Tracking Algorithms Based on Dynamics of Individuals and MultiDimensional Scaling

Jose Maria Cabero*, Fernando De la Torre†, Galder Unibaso* and Aritz Sanchez*
*ROBOTIKER-TECNALIA Technology Centre, Telecom Unit, Zamudio, Bizkaia, 48170, Spain
Email: [jmcabero,gunibaso,aslekue]@robotiker.es
†Carnegie Mellon University, Pittsburgh, PA 15213
Email: ftorre@cs.cmu.edu

***Abstract*—Accurate location of people is a key aspect of many applications such as resource management or security. In this paper, we explore the use of radio communication technologies to track people based on their dynamics. The network consists of two types of radio nodes: static nodes (anchors) and mobile nodes (individuals). From a set of sparse dissimilarity matrices with information about proximity or estimated distances between nodes and individuals' dynamics at each time instant, we infer individuals' trajectories. Depending on the information available, two algorithms are proposed: Dynamic Weighted Multidimensional Scaling with Binary Filter (DWMDS-BF) and Dynamic Weighted Multidimensional Scaling based on Distance Estimations (DWMDS-DE). DWMDS-BF is an algorithm that implements a Binary Filter function that obtains very good tracking results when only connectivity information is available and DWMDS-DE is designed for those networks where a good estimation of distances between nearby nodes is available. Both algorithms implement a dynamic component that regularizes the obtained trajectories according to individuals' dynamics. Extensive simulations show the effectiveness and robustness of the proposed algorithms.**

I. INTRODUCTION

Short and medium range radio communication technologies, due to their cheap cost, are being included in almost all personal electronic devices, such as mobile phones, laptops or PDAs. The widespread use of these devices makes them ideal platforms for location-aware applications. Unlike specific tracking technologies such as GPS or those based on ultrasound or image processing, the main purpose of these technologies is not tracking but communication between devices. The most promising current trend is using techniques that can be applied to almost any radio device and that are based on features of the radio communication technologies, like the Received Signal Strength (RSS). Most tracking algorithms consider tracked objects as generic identities, usually called nodes, where no node is different from any other node in the network. They are either considered static nodes or nodes in motion following synthetically generated trajectories (usually random), what leads to non-realistic situations, such as networks with thousands or even millions of nodes with similar behavior. One of the main contributions of our work is a characterization of the mobile nodes according to their dynamics. Although this characterization could be done for almost any kind of network, we are especially interested in social networks, i.e. networks where the nodes to be tracked are people, with different dynamics and patterns. We present a new approach for the tracking procedure as a two-step problem: one dependent on the technology used and the other one dependent on the particular nature of each mobile. We show the effectiveness of using both aspects working together. The main contributions of this work are:

- Dynamic Weighted Multidimensional Scaling with Binary Filter (DWMDS-BF), a tracking algorithm to be used on networks where only connectivity information between nodes is available.
- DWMDS based on Distance Estimations (DWMDS-DE), a tracking algorithm to be used on networks where distance estimations between nodes are available.
- A mathematical expression (called Dynamic term along the article) that uses the learned mobile nodes' motion patterns to smooth the tracking solution according to their particular dynamics. Individual's dynamics means individual's speed in this article.
- A Binary Filter function to handle those scenarios where only connectivity information between nearby nodes is available.

The rest of the paper is organized as follows: section II reviews previous work and section III formulates DWMDS algorithms. The corresponding experiments and comparison with other location algorithms are reported in Section IV. Section V summarizes the conclusions and discusses future research trends.

II. PREVIOUS WORK

Most popular methods to locate people are based on measurements of radio signals, such as Time of Arrival (ToA) [1], Angle of Arrival (AoA) [2], Time Difference of Arrival (TDoA) [3] and Received Signal Strength (RSS) [4]. The first three techniques need costly customized hardware whereas RSS is the most attractive one because of the variety of personal radio communication devices that cheaply and by default implement it. The first ones are better techniques to obtain accurate estimation of the distances, whereas RSS is prone to errors due to the complexity of the radio channel [5].

In spite of the difficulties to model the radio channel, some works based on RSS use trilateration [4], multilateration [6] or similar methods [7], [8] to make an estimation of the distances between the tracked object and some known

978-1-4244-1653-0/08/$25.00 ©2008 IEEE

anchors. However most RSS-based methods do not try to estimate distances directly from RSS, but consist of a previous measurement phase where a RSS map of the scene is built [4], [8], [9], [10], [11]. This methodology is extremely dependent on the environment and any significant change to the topology implies a costly new re-calibration.

