>1</sup>Bold non-capital letters are used to denote vectors. Bold capital letters are used to denote matrices. All non-bold letters will represent variables of scalar nature.  $d_{ij}$  denotes the scalar in the row  $i$  and column  $j$  of the matrix  $\mathbf{D}$ . The number of mobile devices is  $r$ , the number of static devices is  $q$ ,  $T$  is the number of time stamps. The superindexes  $s$  and  $d$  correspond to the static and mobile nodes respectively.  $sd$  and  $dd$  represent the cross terms static vs. mobile nodes, and mobile vs. mobile nodes respectively.  $\|\mathbf{A}\|_F^2 = \text{tr}(\mathbf{A}^T \mathbf{A}) = \text{tr}(\mathbf{A} \mathbf{A}^T)$  designates the squared Frobenius norm of a matrix  $\mathbf{A}$ . Block diagonal matrices will have the superindex  $B$ , and consist of as many matrices as time stamps.

$$\mathbf{\Gamma}^B = \text{diag} ( \mathbf{\Gamma}^1, \quad \mathbf{\Gamma}^2, \quad \dots \quad \mathbf{\Gamma}^T ) \in \mathbb{R}^{(r+q)T \times (r+q)T}$$

Where  $\mathbf{\Gamma}^t \in \mathbb{R}^{(r+q) \times (r+q)}$  will be  $\mathbf{W}^t$ ,  $\mathbf{D}^t$ ,  $\Delta^t$ ,  $\mathbf{L}^t$ ,  $\mathbf{T}^t$ ,  $\mathbf{N}^t$ ,  $\mathbf{M}^t$ ,  $\mathbf{Z}^t$  in the time stamp  $t$  along the article.

$\mathbf{T}^B$  is an upper matrix of ones,  $\mathbf{W}^B$  is the weight matrix and  $\Delta^B$  is the connectivity matrix.  $\mathbf{D}(\mathbf{X}^{ALL}) \in \mathbb{R}^{(r+q)T \times (r+q)T}$  is the distance matrix dependent on the coordinates of the nodes. In the following developments, the  $\mathbf{X}^{ALL}$  dependency is not shown anymore.

In the dynamic term,  $\mathbf{P} \in \mathbb{R}^{2T \times 2T}$  is the permutation matrix that delays the coordinate matrix one time stamp.  $\mathbf{\Lambda} \in \mathbb{R}^{r \times 2T}$  is the corresponding matrix with the  $\lambda_i^d$ .  $\mathbf{A} \in \mathbb{R}^{r \times r}$  is the diagonal matrix with the tradeoff parameters  $\alpha_i$  in the diagonal.  $\mathbf{I}_{2T_{first}} \in \mathbb{R}^{2T \times 2T}$  is the identity matrix with the first 2 columns set to zero.

Connectivity information is measured according to a metric called connectivity degree that accounts for the average number of connections per node, offering a quick view of the density of the network. As our algorithm introduces

a dynamic term that links different time stamps, the connectivity degree in this paper does not measure only the average connectivity at a time instant, but over the whole set of time stamps. Furthermore, we differentiate four different connectivity degrees: CD\_MM, CD\_MS, CD\_Mobile and CD\_Total corresponding to the connectivity of mobile vs. mobile nodes, mobile nodes vs. anchors, mobile vs. all nodes and all vs. all nodes respectively. Equation 3 shows how to calculate the connectivity degree of a network, where  $\gamma_{ij}$  is 1 if nodes  $i$  and  $j$  are connected, 0 otherwise. The subscripts  $t$ ,  $s$  and  $d$  refers to the time instant, and the type or node, static or mobile respectively.

$$\begin{aligned} CD\_MM &= \sum_{t=1}^T \sum_{i=1}^r \sum_{j=1}^r \frac{\gamma_{ij}^{dd^t}}{rT} \\ CD\_MS &= \sum_{t=1}^T \sum_{i=1}^r \sum_{j=1}^q \frac{\gamma_{ij}^{sd^t}}{rT} \\ CD\_Mobile &= CD\_MM + CD\_MS \\ CD\_Total &= \sum_{t=1}^T \sum_{i=1}^{r+q} \sum_{j=1}^{r+q} \frac{\gamma_{ij}^t}{(r+q)T} \end{aligned} \quad (3)$$

