Robust Localization in Wireless Sensor Networks via Low-rank and Sparse Matrix Decomposition

Beatrice Rossi, Marco Patanè, Pasqualina Fragneto
AST Lab - STMicroelectronics
beatrice.rossi@st.com
marco.patane-ext@st.com
pasqualina.fragneto@st.com

Andrea Fusiello
DPIA- University of Udine
andrea.fusiello@uniud.it

ABSTRACT

This paper proposes a robust matrix completion method to solve the localization problem in wireless sensor networks. A novel cost function is formulated which inherently copes with missing measures and corrupted data. In particular, the proposed algorithm robustly completes the range map between pairs of sensors by casting the problem as a low-rank and sparse matrix decomposition, while constraining the solution to be close to an Euclidean Distance Matrix. Numerical accuracy and computational efficiency are demonstrated by synthetic experiments. The empirical results also show that our method outperforms state-of-the art algorithms in several scenarios.

CCS CONCEPTS

• **Networks → Network algorithms**; • **Theory of computation →** *Theory and algorithms for application domains*;

KEYWORDS

Robust Matrix Completion, Low-rank and Sparse Decomposition, Euclidean Distance Matrix, Euclidean Distance Matrix Completion, Sensor Network Localization, Wireless Sensor Networks

ACM Reference format:
Beatrice Rossi, Marco Patanè, Pasqualina Fragneto and Andrea Fusiello. 2017. Robust Localization in Wireless Sensor Networks via Low-rank and Sparse Matrix Decomposition. In *Proceedings of ICFNDS '17, Cambridge, United Kingdom, July 19-20, 2017,* 10 pages.
DOI: 10.1145/3102304.3102315

1 INTRODUCTION

Wireless Sensor Networks (WSNs) are a collection of autonomous microelectronic devices distributed over a geographical area (indoor or outdoor) that cooperate to monitor several physical or environmental data. Typically, each sensor consists of a low power processor, a limited amount of memory, a sensor board and a wireless network transceiver. They constitute an instance of a wireless *ad-hoc* network, which can also be *mobile*, depending on the application. In this work we will consider both cases.

WSNs are currently exploited in several applications, including environmental monitoring, failure detection or reporting in smart buildings, and target tracking. These applications rely on accurate position information of sensor nodes: it is necessary to orient the nodes with respect to a global coordinate system in order to report data that is geographically meaningful and to express all the measurements in a common reference frame. Furthermore, basic network services such as routing often rely on location information. These facts motivate the research on the problem of Sensor Network Localization (SNL), in which sensors *self-organize* a coordinate system.

Localization methods can provide a *global* reference frame, meaning that sensor nodes are aligned with some external system, or a *relative* one, meaning that their positions are defined up to an arbitrary rigid transformation from the global frame. At least three non-collinear *anchor nodes* are required to define a global reference frame in two dimensions, and at least four non-coplanar in tree dimensions. Anchor nodes are ordinary sensor nodes that have a priori knowledge about their coordinates. Such information could be hard coded, or acquired through some additional hardware like a GPS receiver.

Typically, SNL methods determine the positions of sensor nodes either starting from pairwise *range* or *bearing* measurements. This work focuses on the case of range measurements.

Distances between pairs of nodes can be measured using the Received Signal Strength Indicator (RSSI) of radio signals sent by neighbor sensors. Indeed, the energy of a radio signal diminishes with the distance from the signal's source. As a result, a node listening to a radio transmission should be able to use the strength of the received signal to estimate its distance from the transmitter. Likewise, sensors nodes can asses the Time of Arrival (TOA) or the Time Difference of Arrival (TDOA) of radio or acoustic signals emitted by neighbor sensors, which in turn can be converted into pairwise distance estimates.

Measuring distances using RSSI or ToA is subject to noise, rogue measures (or outliers), and missing data. Noise depends on radio propagation which tends to be highly non-uniform, outliers are due to environmental factors (sensitivity to reflections, and interferences) or hardware malfunctioning (suffering from transmitter, receiver, and antenna variability), and missing data depend on shields, or limitations in the radio range.

Our solution to the SNL problem is inspired by recent advances in the field of Low-Rank and Sparse (LRS) matrix decompositions.

ICFNDS '17, Cambridge, United Kingdom

DOI: 10.1145/3102304.3102315

We exploit this framework to complete the range information between pairs of sensors in the presence of rogue measures, namely to solve a *robust* Matrix Completion (MC) problem. In particular, we devise a novel cost function that not only includes unspecified measurements and outliers in its definition, but also constrains the solution to stay close to an Euclidean Distance Matrix (EDM), i.e., a matrix containing squared distances between sensors. We also develop a minimization procedure – dubbed Euclidean R-GoDec, or ER-GoDec for short – that is able to effectively minimize such cost function. Moreover we propose a simple variation of ER-GoDec which efficiently handles mobile SNL.

The paper is organized as follows. Existing solutions to the SNL problem are reviewed in Section 2. Section 3 is an overview of the theoretical background on Robust Matrix Completion via LRS matrix decomposition and EDM Completion. In Section 4 we describe our method ER-GoDec and its variation for efficiently addressing mobile networks. In Section 5 we show the results of several experiments on synthetic data and we highlight the benefits introduced by applying our method. The conclusions along with possible further developments are presented in Section 6.

2 KNOWN SOLUTIONS AND LIMITATIONS

We first introduce some notations and basic concepts about SNL. Suppose we have n sensors which we refer to as the *nodes* of the network. We denote the positions of the nodes by a set of n points in $\mathbb{R}^m$ (in our case $m = 2, 3$) ascribed to the rows of a matrix $X \in \mathbb{R}^{n \times m}$, $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T$. The distance between node i and j is $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$, and the associated Euclidean Distance Matrix $D \in \mathbb{R}^{n \times n}$ is defined as the matrix of *squared* distances between the nodes, i.e.

$$D_{ij} := d_{ij}^2 = ||\mathbf{x}_i - \mathbf{x}_j||_2^2. \tag{1}$$

In real world applications, we rarely have a perfect EDM. We know that the measured distances could be affected by noise, outliers, and often we can acquire just a subset. We denote as $\hat{D}$ the matrix of squared *measured* distances, henceforth called *data matrix*.

The goal of localization methods is to recover the point coordinates X from $\hat{D}$. An illustration of the SNL problem is shown in Figure 1.

In range-based SNL, it is useful to model the sensor network using the *distance graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where vertices represent the sensor nodes, and edges correspond to the pairs of sensors for which a distance measurement is available. SNL can also be seen as a *graph realization problem*, where the goal is to assign to each vertex coordinates $\mathbf{x}_i \in \mathbb{R}^m$ so that the Euclidean distances between pairs of nodes match the distances of the corresponding edges.

A sensor network is *uniquely localizable* if exists a unique X that realizes the distances in $\widehat{D}$ (up to a rigid transformation). It is demonstrated in [11] that if the distance graph is *generically globally rigid* – GGR for short –, then the network is uniquely localizable. Jackson *et al.* [22] proved that in two dimensions the distance graph is GGR if and only if it satisfies two properties that can be verified in polynomial time, namely *3-connectivity* and *redundant rigidity*. For more details about these concepts see [22].