In the context of sensor networks, classical MDS [12] has been used to locate static sensor nodes in dense wireless sensor networks [13], [14], [15], [16]. When the networks are more sparse, the accuracy quickly decreases. These previous works rely on hop counts and shortest path measurements to estimate the distances between nodes needed for classical MDS, which leads to poor results in non-uniform networks [15]. In order to solve this drawback, both [14] and [16] have a previous phase where the network is split in subnetworks that locally apply shortest path measurements and classical MDS (the error due to non-uniform networks is reduced but not removed), and that are finally merged to get the resultant network. These methods assume continuous communication between the nodes in the network to transmit information of the state of the network, which imply a high communication cost in terms of bandwidth and energy consumption. Our approach only needs to identify the nearby nodes, which is provided by default by most radio communication technologies, so the communication cost is negligible. The work reported by [17] stresses the importance of reducing the communication cost, trying to reduce it with a technique which chooses adaptively a neighborhood of nodes, applying MDS locally and transmitting the updates to the neighbors. The best results in terms of accuracy are reported by the MDS-MAP(P,R) algorithm [16] using classical MDS and shortest path measurements as a starting point followed by a subsequent optimization phase based on least squares minimization.

III. Tracking as a Low Dimensional Embedding Problem

A. Problem Formulation

We approach the tracking problem from two different perspectives depending on which information is available: connectivity information or estimated distances between connected nodes. Two nodes i and j are considered connected or neighbors if and only if node i is inside the coverage radius of node j and vice versa. In both scenarios, the network consists of mobile and static nodes called anchors used as reference to obtain the trajectories of the mobile nodes. The network can be represented as a graph with vertices V and edges E (G=(V,E)), where the vertices are individuals' positions at each time instant, and the edges join connected nodes at that time instant. The value of the edge is 1 in the connectivity scenario and the corresponding estimated distance in the distance scenario. From now on the term dissimilarity will be also used to address the value of the edges, no matter the scenario described. Once the dissimilarities are gathered through time, they are used together with individuals' dynamics as input of DWMDS algorithm to obtain their trajectories.

Dissimilarity at one time instant between two nodes in the network is recorded according to the following procedure: if a node i is connected to a node j, the ij and ji terms of a dissimilarity matrix are set to 1 (connectivity scenario) or to the estimated distance between them (distance scenario). If both nodes are not connected, they are set to 0 no matter the scenario (see figure 1). Nodes i and j can be anchors or individuals. As section IV will show, not only the anchors work actively in the tracking process but the nodes in motion help too. Considering figure 1: if node A were an anchor and node B and C nodes in motion such that at time t are at those positions, then even although 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.

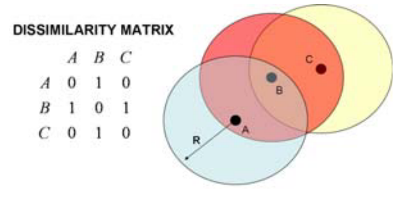

Fig. 1. Connectivity scenario: node B detects A and C. C and A are out of range, so they do not detect each other. Colored circles represent the corresponding coverage areas.

DWMDS-BF is the algorithm for the scenarios with connectivity data, the minimum information available in almost all communication networks. DWMDS-DE can be applied whenever distance estimations are available, no matter the technique used to get them. Its accuracy depends on the accuracy of the technique used to estimate the distances. Both approaches are analyzed in the following sections, although we especially focus on the scenario with connectivity data, which is more widely used and realistic in terms of assumptions.

B. Dynamic Weighted MultiDimensional Scaling (DWMDS)

Multidimensional scaling (MDS) [12] is a powerful statistical dimensionality reduction technique for data analysis which is extensively used in social sciences, engineering and marketing. The starting point of MDS is a matrix consisting of pairwise dissimilarities between data samples in the original space. MDS attempts to find an embedding in a metric space, so that the distances in a low-dimensional space correspond to the given dissimilarities between samples in the original space.

Let[1] $\mathbf{y}_1, ..., \mathbf{y}_n$ be the samples in the original space, and δ_{ij} the corresponding dissimilarities between sample i and j in that space. Let $\mathbf{x}_1, ..., \mathbf{x}_n$ be the coordinates of the samples

[1]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 p, the number of static devices is q, T is the number of time instants. The superindexes s and d correspond to the static and mobile nodes respectively. sd, dd and ss represent the terms static vs mobile nodes, and mobile vs mobile nodes and static vs static nodes respectively.

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. $\mathbf{x}_1, ..., \mathbf{x}_n$) such that d_{ij} in the low-dimensional space is as close as possible to the original dissimilarity in the original space δ_{ij} in the least square sense. It is not usually 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 the ones in equation 1. A local minimum of the these error functions w.r.t $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]$ is usually found by using standard gradient techniques [18]. The general expression of the error function of the DWMDS algorithms consists of two terms, the Static one that uses dissimilarities between nodes and the Dynamic one that comprises the dynamics of the mobile nodes in the network (first and second terms respectively in equation 2). The input to DWMDS algorithms will be a set of matrices with dissimilarity information δ_{ij}^t between nodes at each time instant and the previously learned individuals' dynamics. The final goal is to obtain the coordinates of the nodes through time that minimize the error function.

$$\Omega_1(\mathbf{X}) = \sum_{i<j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}} \quad ; \quad \Omega_2(\mathbf{X}) = \frac{\sum_{i<j}(d_{ij} - \delta_{ij})^2}{\sum_{i<j} \delta_{ij}^2} \; ;$$
$$\Omega_3(\mathbf{X}) = \sum_{i<j} \left(\frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2 \quad . \tag{1}$$