#### 4. OPTIMIZATION

In this section we derive a gradient-based algorithm to find the optimal solution of equation 2 w.r.t.  $\mathbf{X}_n^{ALL}$ , which is  $\mathbf{X}^{ALL}$  at iteration  $n$ . In this paper, we assume that  $\mathbf{X}^s$  is known and we do not update it, but we derive the updates in the interest of generalization. The gradient updates are given by:

$$\mathbf{X}_{n+1}^{ALL} = \mathbf{X}_n^{ALL} - \eta \frac{\partial \Omega_{DWMDS}}{\partial \mathbf{X}_n^{ALL}} \quad (4)$$

Where  $\frac{\partial \Omega_{DWMDS}}{\partial \mathbf{X}_n^{ALL}}$  is a unit vector in the direction of the gradient. One major problem with the update of equation 4 is to determine the optimal  $\eta$ . In our case  $\eta$  is determined with a line-search strategy [6].

$$\begin{aligned} \frac{\partial \Omega_{DWMDS}}{\partial \mathbf{X}^{ALL}} &= 2\mathbf{X}^B \circ (\text{diag}(\text{diag}(\mathbf{N}^B(\mathbf{N}^B)^T)) \mathbf{1}_{(r+q)T2T}^B) \\ &\quad - 2(\mathbf{N}^B)^{\circ 2} \mathbf{X}^B - 2(\mathbf{X}^B \circ (\mathbf{L}^B \mathbf{1}_{(r+q)T2T}^B) - \mathbf{L}^B \mathbf{X}^B) \end{aligned}$$

$$+ 2\mathbf{A}\mathbf{X}^d \mathbf{I}_{2T_{first}} - 2\mathbf{A}(\mathbf{A} \circ \mathbf{X}^d) \mathbf{P} \mathbf{I}_{2T_{first}}$$

$$- 2\mathbf{A} \circ (\mathbf{A}\mathbf{X}^d \mathbf{I}_{2T_{first}} \mathbf{P}^T) + 2\mathbf{A} \circ (\mathbf{A}(\mathbf{A} \circ \mathbf{X}^d) \mathbf{P} \mathbf{I}_{2T_{first}} \mathbf{P}^T)$$

where

$$\begin{aligned} \mathbf{N}^B &= \mathbf{M}^B + (\mathbf{M}^B)^T; & \mathbf{L}^B &= \mathbf{Z}^B + (\mathbf{Z}^B)^T; \\ \mathbf{Z}^B &= \frac{\mathbf{T}^B \circ \mathbf{W}^B}{\mathbf{D}^B}; & \mathbf{M}^B &= \frac{\mathbf{T}^B \circ (\mathbf{W}^B)^{\circ \frac{1}{2}}}{(\mathbf{\Delta}^B)^{\circ \frac{1}{2}}} \end{aligned}$$

$$\mathbf{X}^B = \text{diag}(\mathbf{X}^{ALL,1}, \mathbf{X}^{ALL,2}, \dots, \mathbf{X}^{ALL,T})$$

$$\mathbf{X}^{ALL,t} = \begin{pmatrix} \mathbf{X}^{d,t} \\ \mathbf{X}^s \end{pmatrix} \in \mathfrak{R}^{(r+q) \times 2}$$

$$\mathbf{X}^B \in \mathfrak{R}^{(r+q)T \times 2T}$$

$\text{diag}(\text{diag}(\mathbf{N}^B(\mathbf{N}^B)^T))$  is a diagonal matrix whose diagonal is the diagonal of  $\mathbf{N}^B(\mathbf{N}^B)^T$ .  $\mathbf{1}_{(r+q)T2T}^B$  is a block diagonal matrix with  $T$  ones matrices of dimensions  $(r+q)2$  along the diagonal.

The initialization of the algorithm is random, although any a priori information regarding mobile nodes' positions, could be used to achieve a faster convergence. The computational cost of the algorithm is  $O(n^3T)$  for the MDS term and  $O(r^2T)$  for the dynamic term, where  $n$  is the total number of nodes,  $r$  is the number of mobile nodes and  $T$  is the tracking time (number of time instants).