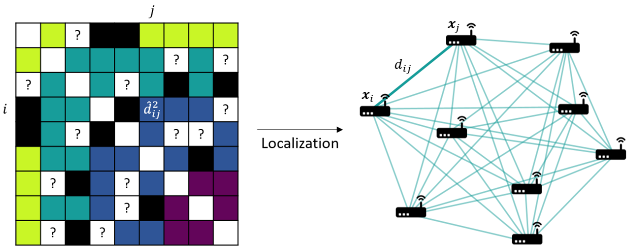

Figure 1: The SNL problem. On the left, the input data matrix $\widehat{D}$: colored entries represent distance measurements $\hat{d}_{ij}^2$, black entries stand for outliers and question marks (?) for unspecified measurements. On the right, the computed positions of the sensor nodes.

We now review the most common localization techniques which appeared throughout the literature.

If all pairwise distance measurements are available, namely $\hat{D}$ is complete, the simplest approach to solving the SNL problem is the classical Multidimensional Scaling (MDS) [8], originally proposed by [29]. MDS exploits the following relationship between the *Gram matrix* $G = XX^T$ and the associated EDM matrix

$$G = -\frac{1}{2}JDJ := \mathcal{T}(D) \tag{2}$$

where J is the *centering operator* defined as $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ ($\mathbf{1}$ denotes the column vector of all ones and I the identity matrix). In particular, MDS minimizes the following cost function

$$\min_X \|\mathcal{T}(\widehat{D}) - XX^T\|_F^2. \tag{3}$$

The point coordinates $X \in \mathbb{R}^m$ can be found by computing the eigen-decomposition of $\mathcal{T}(\widehat{D})$ truncated at the m largest eigenvalues. MDS is fast and closed-form, however it is ineffectual in practice as it does not manage unspecified measurements, and it is not robust to outliers. MDS is summarized in Algorithm 1.

Observe that applying MDS is straightforward, thus there is no practical difference in localization methods between recovering X, or rather the Gram Matrix G, or the EDM D. Indeed, point coordinates can always be derived from G or D by carrying out MDS (in whole or in part).

Algorithm 1 MDS

Require: $\widehat{D}$, m,
Ensure: X,
$J \leftarrow I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$
$G \leftarrow -\frac{1}{2}J\widehat{D}J = \mathcal{T}(\widehat{D})$
$[Q, \Lambda] \leftarrow \text{eig}(G)$
$X \leftarrow Q_m \Lambda_m^{1/2}$

In case of unspecified measurements, the data matrix is a *partial* matrix. This means that the entries of $\widehat{D}$ are specified on a *sampling set*, namely a subset of index pairs, and unspecified elsewhere. Let Ω be a $(0, 1)$-matrix representing the sampling set of $\widehat{D}$, i.e. $\Omega_{ij} = 1$

if $\widehat{D}_{ij}$ is specified and 0 otherwise, then we define $\mathcal{P}_\Omega(\widehat{D}) = \Omega \odot \widehat{D} = A \odot \widehat{D}$, where $\odot$ is the Hadamard (entry-wise) product (where unspecified value multiplied by 0 gives 0) and A is the adjacency matrix of the distance graph $\mathcal{G}$.

An approach to solving the SNL problem in case of missing data is by minimizing the following *stress* function

$$\min_X \|\mathcal{P}_\Omega(\widehat{D}^{1/2} - D^{1/2})\|_F^2. \tag{4}$$

One iterative method to solve (4) is a variant of the gradient descent called stress majorization algorithm, also known as SMACOF [8, 18], originally proposed by [13]. SMACOF is computationally efficient, however, it is not robust to outliers.

The state-of-the-art approach to solve localization is based on Semidefinite Programming (SDP) [6, 7]. SDP is robust and copes with unspecified measurements. The idea is to consider the following cost function

$$\min_X \|\mathcal{P}_\Omega(\widehat{D} - D)\|_1 \tag{5}$$

which can be rewritten in terms of the Gram matrix as

$$\min_G \|\mathcal{P}_\Omega(\widehat{D} - \mathcal{K}(G))\|_1 \quad \text{s.t.} \quad G = XX^T \tag{6}$$

where $\mathcal{K}(G)$ is defined as

$$\mathcal{K}(G) := \operatorname{diag}(G)\mathbf{1}^T - 2G + \mathbf{1}\operatorname{diag}(G)^T = D. \tag{7}$$

Observe that this is another form of the relationship between the Gram Matrix and the associated EDM. The minimization in (6) is then cast into a Semidefinite Programming by relaxing the constraint $G = XX^T$ to $G \succeq 0$, and the resulting problem is solved by interior point methods (as explained in [20]). Naturally, since the minimization variable is G, a partial MDS is then needed to recover the point coordinates $X \in \mathbb{R}^m$.

SDP provides very accurate results. However it does not scale well and relies on sophisticated optimization procedures that cannot be easily implemented on devices with limited resources.

After running any localization procedure which finds point coordinates X, the obtained solution can be expressed in a global coordinate frame using some fixed anchor nodes (at least 3 or 4 in our case) exploiting a *rigid registration* routine. A common procedure exploits Orthogonal Procrustes Analysis [17].

3 MATHEMATICAL BACKGROUND

Our solution to the SNL problem exploits a Low-rank and Sparse decomposition to robustly complete $\mathcal{P}_\Omega(\widehat{D})$. In particular, our method constrains the *completion* of the partial matrix $\mathcal{P}_\Omega(\widehat{D})$ to approach a Euclidean Distance Matrix, namely attempts to compute a EDM completion of the data matrix. For these reasons, we now briefly review the main concepts about robust Matrix Completion via LRS decompositions, and EDM completion.

3.1 Robust Matrix Completion via LRS Matrix Decompositions

Matrix Completion is concerned with the problem of recovering unspecified entries of a *low-rank* partial matrix. A completion of a partial matrix consists in ascribing values to the missing data. Conventional solvers for MC, as for example SVT [10], ALM [25], APGL [28] and OptSpace [23], can fill unspecified entries, but they are not robust to the presence of outlier measurements in the data matrix.

In order to handle outlier measurements, one possibility is to cast MC as a Low-Rank and Sparse matrix decomposition problem. LRS decompositions address the general problem of decomposing a data matrix as the sum of a low-rank term representing some meaningful low-dimensional structure contained into the data, a sparse term representing rogue measures and a term accounting for a diffuse noise. Such a problem arises in a number of applications in Computer Vision, as for example in the separation of foreground objects from the background [3], or in synchronization problems in Structure from Motion [1, 2]. A wide overview of LRS matrix decompositions and their applications can be found in [9].

Let us consider the following LRS decomposition problem

$$\mathcal{P}_\Omega(\widehat{D}) = \mathcal{P}_\Omega(L) + S + N. \tag{8}$$

Here $\mathcal{P}_\Omega(\widehat{D})$ is the partial data matrix, L is an unknown low-rank term representing a *cleaned* completion of the data matrix, S is an unknown sparse matrix representing outliers, and N is a diffuse noise. The problem of computing L (and eventually S) from $\mathcal{P}_\Omega(\widehat{D})$ is known as Robust Matrix Completion (RMC). RMC is illustrated in Figure 2.

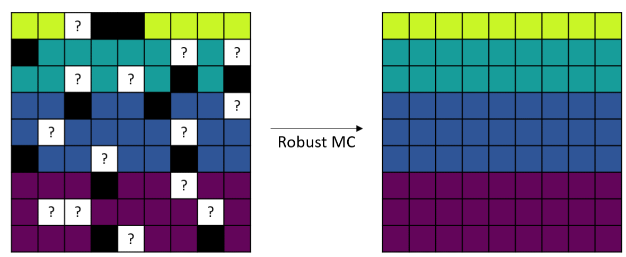

Figure 2: Robust Matrix Completion. On the left, the input matrix: Colors represent the low-rank structure, black entries represent outliers and symbols ? represent unspecified entries. On the right, the completed (and cleaned) low-rank matrix.