Let us denote the coordinates of the nodes (individuals and anchors) in the network as:

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

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

where $\mathbf{X}^{d,t}$ corresponds to the coordinates of the mobile nodes at time instant 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 in the bidimensional plane. Assuming the algorithm gathers data over T time instants:

$$\mathbf{X}^d = \begin{pmatrix} \mathbf{X}^{d,1} & \mathbf{X}^{d,2} & \ldots & \mathbf{X}^{d,T} \end{pmatrix} \qquad \in \Re^{r \times 2T} \quad ,$$

$$\Omega_{DWMDS}(\mathbf{X}^{ALL}) = \sum_{t=1}^{T} \sum_{\substack{i<j \\ j=2}}^{n} m_{ij}^t (f_{ij}^t - \delta_{ij}^t)^2 \tag{2}$$
$$+ \sum_{t=2}^{T} \sum_{i=1}^{p} \alpha_i^2 \left|\left| \left(\mathbf{x}_i^{d,t} - \mathbf{x}_i^{d,t-1} \right) \right|\right|_F^2 \quad ,$$

where at any time instant:

$$\sum_{i<j} m_{ij}(f_{ij} - \delta_{ij})^2 = \sum_{i<j} m_{ij}^{ss}(f_{ij}^{ss} - \delta_{ij}^{ss})^2$$
$$+ \sum_{i<j} m_{ij}^{dd}(d_{ij}^{dd} - \delta_{ij}^{dd})^2 + \sum_{i<j} m_{ij}^{sd}(f_{ij}^{sd} - \delta_{ij}^{sd})^2 \quad .$$

$\mathbf{X}^{ALL}$ comprises the coordinates of the trajectories $\mathbf{X}^d$ and the position of the anchors $\mathbf{X}^s$. Inside the Dynamic term, α_i is a tradeoff parameter dependent on the mobile nodes' velocity to equilibrate the contribution of the dynamics of each mobile node. $\left|\left| \left(\mathbf{x}_i^{d,t} - \mathbf{x}_i^{d,t-1} \right) \right|\right|_F^2$ is the squared Frobenius norm of the difference between the position of a mobile node i in two consecutive time instants, i.e. the squared distance covered between consecutive times instants (l_i^2). The Dynamic term is the same for all DWMDS algorithms, however according to the nature of the dissimilarities, the Static term changes, generating two different expressions of DWMDS that are explained below (See table I).

1) Estimated distances as dissimilarities. DWMDS-DE algorithm: in those scenarios where estimated distances are available, δ_{ij}^t and f_{ij}^t are the estimated distance and the unknown Euclidean distance (dependent on the coordinates of the i and j nodes) respectively between nodes i and j at time t. $m_{ij}^t = \frac{w_{ij}^t}{\delta_{ij}^t}$ where w_{ij}^t is a weight that stresses the difference between f_{ij}^t and δ_{ij}^t.

The more accurate the estimated distances are the more accurate DWMDS-DE is. So far, the scenarios where only RSS information is available are not reliable enough to accurately estimate distances between nodes, so a connectivity-based approach is more adequate for these networks.

2) Connectivities as dissimilarities. DWMDS-BF algorithm: in scenarios when only connectivity information is available, δ_{ij}^t and f_{ij}^t corresponds to the binary connectivity information (1 if both nodes are connected and 0 otherwise) and a Binary Filter function respectively, between the node i and j at time t. Here $m_{ij}^t = w_{ij}^t$.

DWMDS-BF relies only on connectivity data, information perfectly available from most radio technologies, which makes this DWMDS variant very attractive. While the expression of f_{ij}^t in DWMDS-DE is the Euclidean distance between two nodes at time t, the expression used in DWMDS-BF corresponds to a Binary Filter function dependent on three parameters: the Euclidean distance between nodes, the coverage radius, and a control parameter β that is used to change the slope of the Binary Filter function. Figure 2 shows the Binary Filter function versus the distance between nodes (d_{ij}) for a coverage radius $R = 20$ for different values of β.

The Static term penalizes the difference between the connectivity value (1 if connected, 0 otherwise) and the Binary Filter function which ideally works in the following way: if the coverage area of any node were a circle of radius the nominal coverage radius (R) specified by the manufacturer, the output of this filter would be 1 when the distance between nodes i and j is less than R (connected nodes), 0.5 if it is equal to R and 0 otherwise (disconnected nodes). However, the

	m_{ij}^t	f_{ij}^t
DWMDS-DE	$\frac{w_{ij}^t}{\delta_{ij}^t}$	d_{ij}^t
DWMDS-BF	w_{ij}^t	$\frac{e^{-\beta(d_{ij}^t - R)}}{e^{-\beta(d_{ij}^t - R)} + 1}$