#### 5. PERFORMANCE EVALUATION AND COMPARISON

In this section, we report extensive simulation results of the performance of DWMDS in two scenarios (see figure 3): one with 16 anchors and 4 mobile nodes (scenario 1 of dimensions  $4R \times 4R$ ), and the other one with 100 anchors and 70 mobile nodes (scenario 2 of dimensions  $10R \times 10R$ ). Anchors in both scenarios are originally distributed in a uniform grid, but successively the network will become non-uniform and very sparse in the experiments.

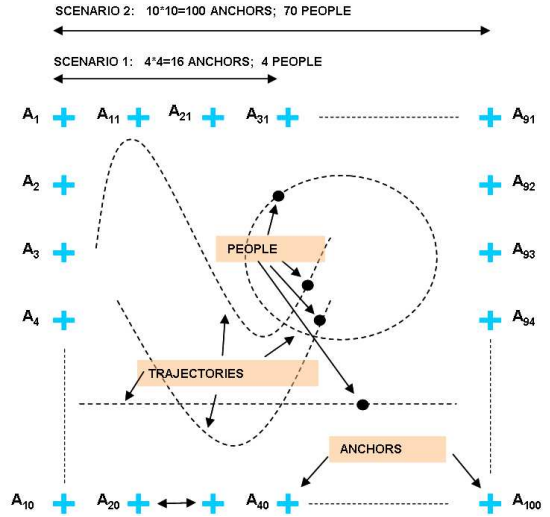


Figure 3: Ground truth for scenarios 1 and 2

Each mobile node follows one of four possible trajectories of duration  $T = 40$  time stamps (this duration is enough to test the effect of the dynamic term. Longer times provide similar accuracy), starting in different positions of the simulation area and with different speeds. The parameters used in the simulations are shown in table 1. Although random trajectories can be used with the same error performance, we have used circular, sinusoidal, convex and linear trajectories to analyze the resultant trajectories visually (examples of the four trajectories are shown in figure 3). The coverage radius is assumed the same for all the nodes. The connectivity value  $\delta_{ij}$  is  $\frac{\text{coverage radius}}{2}$  for all the nodes connected. The position of the anchors is known and the goal is to estimate the mobile trajectories. The reconstruction error in the following sections is the Root Mean Square (RMS) of the difference between the result of the algorithm and the ground truth, unless otherwise stated. The coverage radius

Scenario 1 Area: 4R x 4R	16 Anchors and 4 people
Scenario 2 Area: 10R x 10R	100 Anchors and 70 people
Simulation time (T)	40 Time stamps
$\delta_{ij}$	$\frac{\text{coverage radius}}{2}$
Trajectories	Circular $\frac{0.31R}{\text{timestamp}}$
	Sinusoidal $\frac{0.18R}{\text{timestamp}}$
	Convex $\frac{0.19R}{\text{timestamp}}$
	Linear $\left\{ \begin{array}{l} \frac{0.5R}{\text{timestamp}} \\ \frac{0.25R}{\text{timestamp}} \end{array} \right.$

**Table 1: Simulation parameters**

used in the simulations is  $R$ , unless otherwise stated. All the simulations are done with MATLAB 7.1 (R14SP3).

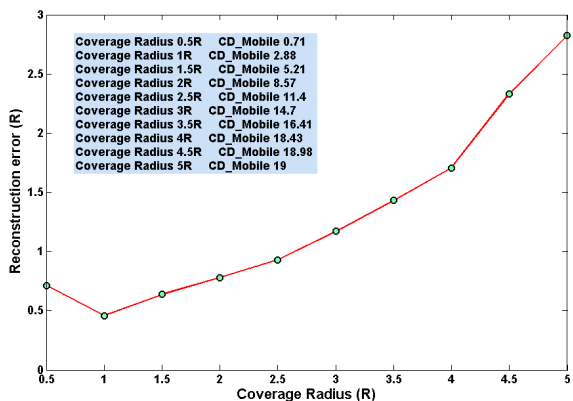
## 5.1 Scenario 1

In Scenario 1, we are interested in measuring the robustness of our approach w.r.t. the coverage radius of the nodes, noise in the anchors' position and missing connectivity data.