R-GoDec [1] is an algorithm that solves RMC under the LRS matrix decomposition framework. R-GoDec is built on GoDec [30] which was originally designed to solve the conventional MC problem only. R-GoDec splits the sparse term as the contribution of two terms S_1 and S_2 with complementary supports:

- S_1 is a sparse matrix with support on Ω representing outlier measurements;
- S_2 has support on $\mho$ (where $\mho$ is the complement of Ω), and it is an approximation of $-\mathcal{P}_\mho(L)$, representing the completion of unspecified entries.

This results in the following decomposition problem

$$\mathcal{P}_\Omega(\widehat{D}) = L + S_1 + S_2 + N \tag{9}$$

which is translated into the minimization

$$\begin{cases} \min\limits_{L,S_1,S_2} \frac{1}{2} \left\| \mathcal{P}_\Omega(\widehat{D}) - L - S_1 - S_2 \right\|_F^2 + \lambda \|S_1\|_1 \\ \text{s.t. } \operatorname{rank}(L) \le r,\ \operatorname{supp}(S_1) \subseteq \Omega,\ \operatorname{supp}(S_2) = \mho \end{cases} \tag{10}$$

R-GoDec solves the problem above using a block-coordinate scheme that alternates the L, S_1, and S_2 updates. L is updated as the rank-r projection of $\mathcal{P}_\Omega(\widehat{D}) - S_1 - S_2$ using Bilateral Random Projections (BRP) [19]. S_1 is updated by applying the *soft-thresholding* operator Θ_λ [5] to the matrix $\mathcal{P}_\Omega(\widehat{D} - L)$. S_2 is updated as $-\mathcal{P}_\mho(L)$. These steps are iterated until convergence.

3.2 Euclidean Distance Matrix Completion

A matrix $D \in \mathbb{R}^{n\times n}$ is an EDM if and only if there exist n points in $\mathbb{R}^m$ ascribed to the row of a matrix $X \in \mathbb{R}^{n\times m}$, $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T$, such that the squared distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is given by $D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$. In particular, every EDM is a non-negative symmetric matrix with zero diagonal.

A notable fact about EDMs is that their rank is at most $m + 2$ (this can be easily derived from (7)). As for the SNL problem, the rank of the associated EDM matrix is thus at most 4 (if $m = 2$) or 5 (if $m = 3$). This result also states that the rank of an EDM is independent of the number n of points. Thus, since we can suppose that $m << n$ in the SNL problem, the associated EDM is low-rank.

Other notable facts about EDMs regard their geometrical structure. The set of all EDMs in $\mathbb{R}^{n\times n}$ forms a closed convex cone referred to as $\mathbb{EDM}^n$. Furthermore, let $\mathcal{K}_1$ and $\mathcal{K}_2$ be two convex cones defined as follows

$$\mathcal{K}_1 = \mathbb{S}_h^n \tag{11}$$

$$\mathcal{K}_2 = \{M \in \mathbb{S}^n \text{ s.t. } -JMJ \succeq 0\} \tag{12}$$

where $\mathbb{S}^n$ is the set of $n \times n$ symmetric matrices, $\mathbb{S}_h^n$ is the set of $n \times n$ symmetric matrices with zero diagonal and J is the centering operator. Then it is demonstrated in [12] that $\mathbb{EDM}^n$ is the intersection of $\mathcal{K}_1$ and $\mathcal{K}_2$. There also exist analytic formulas to project symmetric matrices respectively on $\mathcal{K}_1$ and $\mathcal{K}_2$. Let be $M \in \mathbb{S}^n$, then the projections are given by

$$\mathcal{P}_{\mathcal{K}_1}(M) = M - \operatorname{diag}(M) \tag{13}$$

$$\mathcal{P}_{\mathcal{K}_2}(M) = M - \mathcal{P}_{\mathbb{S}_+^n}(JMJ) \tag{14}$$

where $\mathcal{P}_{\mathbb{S}_+^n}(JMJ) := Q\Lambda^+Q^T$, $Q\Lambda Q^T$ is the eigen-decomposition of JMJ with eigenvalues arranged in non-increasing order, and $\Lambda_{ii}^+ = \max\{0, \Lambda_{ii}\}$ for $i = 1, \dots, n$.

Let now $\mathcal{P}_\Omega(\widehat{D})$ be a partial EDM. A matrix D is an EDM *completion* if D is a completion of $\mathcal{P}_\Omega(\widehat{D})$ and it is an EDM. In other terms, the EDM completion is the problem of finding an EDM that completes a given partial EDM.

We briefly recall some of the available approaches to solve the EDM completion. Observe that, in general, algorithms for Robust Matrix Completion do not guarantee that the completion of a partial EDM is again an EDM.

A first possibility is to combine conventional solvers for RMC with the Alternating Projections Algorithm (APA) [14]. Given two convex sets A, B and a starting point M, APA alternately projects the point on A and B in order to reach a point in their intersection. In [4], it is shown that APA converges as the number of iteration approaches to infinity. As a consequence, the method can be slow, but it can be useful if we have some efficient method, such as closed-form, for the projections.

Exploiting the geometric characterization of $\mathbb{EDM}^n$, and the closed-form projections on $\mathcal{K}_1$ and $\mathcal{K}_2$, APA can be applied to get an EDM which is close to the output of a RMC method. This methodology is depicted in Figure 3.

Mishra *et. al.* in [26] proposed a Manifold Based-Optimization method – MBO for short – by casting the EDM completion into a minimization over the set of low-rank positive semidefinite matrices. In particular, they derived an unconstrained optimization problem on a smooth Riemannian manifold which they tackled via gradient descent.

Another possibility is to reformulate the EDM completion as an instance of the SDP, as explained in [16, 24]. Indeed, since a symmetric zero-diagonal matrix M is an EDM if and only if $\mathcal{T}(M) \succeq 0$, then forcing $\mathcal{T}(M)$ to be PSD is the same of imposing M to belong to $\mathcal{K}_2$. Thus, intuitively, the SDP method, while searching for the Gram Matrix G, also finds an EDM completion of $\mathcal{P}_\Omega(\widehat{D})$.

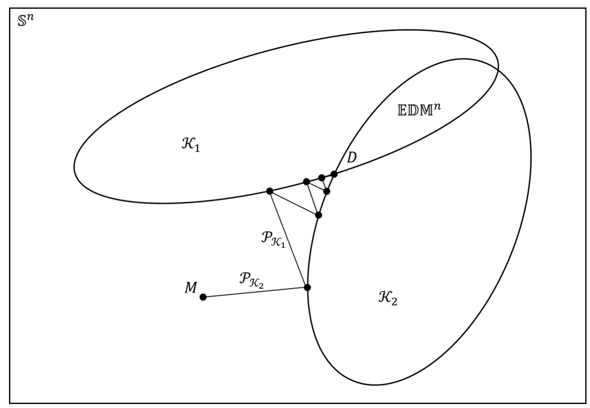

Figure 3: Alternating projections algorithm on $\mathbb{EDM}^n$.

4 PROPOSED METHOD

In this section we describe an algorithm we dubbed ER-GoDec, that robustly solves the EDM completion problem.