TABLE I
STATIC TERM PARAMETERS

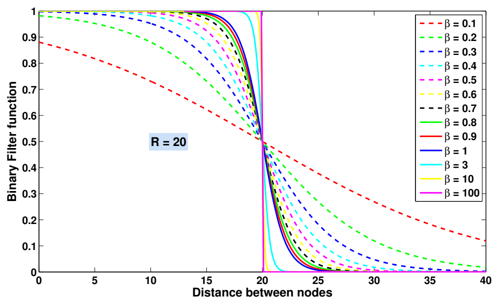

Fig. 2. Binary Filter function for different values of β parameter.

electromagnetic environment is far from being ideal [19], so the question is when to consider two nodes connected. In a real scenario, the complex nature of the electromagnetic field can generate situations where two nodes separated by a distance $d_{ij} < R$ were not connected or even if they were located farther than R, they could detect each other. The trickiest zone is the one around the nominal coverage radius (20 in figure 2) where the value of the function changes abruptly. Far from the nominal radius and for a fixed value of β, the value of the function grows when the distance between nodes tends to 0 (1 is the limit) and tends to 0 when the distance between nodes tends to infinity. The slope of the Binary Filter function and consequently its output is controlled with the β parameter. High values of β mean steep slopes, i.e. almost ideal scenarios, whereas low values of β have the contrary effect, with flatter slopes, for characterizing very unsteady scenarios. Section IV-A shows the impact of the β parameter in medium scenarios, neither especially noisy nor steady.

C. Optimization phase

Once DWMDS expressions are defined, the algorithm enters an optimization phase based on a gradient descent technique 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. The gradient updates are given by:

$$\mathbf{X}_{n+1}^{ALL} = \mathbf{X}_n^{ALL} - \eta \frac{\partial \widehat{\Omega_{DWMDS}}}{\partial \mathbf{X}_n^{ALL}} \quad , \tag{3}$$

where $\frac{\partial \widehat{\Omega_{DWMDS}}}{\partial \mathbf{X}_n^{ALL}}$ is a unit vector in the direction of the gradient. One major problem with the update of equation 3 is to determine the optimal η. In our case η is determined using a line search strategy [18].

D. Weights, Dynamic Term, local minima, initialization, computational cost and connectivity degree

1) Adjustment of weights and the importance of the Dynamic term: weights w_{ij}^t and α_i have to fulfill two requirements: one regarding the term they are working on and the other as a tradeoff parameter between the Dynamic and Static terms. Focusing on the Static term, w_{ij}^t will have a higher value when the corresponding ij term is a term of high confidence, otherwise its value will be low. In DWMDS-DE, if all the estimated distances δ_{ij}^t have the same level of confidence (all of them have been estimated in the same way, which is usual), they will have the same weight, except for those time instants when nodes are not connected, when there are not estimated distances, and consequently the corresponding weights are 0. Unlike DWMDS-DE, DWMDS-BF takes into account those positions out of the coverage radius (0 output in the filter), so weights will have the same value in and out of the coverage radius. The terms in the diagonal $i = j$ are always 0 in both algorithms.

The α_i weights work on the Dynamic term in such a way that the gradient descent technique respects each mobile node's dynamics. Every mobile node will have a specific speed v_i, that lets cover a specific distance l_i between two consecutive time instants (the difference between two consecutive time instants is the time step Δt). Assuming that every mobile node has a constant velocity in motion (which is pretty accurate when the mobile nodes are people walking), the relation between the distance covered by two different mobile nodes i and j in Δt is $l_j = \frac{v_j}{v_i} l_i$. If we consider α_i equal for all i and apply gradient descent method (see section III-C) to minimize the error expression (equation 2), then the fastest nodes (nodes with larger values of l_i), would have more importance in this minimization process. In order to compensate for that effect and treat all the nodes in a democratic way we consider α_i such that

$$\alpha_i = \alpha \frac{v_k}{v_i} \qquad \text{where} \qquad v_k = max\{v_i\} \quad \forall i \,\epsilon\, p. \tag{4}$$

We show an example with two nodes during T time instants, where node 1 is 3 times faster than node 2 ($\alpha_2 = 3\alpha_1$):

$$l_i^2 = \left|\left|\left(\mathbf{x}_i^{d,t} - \mathbf{x}_i^{d,t-1}\right)\right|\right|_F^2 \qquad \forall t, i \quad ,$$

$$\sum_{t=2}^{T}\sum_{i=1}^{2} \alpha_i^2 \left|\left|\left(\mathbf{x}_i^{d,t} - \mathbf{x}_i^{d,t-1}\right)\right|\right|_F^2 = \alpha_1^2 (T-1) l_1^2 + \alpha_2^2 (T-1) l_2^2 = \alpha^2 (T-1)(l_1^2 + 9 l_2^2) \quad ,$$

where $l_1^2 = 9 l_2^2$, so no matter the speed of the mobile nodes, the gradient descent technique will treat them similarly, and respect their different dynamics. Figure 3 shows the distortion in the obtained trajectories when there are two anchors and two linear trajectories with different speeds that last T time instants, but whose dynamics are considered the same ($\alpha_1 = \alpha_2$).