### 5.1.1 Coverage radius of the nodes

This section reports the behavior of the tracking algorithm when the coverage radius changes. Figure 4 shows the reconstruction error w.r.t.  $R$  versus the coverage radius. The radius with which the minimum reconstruction error is obtained is the "Optimum Radius". Any value bigger or smaller than the Optimum Radius leads to bigger errors and it is a parameter highly dependent on the network status as it would be explained in section 5.2.4

The legend in figure 4 shows the network connectivity degree when the radius changes. Unlike previous works [7, 18, 17, 16], DWMDs does not obtain better results when the connectivity degree increases, due to the increase of the coverage radius. The next sections will show that when the connectivity increases because of the increase of nodes in the network, keeping the coverage radius constant, then the reconstruction error tends to decrease.

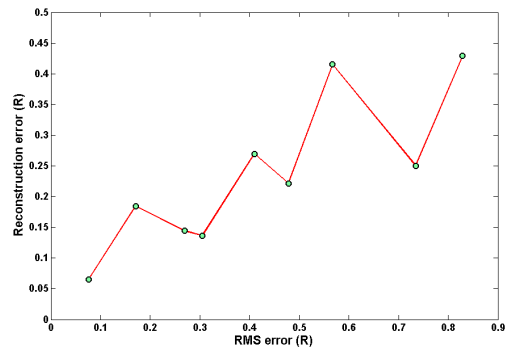


**Figure 4: Reconstruction error versus the coverage radius**

### 5.1.2 Anchors' noisy positions

The previous section assumes that the positions of the anchors are known. This section studies the stability of the solution when the position of the anchors is not accurately known. Figure 5 shows the reconstruction error w.r.t.  $R$  versus the random RMS error w.r.t.  $R$  introduced in the anchors' position. In this section, the reconstruction error is the difference between the result of the algorithm and the result of the algorithm without noise in the position of the anchors.

The results show that the reconstruction error increases slower than the error introduced in the position of the anchors. Sometimes due to the ambiguities of the problem, when the introduced noise increases the reconstruction error is smaller. This effect counteracts the effect between the intrinsic ambiguity of the algorithm and the error introduced in the anchors.



**Figure 5: Reconstruction error when the position of the anchors is noisy**

### 5.1.3 Noise in the connectivity matrices

This section tests the stability of our algorithm w.r.t. noise in the connectivity matrices. In particular, we will study the situation where some nodes are in range, but the system does not detect it. Figure 6 shows the reconstruction error versus CD\_Mobile. The smaller the CD\_Mobile, the more connectivity information is missing (2.88 is the CD\_Mobile when no data are missing).

Figure 6 shows smooth increases of the error up to CD\_Mobile around 1.5. With smaller values of connectivity the reconstruction error increases faster, but within reasonable values, even when CD\_Mobile is smaller than 1.

## 5.2 Scenario 2

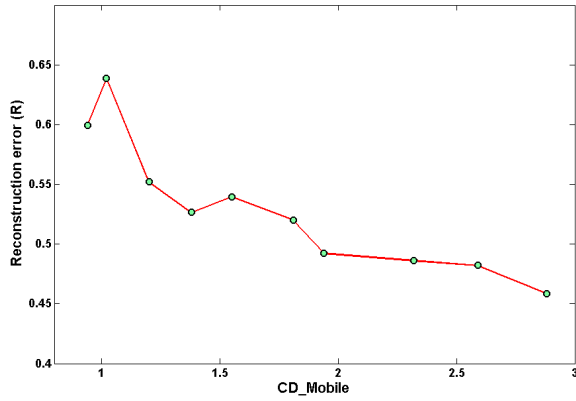
In scenario 2 we study the effect of reducing the connectivity degree by removing both anchors and mobile trajectories.

### 5.2.1 Removing inner anchors

In this section, we randomly remove inner anchors from the network and observe the impact on reconstruction error. The starting point of the simulation is 70 people and 100 anchors. The simulation runs 8 times, each of them with the same 70 trajectories and subtracting 8 anchors randomly chosen. The simulation finishes running the algorithm with 36 boundary anchors and 70 mobile trajectories.