Starting from the LRS decomposition proposed by R-GoDec in Eq. (9), we constrain the completion of $\mathcal{P}_\Omega(\widehat{D})$ to belong to $\mathcal{K}_1$ and $\mathcal{K}_2$ through the following minimization problem

$$\begin{cases} \min\limits_{L,S_1,S_2} \frac{1}{2} \left\| \mathcal{P}_\Omega(\widehat{D}) - L - S_1 - S_2 \right\|_F^2 + \lambda \|S_1\|_1 \\ \text{s.t. } \operatorname{rank}(L) \le m + 2, \\ L \in \mathcal{K}_1, \\ L \in \mathcal{K}_2, \\ \operatorname{supp}(S_1) \subseteq \Omega, \\ \operatorname{supp}(S_2) = \mho. \end{cases} \tag{15}$$

This problem can be solved using the procedure reported in Algorithm 2. As in R-GoDec, variables L, S_1 and S_2 are updated iteratively one at a time. A rank-$(m+2)$ projection of $\mathcal{P}_\Omega(\widehat{D}^2) - S_1 - S_2$ is performed in order to find the low-rank term L. The projection is computed using Bilateral Random Projections (BRP) [19]. Then, the resulting L is first projected on $\mathcal{K}_2$ and then on $\mathcal{K}_1$ according to Eq. (14) and Eq. (13) respectively. Finally, S_1 is updated via soft-thresholding of $\mathcal{P}_\Omega(\widehat{D}-L)$ and $-\mathcal{P}_\Omega(L)$ is assigned to S_2.

Algorithm 2 ER-GoDec

Require: $\widehat{D}, \Omega, m, \epsilon, \lambda$
Ensure: L, S_1, S_2
Initialize: $L=\widehat{D}, S_1=0, S_2=0$
while $\left\|\mathcal{P}_\Omega(\widehat{D})-L-S_1-S_2\right\|_F^2 / \left\|\mathcal{P}_\Omega(\widehat{D})\right\|_F^2 > \epsilon$ **do**
(1) $L \leftarrow$ rank-$(m+2)$ approximation of $\mathcal{P}_\Omega(\widehat{D})-S_1-S_2$ via BRP
(2) $L \leftarrow \mathcal{P}_{\mathcal{K}_2}(L)$
(3) $L \leftarrow \mathcal{P}_{\mathcal{K}_1}(L)$
(4) $S_1 \leftarrow \Theta_\lambda(\mathcal{P}_\Omega(\widehat{D}-L))$
(5) $S_2 \leftarrow -\mathcal{P}_\mho(L)$
end while

Once the EDM completion problem is solved via ER-GoDec, the point coordinates X can be obtained by applying the MDS algorithm to L. The complete procedure is depicted in Figure 4.

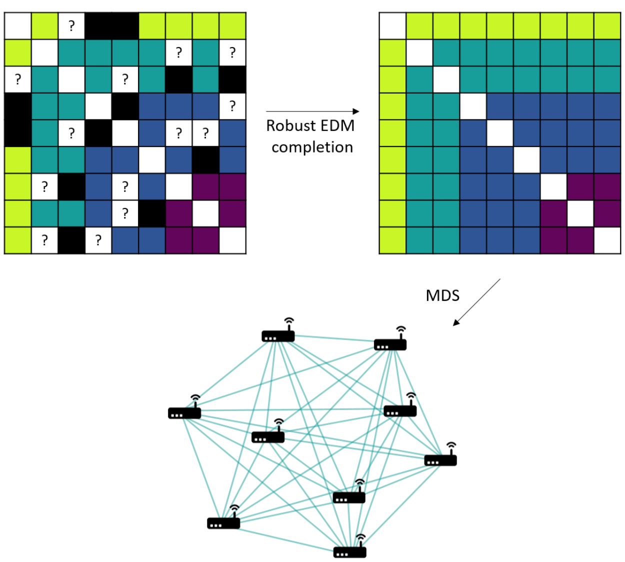

Figure 4: SNL as a robust EDM completion problem. On the top left, the input data matrix: colors represent distance measurements $\widehat{d}_{ij}$, black entries represent outliers and symbols ? represent unspecified entries. On the top right, the completed (and cleaned) EDM. On the bottom, the positions of the sensor nodes computed applying the MDS algorithm to the EDM.

4.1 Incremental ER-GoDec for mobile SNL

Most of the state-of-the-art techniques assume that sensor nodes are in fixed positions, and if they move a new instance of SNL has to be solved from scratch.

Up to this point we worked under the same assumption, albeit not necessary. As a matter of fact, ER-GoDec, can easily cater for mobile sensors. We now present a simple variation of the ER-GoDec algorithm, dubbed Incremental ER-GoDec or IER-GoDec, which efficiently addresses mobile SNL, namely networks where a subset of sensors is moving.

We propose to first perform localization as explained in Section 4 using ER-GoDec. Then, localization in subsequent time points can be obtained using IER-GoDec. If consecutive network configurations are sufficiently close one to each other, then applying IER-GoDec solves the SNL problem more quickly than applying ER-GoDec afresh with new range measurements.

IER-GoDec exploits an alternative initialization of the variables with respect to ER-GoDec. Let us suppose that at time $\tau = t$ the localization is performed via ER-GoDec, and that S_1^t and S_2^t are the sparse terms computed by the algorithm. Let $\mathcal{P}_\Omega(\widehat{D}^{t+1})$ be the partial data matrix at time at time $\tau = t+1$. Then IER-GoDec initializes the low-rank term L as $\widehat{D}^{t+1}-S_1^t-S_2^t$ and then continues implementing the same steps as ER-GoDec.

Again, once the EDM completion problem is solved, the point coordinates X^{t+1} can be obtained by applying the MDS algorithm to L^{t+1}.

The Incremental ER-GoDec is summarized in Algorithm 3.

Algorithm 3 IER-GoDec

Require: $\widehat{D}^{t+1}, S_1^t, S_2^t, \Omega, m, \epsilon, \lambda$
Ensure: L, S_1^{t+1}, S_2^{t+1}
Initialize: $L=\widehat{D}^{t+1}-S_1^t-S_2^t, S_1^{t+1}=S_1^t, S_2^{t+1}=S_2^t$
while $\left\|\mathcal{P}_\Omega(\widehat{D}^{t+1})-L-S_1^{t+1}-S_2^{t+1}\right\|_F^2 / \left\|\mathcal{P}_\Omega(\widehat{D}^{t+1})\right\|_F^2 > \epsilon$ **do**
(1) $L \leftarrow$ rank-$(m+2)$ approximation of $\mathcal{P}_\Omega(\widehat{D}^{t+1}) - S_1^{t+1}-S_2^{t+1}$ via BRP
(2) $L \leftarrow \mathcal{P}_{\mathcal{K}_2}(L)$
(3) $L \leftarrow \mathcal{P}_{\mathcal{K}_1}(L)$
(4) $S_1^{t+1} \leftarrow \Theta_\lambda(\mathcal{P}_\Omega(\widehat{D}^{t+1}-L))$
(5) $S_2^{t+1} \leftarrow -\mathcal{P}_\mho(L)$
end while

5 EXPERIMENTS

In this section we validate our solution through synthetic experiments in MATLAB®.

We first analyze some algorithmic features of ER-GoDec. In particular, we tune the soft-thresholding parameter λ and we evaluate some alternative iteration schemes. Then, we compare ER-GoDec respectively with two methods for SNL (SDP and SMACOF), a combination of a solver for RMC and APA (R-GoDec + APA), and an algorithm for EDM completion (MBO). The implementations of SDP [6] (based on the SDPT3 solver [27]), MBO [26] and SMACOF are available online, while for ER-GoDec and R-GoDec + APA we use our implementations. We compare the performances of the methods in terms of noise resilience, robustness to outliers, sensitivity to missing data, and efficiency. Finally, we assess IER-GoDec in a mobile scenario.