α parameter in equation 4 is a tradeoff parameter between the Dynamic and Static term, that equalizes the value of both terms so that both of them have a similar weight in the general stress expression in equation 2. In this article we deal with synthetic trajectories, so we assume that the dynamics of

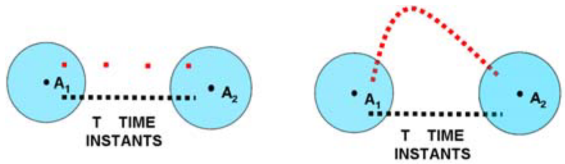

Fig. 3. Left: ground truth. Right: obtained trajectories when $\alpha_1 = \alpha_2$.

each mobile node is known. In real testbeds, there will be a short learning stage where the dynamics of every person in the network is modeled. The procedure is straightforward and consists of measuring the average time it takes for every individual to cover a known distance between two anchors. This can be measured automatically as the time between the connection with one anchor (starting time) and the connection with the other (stop time). The Dynamic term is especially necessary in those networks that are very sparse, where there are gaps with isolated nodes (nodes not connected to any other nodes). This term acts interpolating the position of the nodes in those out-of-coverage parts of the network. When the network becomes more connected, the importance of the Dynamic term varies depending on the scenario:

- When only connectivity data is available (DWMDS-BF), the Dynamic term is always useful no matter the topology of the network. The reason is that in addition to the connectivity gap filler function, this term is necessary to counteract and smooth the otherwise abrupt resultant trajectories from the Static term.
- When estimated distances are available (DWMDS-DE) and the network becomes more connected (fewer connectivity gaps), the obtained trajectories converge gradually with and without the Dynamic term. A different approach is to consider the dissimilarity matrices as the degrees of freedom of the network, so the more connected the network is (the matrices are less sparse because more estimated distances are available), the smaller the ambiguity is, and consequently the less necessary the Dynamic term is. The extreme situation is when the coverage radius of the nodes covers the entire network (dissimilarity matrices totally full), then the DWMDS-DE algorithm has similar results with and without the Dynamic term and the accuracy of the obtained trajectories is 100% dependent on the error in the estimated distances.

2) Local minima: while DWMDS-BF algorithm is pretty robust, DWMDS-DE can converge to local minima due to errors in the estimated distances. Such errors make the gradients of the Static and Dynamic terms of equation 2 try to reduce the error in opposite ways, which makes the algorithm unpredictable. Figure 4 is an example of such a situation, where a mobile node, going from anchor A_1 to A_2 and consequently at $t+1$ is farther from anchor A_1 than at time t, seems to be nearer due to the error in the estimated distance. The Dynamic term gradient would try to follow the trajectory according to the learned dynamics from node A_1 to node A_2, while the Static term would try to take the mobile node backwards.

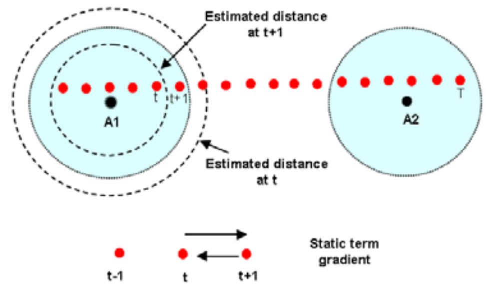

Fig. 4. Up: dotted circles represent estimated distances. Blue circles represent coverage areas. Down: cross direction of the gradient of the Static term over the real positions because of the error in the estimated distances.

The DWMDS-DE performs well when the errors in the estimated distances with respect to the real ones are smaller than the distance covered by the corresponding mobile nodes during a time step Δt, what makes the Dynamic and Static term gradient agree. In very well or full connected networks (nodes connected to all the rest of mobile nodes), the Static term predominates over the Dynamic one, and the contradictory effect disappears.

3) Connectivity degree, initialization and computational cost: connectivity degree is the metric used to measure the average number of connections per node offering a quick view of the density of the network. As we introduce a dynamic term that links different time instants to do the tracking procedure, the connectivity degree in this paper does not measure only the average connectivity at a time instant, but throughout the whole set of time instants in the survey. Equation 5 shows how to calculate the connectivity degree (CD_Total) of a network, where γ_{ij}^t is 1 if nodes i and j are connected at time instant t, 0 otherwise.

$$CD\_Total = \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\gamma_{ij}^t}{nT} \tag{5}$$

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

IV. Performance Evaluation and Comparison

In this section, we report extensive simulation results of the performance of DWMDS-BF and DWMDS-DE in the following scenario: a square of 10r x 10r (r is the reference unit used during the experiments) where there are a variable number of anchors and mobile nodes with random trajectories and speeds. Every mobile node moves according to a constant speed during the tracking time. The velocity range is:

$$v_i \, \epsilon \, \frac{[0.1r, r]}{\Delta t} \quad \forall i \, \epsilon \, p$$

Random trajectories have a duration of $T = 40$ time instants. The positions of the anchors are known and the goal is to obtain the trajectories of the mobile nodes. The reconstruction error in the following sections is the mean error of the difference between the obtained trajectories from the algorithm and the ground truth, unless otherwise stated. The coverage radius (R) is the same for all the nodes, unless otherwise stated. All the simulations are done with MATLAB 7.1 (R14SP3).