The only anchors left at the end of the simulation are the ones on the boundary, since these anchors have a special im-



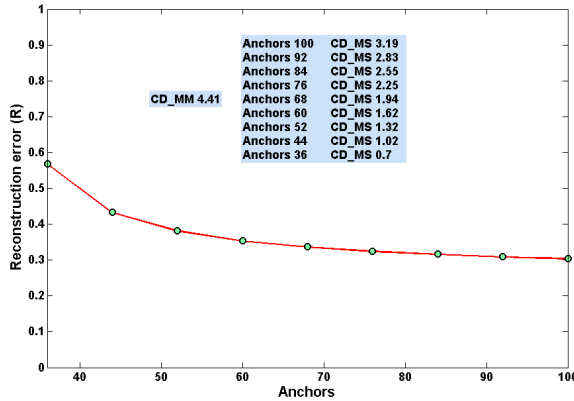


**Figure 6: Reconstruction error when there are connectivity data missing**

impact on the network. Some anchors should delimit the network, otherwise the trajectories obtained by the algorithm tend to be limited and distorted around the positions of the outer anchors, even if the trajectories go originally farther. This is what is called Shrinking effect.

*Definition 1. Shrinking effect* is the attraction that the boundary anchors have over people's trajectories in the surroundings, making them concentrate around them even when they go farther.

Figure 7 shows that the reconstruction error is very similar (around  $0.4R$ ) even when all the inner anchors are removed, which emphasizes the importance that mobile nodes have to get other mobile nodes tracked. Another interesting effect is the smoothness of the error, which smoothness increases with the number of anchors in the network.



**Figure 7: Reconstruction error when the number of inner anchors decreases**

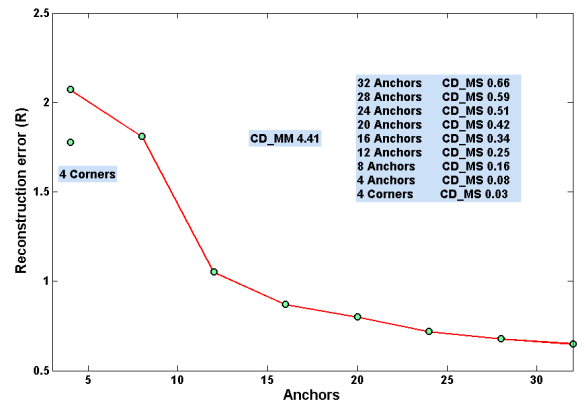
*Definition 2. Saturation point* is the number of anchors from which the accuracy of the tracking algorithm does not increase considerably on adding more anchors.

Figure 7 shows that the difference in the reconstruction error decreases until it becomes negligible when the number of

anchors is near 100, which could be considered as the Saturation point of this network. The Saturation point depends on aspects such as the structure of the network, the coverage radius, the strategic location of the anchors and the difference in the error allowed by the designer. Minimizing the Saturation point is a future research point beyond the scope of this article.

### 5.2.2 Removing boundary anchors

Once the impact of removing inner anchors has been reported in the previous section, here the analysis focuses on the impact of removing outer nodes when there are not inner ones. The starting point of the simulation is 36 boundary anchors and 70 mobile nodes. Progressively, one randomly-chosen node per side (4 anchors each time) is removed until reaching a minimum of 4 (figure 8). A special situation exists when the last four remaining anchors are the ones in the corners (isolated dot in figure 8).



**Figure 8: Reconstruction error when the number of anchors decreases**

The results show how the reconstruction error is around  $R$  when the number of anchors is decreased to 12 boundary nodes. The special situation with 4 nodes in the corners reports a smaller error compared to the situation where the 4 remaining nodes are randomly-chosen, since the former one delimits the network better.

### 5.2.3 Removing anchors and mobile nodes

Previous sections are focused on the impact of removing anchors. This section surveys the impact of removing anchors (decreasing the  $CD_{MS}$ ) and removing mobile nodes (decreasing the  $CD_{MM}$ ) at the same time.

The starting scenario consists of 36 boundary anchors and 70 mobile nodes' trajectories. The algorithm runs 90 times, each of them with a different number of anchors and people. Figure 9 shows the results when the algorithm converges. Each white cell shows the reconstruction error and  $CD_{MS}$  w.r.t. number of people (blue row in the main table) and anchors (blue column) used in the simulation. The small table on top of the main one shows  $CD_{MM}$ , which remains constant for each column.