In each experiment, we generate n random points in a squared box of side l centered in the origin, representing the ground truth node positions. We perturb the true distances between points with Gaussian noise with 0-mean and standard deviation σ. We discard a portion ($H\%$) of distances to simulate unspecified data. We set a portion ($O\%$) of the specified distances to random values in the interval $[0, l]$ to simulate outliers. We check the localizability of the distance graph using our implementations of redundant rigidity [21] and 3-connectivity [15].

The error measure we used is the *Root Mean Square Error* (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\left\|X - \tilde{X}\right\|_F^2} \tag{16}$$

where $\tilde{X}$ are the estimated point coordinates and X the true ones. The RMSE is computed after the alignment of $\tilde{X}$ with respect to $\tilde{X}$ obtained using Orthogonal Procrustes Analysis [17] (also available in MATLAB® through the *procrustes* command).

In each experiment, results are averaged over 100 different simulations minus the number of tests in which the localizability check fails.

5.1 ER-GoDec Setup

We first study the behavior of ER-GoDec with respect to the soft-thresholding parameter λ. Figure 5 shows the (average) RMSE obtained using ER-GoDec with different values of λ. The considered parameter configuration is $n = 100$, $l = 100$, $H = 50\%$, $O = 20\%$, and $\sigma = 0.6$.

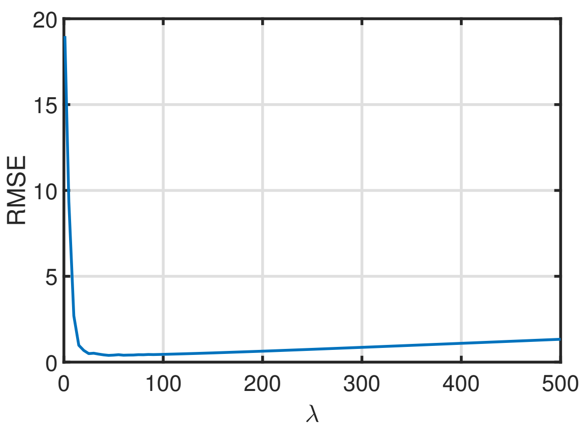

Figure 5: Performances of ER-GoDec as a function of λ. On the x-axis the values of λ, on the y-axis the corresponding (average) RMSE obtained using ER-GoDec. The number of sensors is $n = 100$, the box side is $l = 100$, the portion of unspecified entries is $H = 50\%$, the portion of outliers is $O = 20\%$ and the noise std deviation is $\sigma = 0.6$.

ER-GoDec achieves the best performances (RMSE ≤ 0.5) for $\lambda \in [25, 150]$, hence a reasonable value for λ can be picked within this interval. Since similar trends have been observed also with other parameter configurations, we choose to fix the value $\lambda = 50$ for all the next experiments.

We are also interested in tuning the number of internal projections of ER-GoDec (respectively on $\mathcal{K}_2$ and $\mathcal{K}_1$) with respect to outer iterations, with the purpose of attaining best efficiency and accuracy. Besides the ER-GoDec algorithm described in Algorithm 2, we analyze two different iteration schemes. The first scheme projects L alternatively on $\mathcal{K}_2$ and $\mathcal{K}_1$ 5 times for each outer iteration. The second scheme projects L on $\mathcal{K}_2$ and $\mathcal{K}_1$ every 5 outer iterations. We denote the two schemes respectively as ER-GoDec-A and ER-GoDec-B.

In Figure 6 we plot the *residuals* corresponding to ER-GoDec and the two schemes just defined as function of the number of outer iterations. For each scheme, the residue is defined as $\|\mathcal{P}_\Omega(\widehat{D}) - L - S_1 - S_2\|_F^2 / \|\mathcal{P}_\Omega(\widehat{D})\|_F^2$. The considered parameter configuration is $n = 100$, $l = 100$, $H = 50\%$, $O = 20\%$, and $\sigma = 0.6$.

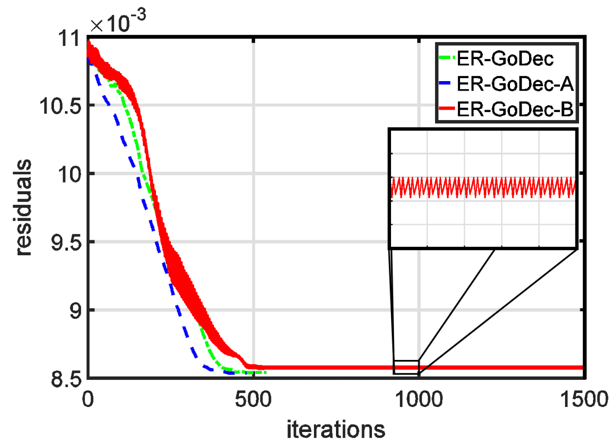

Figure 6: Behaviors of different iteration schemes. On the x-axis the number of iterations, on the y-axis the residuals. The number of sensors is $n = 100$, the box side is $l = 100$, the portion of unspecified entries is $H = 50\%$, the portion of outliers is $O = 20\%$ and the noise std deviation is $\sigma = 0.6$.

The resulting RMSE values are respectively 0.383 for ER-GoDec, 0.379 for ER-GoDec-A, 0.402 for ER-GoDec-B. Figure 6 shows that ER-GoDec and ER-GoDec-A are almost equivalent both in terms of convergence speed (measured in number of outer iterations) and accuracy (measured by RMSE). However, ER-GoDec-A is less efficient than ER-GoDec since each outer iteration of ER-GoDec-A is computationally about 5 time more complex than an outer iteration of ER-GoDec. Conversely, the residuals of ER-GoDec-B present an undesired oscillating behavior, and the method stops after reaching the maximum number of iterations (1500). On the whole, we choose ER-GoDec as reference method, since it balances the trade-off between accuracy and efficiency.

5.2 Noise Resilience

In this experiment we evaluate the noise resilience of ER-GoDec without considering unspecified and outliers data. In particular we set $n = 100$, $l = 100$, $H = 0\%$ (no undefined entries), and $O = 0\%$ (no outliers). Clearly in this case the localizability test is trivial, since the input matrices are complete. Results are shown in Figure 7.

Observe that SMACOF achieves the most accurate results, followed by MBO. This depends on the fact that the SMACOF maximizes a likelihood function (stress), hence it is statistically optimal

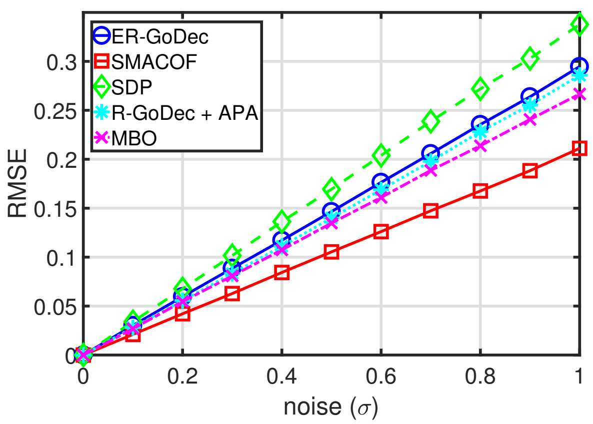

Figure 7: Noise Resilience. On the x-axis the noise std deviation σ, on the y-axis the corresponding (average) RMSE obtained with several methods. The number of sensors is $n = 100$, the box side is $l = 100$, the portion of unspecified entries is $H = 0\%$ and the portion of outliers is $O = 0\%$.

in presence of i.i.d. Gaussian noise. The robust methods, namely ER-GoDec, R-GoDec + APA and SDP, are sub-optimal because they trade statistical efficiency for robustness. All errors grow linearly with respect to the noise std deviation. Thus we fix a reasonable level of noise, namely $\sigma = 0.6$, for the following experiments. This level of noise provides an RMSE approximately lower than 0.2 for all the methods.