A. Scenario with connectivity data: DWMDS-BF algorithm

Section III-B2 remarks the importance of the β parameter in the performance of the Binary Filter function. The goal of this parameter is to adapt the filter to the situation expected in a real scenario where the coverage radius is not the same for every node, even different through time according to changes in the surroundings. To analyze the impact of the β parameter in the performance of the filter and consequently in the whole tracking algorithm, this section analyzes the error of the obtained trajectories in a medium scenario (nor very noisy neither very steady) when the β parameter changes. The noise in the scenario is introduced by modeling the coverage radius of every node as a normal distribution whose mean is the nominal coverage radius (usually provided by the manufacturer of the radio device and considered 2.5r in this experiment) and whose standard deviation is 10% of this nominal coverage radius. The coverage radius changes also through time according to this normal distribution.

Figure 5 shows the best results (and very similar) for β values that make the filter neither very flat nor very steep (β = 0.3,1). Values of the β parameter that make the slope very steep ($\beta = 10$) or flat ($\beta = 0.1$) become in larger reconstruction errors. Although this is the general behavior, there are differences depending on the network configuration:

- For a constant number of anchors:
  - High number of anchors (upper left corner graph): no matter the number of mobile nodes, the difference in the error due to β changes remains pretty constant.
  - Low number of anchors (lower right corner graph): the reconstruction errors obtained with different β values are very similar in sparse networks (they almost converge for very sparse networks) and the differences between them remain pretty constant when the network becomes more connected.

From now on the simulations are carried out with $\beta = 1$.

Figure 6 shows the reconstruction error in DWMDS-BF when the connectivity degree changes due to changes in the coverage radius and the number of nodes. For instance, the lower left corner shows and scenario with 10 anchors and a variable number of mobile nodes when the coverage radius changes from 1.25r to 2.5r in 0.25r steps. Given a constant number of mobile nodes, the colored area shows the reconstruction error when the connectivity degree changes

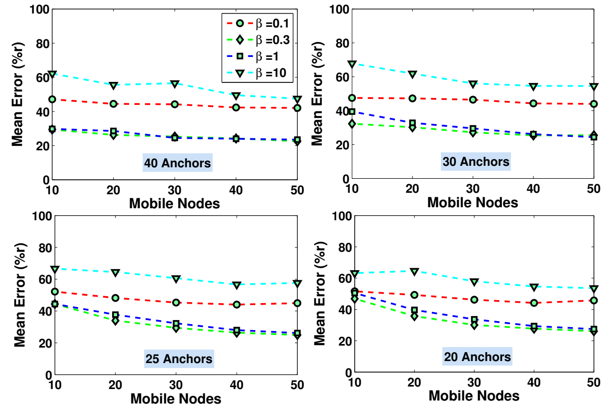

Fig. 5. Reconstruction error when $\beta = 0.1, 0.3, 1, 10$

due to changes in the coverage radius. The results show that whenever the coverage radius increases, the accuracy is higher. When the number of nodes increases (static or mobile ones), the accuracy also increases, with a higher improvement when the nodes are anchors, which was expected because they are known static references.

B. Scenario with estimation of distances: DWMDS-DE algorithm

In This section we focus on the performance of DWMDS-DE when the available estimated distance between nodes are not accurate enough and the tracking algorithm suffers from the pernicious effect detailed in section III-D2.The upper graph of figure 7 compares the performance of DWMDS-BF and DWMDS-DE in networks where the estimation of the distances follows a normal distribution whose mean is the real distance between nearby nodes with standard deviation 5% of this real distance, and the coverage radius used to calculate the connectivities in DWMDS-BF changes according to a normal distribution of mean 2.5r and standard deviation 5% of this radius. When the coverage radius is 2.5r, the errors in the estimated distances are larger than some nodes' speed, happening the effect explained in section III-D2, and consequently the results obtained with DWMDS-BF are even better than the ones obtained with DWMDS-DE. When the radio coverage is enough to cover almost the whole network (very well connected network), the Static term becomes more important than the Dynamic one and takes over the optimization phase. Then the contradictory effect disappears and the results with and without Dynamic term converge (picture at the bottom of figure 7).