Analyzing the table per columns, it shows how when the number of mobile nodes remains constant, the obtained trajectories error increases when  $CD_{MS}$  decreases. Analyzing the table per rows, it shows how when the number of anchors remains constant, the obtained trajectories error increases

CD_MM	4.41	3.83	3.38	2.89	2.43	1.94	1.47	1.02	0.74	0.11
PEOPLE ANCHORS	70	63	56	49	42	35	28	21	14	7
36	0.57 0.70	0.56 0.71	0.68 0.72	0.91 0.74	1.05 0.73	1.10 0.76	1.36 0.81	1.44 0.82	1.58 0.89	1.43 0.86
32	0.65 0.66	0.59 0.67	0.70 0.68	0.94 0.70	0.98 0.68	1.12 0.71	1.42 0.75	1.46 0.76	1.59 0.81	1.47 0.81
28	0.68 0.59	0.62 0.60	0.72 0.61	0.98 0.62	1.07 0.61	1.06 0.63	1.41 0.66	1.47 0.69	1.48 0.72	1.54 0.71
24	0.72 0.51	0.67 0.52	0.77 0.53	0.98 0.54	1.05 0.53	1.07 0.56	1.47 0.59	1.46 0.62	1.55 0.63	1.53 0.63
20	0.80 0.42	0.77 0.43	0.86 0.44	1.01 0.45	1.14 0.44	1.19 0.47	1.50 0.50	1.57 0.52	1.59 0.52	1.60 0.51
16	0.87 0.34	0.84 0.35	0.91 0.35	1.04 0.36	1.14 0.35	1.25 0.37	1.51 0.39	1.56 0.40	1.82 0.40	1.54 0.39
12	1.05 0.25	1.02 0.25	1.08 0.25	1.19 0.26	1.33 0.25	1.39 0.26	1.56 0.28	1.74 0.29	1.75 0.31	1.71 0.30
8	1.81 0.16	1.83 0.16	1.86 0.16	1.97 0.17	2.29 0.16	2.16 0.17	2.40 0.18	2.58 0.19	2.82 0.21	2.84 0.21
4	2.07 0.08	2.07 0.08	2.16 0.08	2.29 0.08	2.50 0.08	2.54 0.09	2.68 0.09	2.75 0.09	2.94 0.09	2.98 0.10

ERROR (R)
CD_MS

Figure 9: Reconstruction error w.r.t.  $R$  when the number of anchors and mobile nodes change

when  $CD_{MM}$  decreases. The table shows that removing anchors has a bigger impact on the accuracy than removing mobile nodes. This result was expected, since although both of them are used as references, the anchors' positions are static and known, while the others are mobile and with unknown positions.

### 5.2.4 Valley effect

As it was reported in section 5.1.1, increasing the coverage radius, i.e. increasing the connectivity degree, does not necessarily mean increasing the accuracy of the tracking system. However, the opposite is not 100% correct either. The Optimum Radius depends on the structure of the network, which is called Valley effect.

*Definition 3.* **Valley effect** is a tradeoff between the density of the network and the coverage radius, which leads to bigger reconstruction errors when the coverage radius is bigger or smaller than the Optimum one.

The coverage radius can be either a fixed or an adjustable parameter (for example, through an automatic gain control in the radio device). The latter case opens a new research field that will be studied in a near future. Figure 10 shows the situation in which there are 10 anchors in the scenario 2 (4 in the boundary of the simulation scenario and 6 inner anchors randomly chosen), and the number of mobile nodes is reduced from 63 to 7, which assures different network topologies. The coverage radius varies from  $1.5R$  to  $4R$  in each network topology, which illustrates the impact of different connectivity situations due to changing coverage radius, and how the reconstruction error is affected. The 3D figure on top and the corresponding contour underneath shows clearly the Valley effect mentioned above.

## 5.3 Comparison with other tracking algorithms

Most of the proposed tracking algorithms [7, 18, 17, 16], formulate the tracking problem as a sequence of independent snapshots, taking advantage of the high density of nodes in the network (high connectivity degree) to infer the trajectory of the mobile nodes. When the network is not very