5.3 Sensitivity to Unspecified Data

In this experiment we study the sensitivity of ER-GoDec to unspecified data. We set $n = 100$, $l = 100$, $\sigma = 0.6$ and two portions of outliers, namely $O = 0\%$ (no outliers) and $O = 30\%$. Results are shown in Figure 8.

In the case without outliers (upper plot of Figure 8), all the methods are insensitive to unspecified entries up to a portion of the 50%. When the portion of unspecified entries is between the 50% and the 80%, ER-GoDec achieves similar results compared to SDP, better results compared to R-GoDec + APA and MBO, while SMACOF performances get worse. When the portion of unspecified entries is greater than the 80%, the accuracy drops for all the methods, although ER-GoDec performances still remain acceptable.

The lower plot of Figure 8 shows that SMACOF and MBO become sensitive to unspecified entries when outliers are present (although not reported in the figure, their RMSE stands at around 10 even for portions of unspecified entries lower than or equal to the 50%). R-GoDec + APA, ER-GoDec and SDP are not sensitive to unspecified entries up to a portion of the 50%, even in the presence of outliers. When the percentage of unspecified entries is greater than the 50% all the methods worsen. ER-GoDec shows the best breaking point at the 70%.

In our 100 simulations the localizability check fails approximately 20 times when the portion of unspecified entries is greater than the 80%, while for portions lower than the 80% the check never fails.

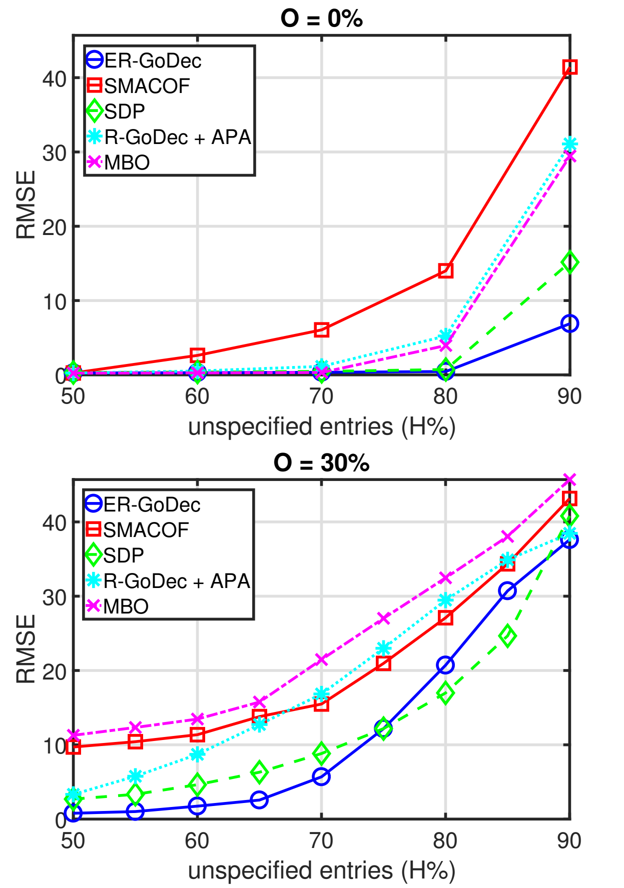

Figure 8: Sensitivity to unspecified data. On the x-axis the portion of unspecified entries $H\%$, on the y-axis the (average) RMSE obtained with several methods. The number of sensors is $n = 100$, the box side is $l = 100$ and the noise std deviation is $\sigma = 0.6$. The portions of outliers are respectively $O = 0\%$ (no outliers) in the upper plot and $O = 30\%$ in the lower plot.

In Figure 9 we fix a set of parameters in order to appreciate the results of a single simulation. As expected, ER-GoDec provides the most accurate localization, followed by SDP and R-GoDec + APA which fail in localizing a couple of sensors (located near the top-right border). Instead SMACOF and MBO obtain inaccurate positions for almost all the sensors. The bottom-right plot is the scatter plot between $D(X)$ (the ground truth EDM) and $D(\tilde{X})$) (the distance matrix associated to the computed positions). Observe that distance values computed by of SDP are almost undetectable to those computed by ER-GoDec.

5.4 Robustness to Outliers

In this experiment we analyze the robustness to outliers of ER-GoDec. We set $n = 100$, $l = 100$, $\sigma = 0.6$ and two portions of unspecified entries, namely $H = 0\%$ (no unspecified entries) and $H = 60\%$.

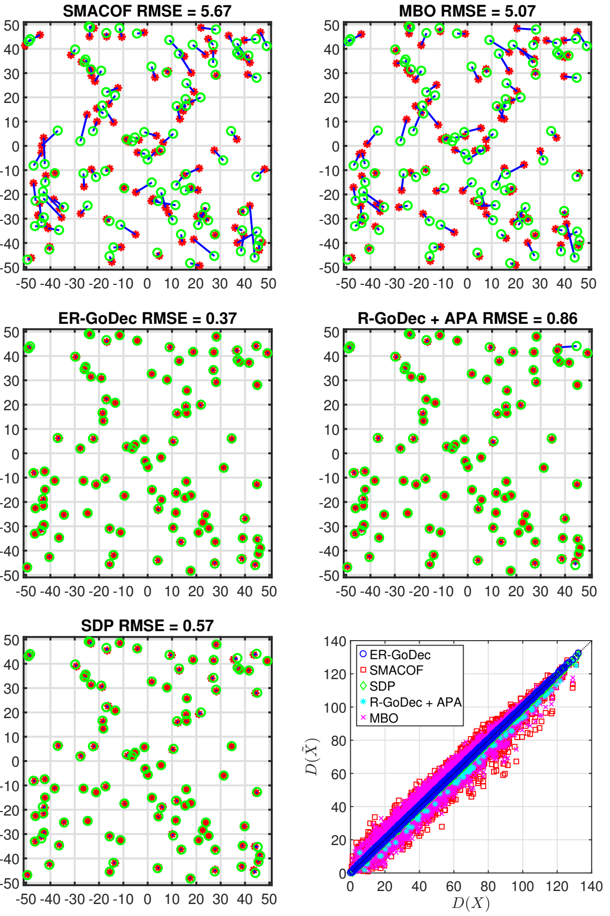

Figure 9: Localization of a network of $n = 100$ sensors in a box $[-50, 50]$ (i.e. $l = 100$) with noise std deviation $\sigma = 0.6$. The portion of unspecified entries is $H = 60\%$ and the portion of outliers is $O = 10\%$. Green circles represent the true point coordinates, red stars the estimated ones and segments the discrepancies between them. The bottom-right plot is the scatter plot between $D(X)$ and $D(\tilde{X})$.