C. Comparison with other tracking algorithms

Most proposed tracking algorithms [13], [14], [15], [16], formulate the tracking problem as a sequence of independent time instants, taking advantage of the high density of nodes in the network to infer the position of the nodes at each time instant. When the network is not very dense, or it does not have uniformity, these methods lose accuracy exponentially. Unlike

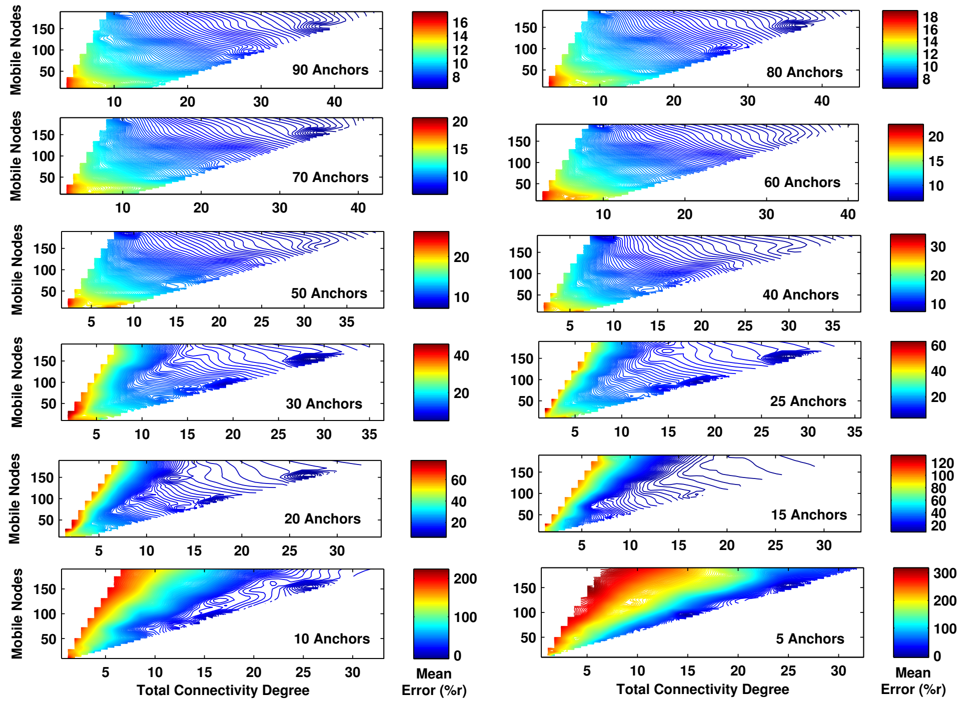

Fig. 6. Mean reconstruction error for different network configurations and coverage radius. Each figure represents networks with a variable number of mobile nodes and the same number of anchors.

previous works, the localization system proposed on this paper, takes advantage of individuals' dynamics and approaches the tracking problem as a whole picture, correlated on time according to each individual's dynamics. Unlike location methods such as MDS-MAP, MDS-MAP(P) and MDS-MAP(P,R), DWMDS algorithms are not based on shortest path measurements, so there is not communication cost between nodes (with the corresponding savings on battery and bandwidth), and their performance is the same no matter the distribution of the nodes in the network (uniform or non-uniform networks). The computational cost dominant term is $O(n^3T)$ in DWMDS and MDS-MAP(P,R), the method that obtains the best results of the MDS-MAP family. DWMDS algorithms obtain better results when the number of anchors increases, while the other methods do not improve the performance above 10 anchors. DWMDS algorithms improve their performance when the radio coverage increases, while the MDS-MAP algorithms do not get better performance from a critical point.

Figure 8 shows the median reconstruction error of four different tracking methods, MDS-MAP, MDS-MAP(P), MDS-MAP(P,R) and DWMDS-BF, in two scenarios where only connectivity data is available: a random-uniform network of 200 nodes and a random-non-uniform network that consists of 160

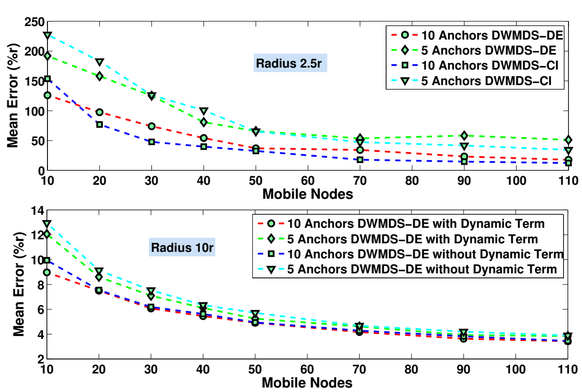

Fig. 7. Up: scenarios with noisy estimated distances. Down: performance of noisy DWMDS-DE with and without Dynamic term.

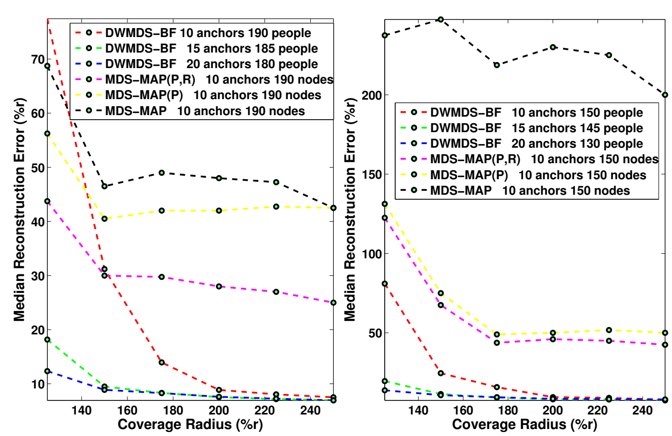

Fig. 8. Left: error in random-uniform networks. Right: error in random-non-uniform networks.