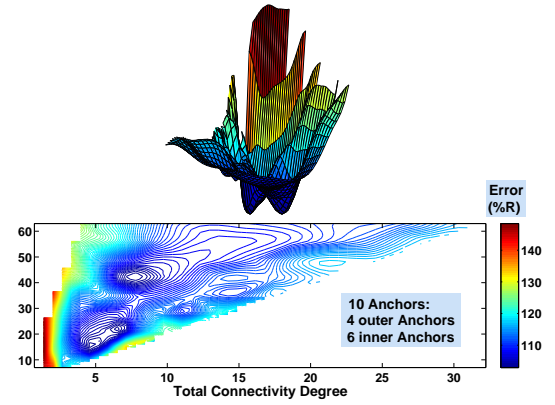


Figure 10: Valley effect

dense, or it does not have certain regularity and the connectivity degree is not so high, these methods lose accuracy exponentially. The tracking system proposed in this paper, is based on short-range radio technologies, seizing on the dynamics of the tracked people and reducing considerably the average connectivity degree needed for a similar performance in previous approaches.

In this section we compared DWMSD with the performance of MDS-MAP(P) and MDS-MAP(P,R) [16], two location algorithms based on classical MDS. Figure 11 plots the median reconstruction error of these three algorithms. MDS-MAP(P) and MDS-MAP(P,R) results are based on a non-uniform network of dimensions  $10R \times 10R$ , and their connectivity degree is augmented, increasing the coverage radius from  $1.25R$  to  $2.5R$  in  $0.25R$  steps, whose values are represented by markers in the figure (green and blue lines). DWMSD results are obtained in a network of the same dimensions (scenario 2) with coverage radius  $R$  and two different configurations: one with 70 mobile nodes and a variable number of anchors, from 100 to 36 (removing 8 anchors each simulation), and from 36 to 12 (removing 4 anchors each simulation), and the second one with 12 anchors and a variable number of mobile nodes, from 70 to 35 (removing 7 mobile nodes each simulation). The red and magenta lines with the corresponding markers report the results respectively.

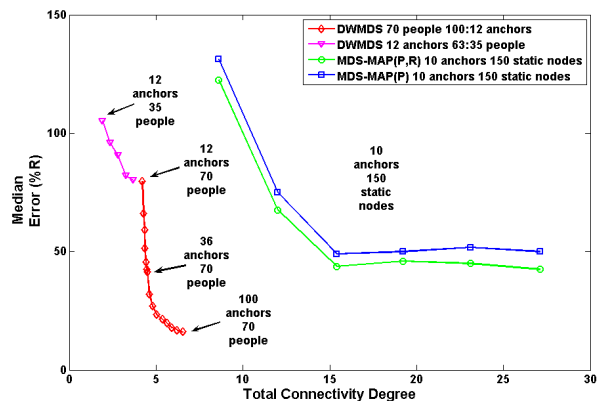


Figure 11: Comparison of reconstruction error of MDS-MAP(P), MDS-MAP(P,R), MDS-MAP and DWMSD

The results show how in MDS-MAP(P) and MDS-MAP(P,R) the bigger the coverage radius (bigger connectivity degree), the smaller the error, until they reach an almost steady level. When the connectivity degree decreases, the reconstruction error in MDS-MAP(P) and MDS-MAP(P,R) increases exponentially. DWMDs delays this effect until the connectivity degree is smaller, due to the removal of anchors and mobile nodes (the network is sparser). As reported in section 5.1.1, DWMDs generally loses accuracy when the coverage radius increases, i.e., when the high connectivity degree is obtained increasing the coverage radius (recall Valley effect). Figure 11 confirms that if the connectivity degree is increased due to a denser network (more anchors, mobile nodes or both of them), then the result is better in general.

Another difference is the Saturation point. MDS-MAP(P) and MDS-MAP(P,R) do not improve accuracy over 10 anchors, while DWMDs has a Saturation point when the network is fully covered by the coverage radius of the anchors, which occurs around 100 anchors. Although the more anchors there are, the more expensive the network will be, our approach also behaves accurately with considerably fewer anchors, and if we increase the number of people (mobile beacons) the accuracy increases in general. Considering that DWMDs works with low connectivity degrees, it can be considered as an ideal tracking method for sparse mobile networks with low connectivity degree where the other methods cannot work properly.

## 6. CONCLUSIONS AND FUTURE RESEARCH

In this paper we have proposed DWMDs, an efficiently and numerically stable algorithm to infer people's trajectories from a set of connectivity matrices through time. The main contribution of the paper has been an extension of classical MDS, DWMDs, that makes use of people's dynamics in the network. The dynamic term effectively links the connectivity data through time regularizing the tracking solution.