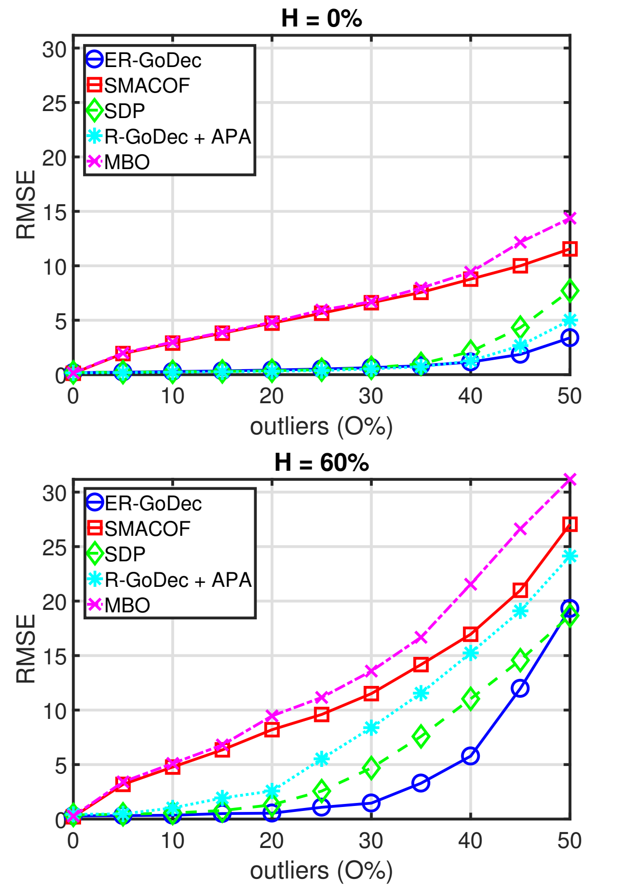

Figure 10: Robustness to outliers. On the x-axis the portion of outliers $O\%$, on the y-axis the (average) RMSE. The number of sensors is $n = 100$, the box side is $l = 100$, and the noise std deviation is $\sigma = 0.6$. The portion of unspecified entries are respectively $H = 0\%$ (no unspecified entries) in the upper plot and $H = 60\%$ in the lower plot.

Results in the upper plot of Figure 10 confirm that ER-GoDec, R-GoDec + APA and SDP are robust, since they obtain rather good results up to a portion of outliers of the 40%. On the contrary, SMACOF and MBO are non-robust, indeed the RMSE is considerable also for portion of outliers lower than the 5%. The lower plot of Figure 10 shows that ER-GoDec has the best breaking point at the 30%, achieving accurate results also for high portions of both unspecified data and outliers, while R-GoDec + APA and SDP obtain worse results. SMACOF and MBO produce highly inaccurate positions.

An explanation for these results is that ER-GoDec and R-GoDec + APA correctly identify (and discard) outliers because they are robust MC methods. The robustness of SDP depends on the fact that it minimizes a robust cost function (based on ℓ^1-norm). On the contrary, SMACOF and MBO fail in identifying outliers since they minimize non-robust cost functions (based on the Frobenius norm).

As for the localizability check, we observe that with these parameter configurations the check never fails.

Figure 11 confirms that the accuracy of the localization is more influenced by outliers rather than by random unspecified entries. Furthermore, only ER-GoDec provides an accurate localization, while R-GoDec + APA and SDP cannot manage such portions of unspecified data and outliers. SMACOF and MBO get completely inaccurate localizations. The scatter plot between $D(X)$ and $D(\tilde{X})$ (the bottom-right plot) verifies that only ER-GoDec accurately recovers the distance matrix.

5.5 Execution Time

In this experiment we evaluate the execution time as a function of the number of sensors. We consider $l = 100$, $\sigma = 0.6$, $H = 30\%$ and $O = 10\%$. Results are reported in Figure 12.

From the upper plot of Figure 12 we can see that SDP is drastically slow (10 to 50 times slower than the other methods). Also R-GoDec + APA is quite slow with respect to SMACOF, ER-GoDec and MBO. In

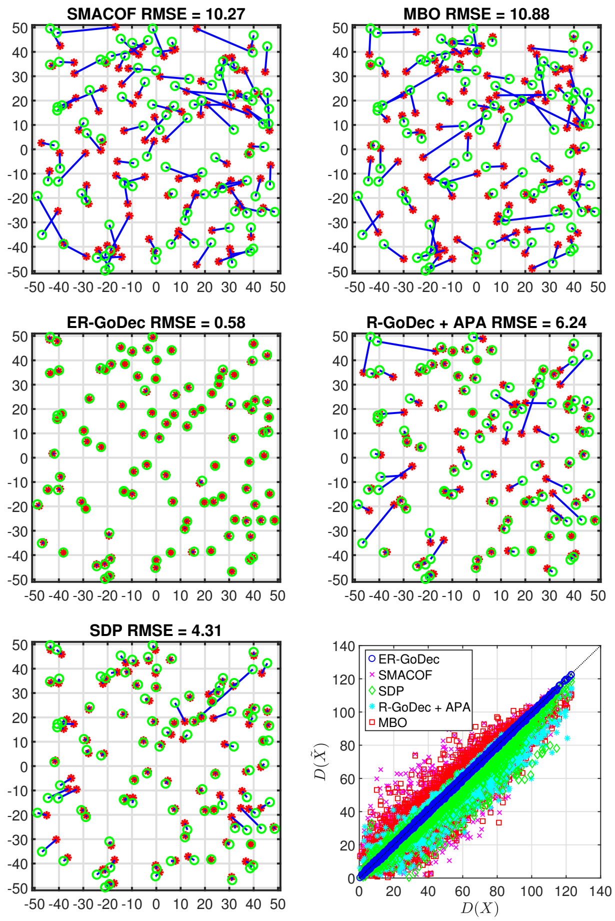

Figure 11: Localization of a network of $n = 100$ sensors in a box $[-50, 50]$ (i.e. $l = 100$) with noise std deviation $\sigma = 0.6$. The portion of unspecified entries is $H = 60\%$ and the portion of outliers is $O = 30\%$. Green circles represent the true point coordinates, red stars the estimated ones and segments the the discrepancies between them. The bottom-right plot is the scatter plot between $D(X)$ and $D(\tilde{X})$.

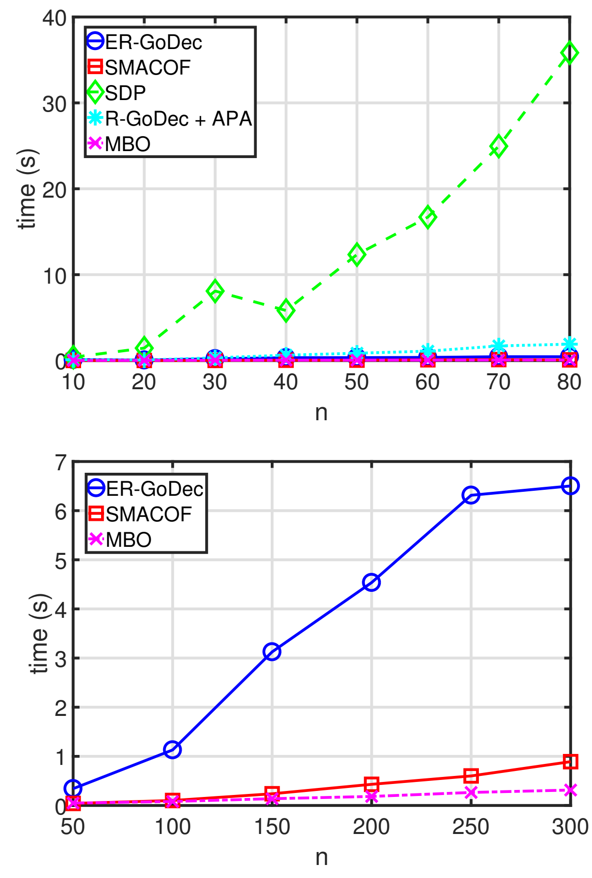

Figure 12: In the upper plot, the execution time of all the methods. In the lower plot, execution time of SMACOF, ER-GoDec and MBO. On the x-axis the number of sensor nodes (n), on the y-axis the execution time (in seconds). The box side is $l = 100$, the portion of unspecified entries is $H = 30\%$, the portion of outliers is $O = 10\%$ and the noise std deviation is $\sigma = 0.6$.

order to compare the execution time of the most efficient methods, we increased the number of sensors excluding SDP and R-GoDec + APA from the evaluation. The lower plot of Figure 12 shows that MBO and SMACOF are noticeably faster with respect to ER-GoDec, even if ER-GoDec still remains acceptable.