nodes (in both scenarios, some of the nodes are anchors and all the rest are either mobile (DWMDS-BF) or static (MDS-MAP family) nodes). The coverage radius increases from 1.25r to 2.5r in 0.25r steps. DWMDS-BF gets better accuracy than MDS-MAP algorithms when the coverage radius increases (MDS-MAP family algorithms reach an almost steady level) and more in random-non-uniform networks than in random-uniform ones. The accuracy of DWMDS-BF increases when the number of anchors increases. MDS-MAP algorithms do not improve the performance over 10 anchors [16].

V. CONCLUSIONS AND FUTURE RESEARCH

In this paper we have proposed two efficient and numerically stable tracking algorithms to infer individuals' trajectories from a set of dissimilarity matrices through time. DWMDS-BF is an algorithm that obtains very good tracking results for any network when only connectivity information is available and DWMDS-DE is the algorithm designed for those networks when an estimation of the distances between neighbors is available. The main contributions of the algorithms of DWMDS family are a Dynamic term that effectively links the dissimilarity data through time regularizing the tracking solution according to the dynamics of the individuals tracked and a novel Binary Filter function in the Static term of the DWMDS-BF algorithm. Compared to other location algorithms such as the well known MDS-MAP family, DWMDS algorithms do not need communication between nodes, what saves energy and network bandwidth, and work equally well in uniform or non-uniform networks.

Currently, we are extending this work in several ways:

- Setting up a testbed based on Bluetooth technology in an office scenario (headquarters of ROBOTIKER-TECNALIA Technology Centre, Spain) with more than 50 people carrying special designed Bluetooh devices (Bluetooth Medallions). This testbed will be used as a real environment where DWMDS-BF will be tested and refined with real data. Bluetooth technology is already implemented in most of PCs, which contributes to have plenty and very spread anchors all over the scenario.
- Extraction of real tracking traces for Mobile Ad hoc NETworks (MANET) using DWMDS to measure the impact that the mobility of the nodes has in the performance of the MANET routing protocols.
- Study in detail anchors distribution techniques to minimize the number of anchors without compromising the accuracy of the algorithm.

REFERENCES

[1] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Mobile Computing and Networking*, 2001, pp. 166–179. [Online]. Available: citeseer.ist.psu.edu/savvides01dynamic.html

[2] D. Niculescu and B. Nath, "Ad hoc positioning system (aps) using aoa," in Proceedings of INFOCOM 2003, San Francisco, CA. [Online]. Available: citeseer.ist.psu.edu/article/niculescu03ad.html

[3] 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. [Online]. Available: citeseer.ist.psu.edu/priyantha00cricket.html

[4] 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*. Washington, DC, USA: IEEE Computer Society, 2003, p. 480.

[5] E. Elnahrawy, X. Li, and R. P. Martin, *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004*, pp. 406–414.

[6] 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*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 379–384.

[7] 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*. Piscataway, NJ, USA: IEEE Press, 2005, p. 10.

[8] K. Lorincz and M. Welsh, "A robust, decentralized approach to rf-based location tracking," Harvard University, Tech. Rep. TR-04-04, 2004. [Online]. Available: citeseer.ist.psu.edu/lorincz04motetrack.html

[9] 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*. Piscataway, NJ, USA: IEEE Signal Processing Society, 2003, pp. 2137–2148.

[10] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system." in *INFOCOM*, 2000, pp. 775–784.

[11] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*. New York, NY, USA: ACM, 2005, pp. 205–218.

[12] T. F. Cox, M. A. A. Cox, and T. F. Cox, *Multidimensional Scaling, Second Edition*. Chapman & Hall/CRC, September 2000. [Online]. Available: http://www.amazon.co.uk/exec/obidos/ASIN/1584880945/citeulike-21

[13] X. Ji, "Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling," In Infocom 2004. [Online]. Available: citeseer.ist.psu.edu/ji04sensor.html

[14] V. Vivekanandan and V. W. Wong, "Ordinal mds-based localization for wireless sensor networks," in *VTC-2006: IEEE 64th Vehicular Technology Conference*, 2006, pp. 1–5.

[15] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from mere connectivity," MobiHoc'03, Annapolis, Maryland. June 2003. [Online]. Available: citeseer.ist.psu.edu/shang03localization.html

[16] Y. Shang and W. Ruml, "Improved mds-based localization," In Infocom 2004. [Online]. Available: citeseer.ist.psu.edu/shang04improved.html

[17] J. A. Costa, N. Patwari, and I. Alfred O. Hero, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 1, pp. 39–64, 2006.

[18] R. Fletcher, *Practical methods of optimization; (2nd ed.)*. New York, NY, USA: Wiley-Interscience, 1987.

[19] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Piscataway, NJ, USA: IEEE Press, 1996.