DWMDs has proven to be robust to disturbing factors such as erroneous anchors' positions or missing connectivity data. Especially effective when working with sparse networks and short-range technologies, its performance is generally degraded when the coverage radius starts increasing, although an Optimum Radius (recall Valley effect) is dependent on the network conditions. Both anchors and mobile nodes are used to improve the tracking procedure, with especial relevance of those anchors located in the limits of the deployment area (recall Shrinking effect). Compared to other location methods based on connectivity and classical MDS, the results show that DWMDs behaves considerably better in sparse networks.

Currently, we are extending this work in several ways:

- Set up a testbed based on Bluetooth technology in an office scenario with more than 50 people carrying Bluetooth devices. This testbed will be used as a real environment, where DWMDs will be tested and refined. This experiment will allow learning and modeling people's dynamics based on real data.
- Extraction of real location traces for Mobile Ad hoc NETWORKS (MANET) using DWMDs. Measure the impact that the mobility of the nodes has in the performance of the MANET routing protocols.

- Study in detail anchors distribution techniques that minimize the Saturation point, i.e., minimize the number of anchors without compromising the accuracy of the algorithm.
- Study techniques of automatic control of the coverage radius, depending on network topologies pursuing the Optimum Radius at each time instant.

## 7. REFERENCES

- [1] G. Anastasi, R. Bandelloni, M. Conti, F. Delmastro, E. Gregori, and G. Mainetto. Experimenting an indoor bluetooth-based positioning service. In *ICDCSW '03: Proceedings of the 23rd International Conference on Distributed Computing Systems*, page 480, Washington, DC, USA, 2003. IEEE Computer Society.
- [2] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal Ubiquitous Comput.*, 7(5):275–286, 2003.
- [3] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM*, pages 775–784, 2000.
- [4] T. F. Cox, M. A. A. Cox, and T. F. Cox. *Multidimensional Scaling, Second Edition*. Chapman & Hall/CRC, September 2000.
- [5] E. Elnahrawy, X. Li, and R. P. Martin. The limits of localization using signal strength: a comparative study. *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004*, pages 406–414, 2004.
- [6] R. Fletcher. *Practical methods of optimization; (2nd ed.)*. Wiley-Interscience, New York, NY, USA, 1987.
- [7] X. Ji. Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling, In *Infocom 2004*.
- [8] K. Lorincz and M. Welsh. A robust, decentralized approach to rf-based location tracking. Technical Report TR-04-04, Harvard University, 2004.
- [9] N. Marmasse and C. Schmandt. Location-aware information delivery with commotion. In *HUC*, pages 157–171, 2000.
- [10] D. Niculescu and B. Nath. Ad hoc positioning system (aps) using aoa, in *Proceedings of INFOCOM 2003*, San Francisco, CA.
- [11] Y. Ohta, M. Sugano, and M. Murata. Autonomous localization method in wireless sensor networks. In *PERCOMW '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 379–384, Washington, DC, USA, 2005. IEEE Computer Society.
- [12] N. Patwari, A. III, M. Perkins, N. Correal, and R. O'Dea. Relative location estimation in wireless sensor networks. In *IEEE Transactions on Signal Processing*, pages 2137–2148, Piscataway, NJ, USA, 2003. IEEE Signal Processing Society.
- [13] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MobiCom '00: Proceedings of the 6th Annual ACM International Conference on Mobile Computing and Networking*, August 2000.
- [14] T. S. Rappaport. *Wireless Communications: Principles and Practice*. IEEE Press, Piscataway, NJ, USA, 1996.
- [15] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Mobile Computing and Networking*, pages 166–179, 2001.
- [16] Y. Shang and W. Ruml. Improved mds-based localization, In *Infocom 2004*.
- [17] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz. Localization from mere connectivity, *MobiHoc'03*, Annapolis, Maryland. June 2003.
- [18] V. Vivekanandan and V. W. Wong. Ordinal mds-based localization for wireless sensor networks. In *VTC-2006: IEEE 64th Vehicular Technology Conference*, pages 1–5, 2006.
- [19] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler. The effects of ranging noise on multihop localization: an empirical study. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 10, Piscataway, NJ, USA, 2005. IEEE Press.