We observe that with this portion of unspecified entries (30%) the localizability check never fails.

5.6 IER-GoDec

We evaluate Incremental ER-GoDec through the following experiment: we proceed as explained in Section 5 for generating $\widehat{D}^t$ and we localize the network using ER-GoDec. Then we move approximatively one unity (in the plane) a random subset of sensors generating $\widehat{D}^{t+1}$. Figure 13 shows the speed-up obtained by using IER-GoDec at time $\tau = t + 1$ rather than using ER-GoDec at time $\tau = t + 1$ from scratch. The accuracy of the two methods is the same.

We can appreciate that the speed-up is significantly high if few sensors are allowed to move. Anyway, even if all sensors are moving the speed up is still favorable to IER-GoDec.

We can conclude that IER-GoDec is a promising approach to solve the SNL problem in mobile scenarios as those described in Section 4.1.

6 CONCLUSIONS

In this paper we have presented ER-GoDec: a novel algorithm which exploits a Low-Rank and Sparse matrix decomposition tailored to SNL. We argue that the benefit yielded by ER-GoDec is twofold. First, it is both efficient and highly accurate, as demonstrated by synthetic experiments. Such occurrence permits to localize the network using low-power processor. The second benefit

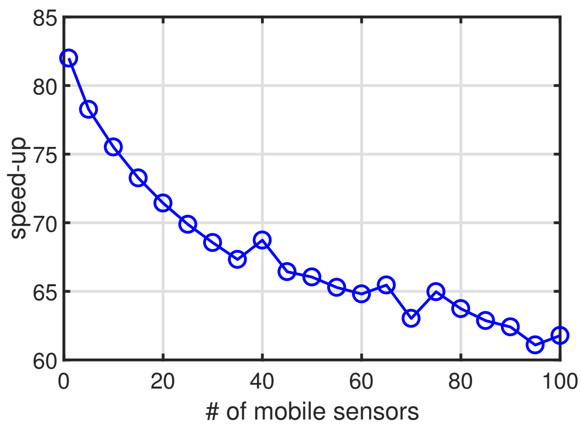

Figure 13: Efficiency of IER-GoDec. On the x-axis the number of mobile sensors, on the y-axis the speed-up (expressed in terms of gained number of iterations of IER-GoDec with respect to ER-GoDec, namely $\left(1 - \frac{\#\mathbf{iter}(iER)}{\#\mathbf{iter}(ER)}\right) 100\%$) obtained by using IER-GoDec at time $\tau = t + 1$ (after having applied ER-GoDec at time $\tau = t$) rather than using ER-GoDec at time $\tau = t + 1$ from scratch. The box side is $l = 100$, the portion of unspecified entries is $H = 30\%$, the portion of outliers is $O = 10\%$ and the noise std deviation is $\sigma = 0.6$.

is that, thanks to its structure, ER-GoDec can easily cater for networks where a subset of sensors is moving and therefore it can be profitably exploited for tracking purposes.

Future work will be aimed at better characterizing the performances of IER-GoDec with respect to node density, node speed and motion model.

REFERENCES

[1] Federica Arrigoni, Luca Magri, Beatrice Rossi, Pasqualina Fragneto, and Andrea Fusiello. 2014. Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In *2014 2nd International Conference on 3D Vision*, Vol. 1. IEEE, 491–498.

[2] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. 2016. Global Registration of 3D Point Sets via LRS Decomposition. In *European Conference on Computer Vision*. Springer, 489–504.

[3] Laura Balzano, Robert Nowak, and Benjamin Recht. 2010. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 704–711.

[4] Heinz H Bauschke and Jonathan M Borwein. 1993. On the convergence of von Neumann's alternating projection algorithm for two sets. *Set-Valued Analysis* 1, 2 (1993), 185–212.

[5] Amir Beck and Marc Teboulle. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences* 2, 1 (2009), 183–202.

[6] Pratik Biswas, Tzu-Chen Liang, Kim-Chuan Toh, Yinyu Ye, and Ta-Chung Wang. 2006. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE transactions on automation science and engineering* 3, 4 (2006), 360.

[7] Pratik Biswas, Kim-Chuan Toh, and Yinyu Ye. 2008. A distributed SDP approach for large-scale noisy anchor-free graph realization with applications to molecular conformation. *SIAM Journal on Scientific Computing* 30, 3 (2008), 1251–1277.

[8] Ingwer Borg and Patrick JF Groenen. 2005. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media.

[9] Thierry Bouwmans, Necdet Serhat Aybat, and El-hadi Zahzah. 2016. Handbook of Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing. (2016).

[10] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20, 4 (2010), 1956–1982.

[11] Robert Connelly. 2005. Generic global rigidity. *Discrete & Computational Geometry* 33, 4 (2005), 549–563.

[12] Jon Dattorro. 2010. *Convex optimization & Euclidean distance geometry*. Lulu. com.

[13] Jan De Leeuw, In JR Barra, F Brodeau, G Romier, B Van Cutsem, and others. 1977. Applications of convex analysis to multidimensional scaling. In *Recent Developments in Statistics*. Citeseer.

[14] Richard L Dykstra. 1983. An algorithm for restricted least squares regression. *J. Amer. Statist. Assoc.* 78, 384 (1983), 837–842.

[15] Shimon Even. 2011. *Graph algorithms*. Cambridge University Press.

[16] Haw-ren Fang and Dianne P O'Leary. 2012. Euclidean distance matrix completion problems. *Optimization Methods and Software* 27, 4-5 (2012), 695–717.

[17] Colin Goodall. 1991. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)* (1991), 285–339.

[18] Craig Gotsman and Yehuda Koren. 2004. Distributed graph layout for sensor networks. In *Graph Drawing*. Springer, 273–284.

[19] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. 2009. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. (2009).

[20] Christoph Helmberg, Franz Rendl, Robert J Vanderbei, and Henry Wolkowicz. 1996. An interior-point method for semidefinite programming. *SIAM Journal on Optimization* 6, 2 (1996), 342–361.

[21] Bruce Hendrickson. 1992. Conditions for unique graph realizations. *SIAM J. Comput.* 21, 1 (1992), 65–84.

[22] Bill Jackson and Tibor Jordán. 2005. Connected rigidity matroids and unique realizations of graphs. *Journal of Combinatorial Theory, Series B* 94, 1 (2005), 1–29.

[23] Raghunandan H Keshavan, Sewoong Oh, and Andrea Montanari. 2009. Matrix completion from a few entries. In *2009 IEEE International Symposium on Information Theory*. IEEE, 324–328.

[24] Monique Laurent. 2009. Matrix Completion Problems. *Encyclopedia of Optimization* 3 (2009), 221–229.

[25] Zhouchen Lin, Minming Chen, and Yi Ma. 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055* (2010).

[26] Bamdev Mishra, Gilles Meyer, and Rodolphe Sepulchre. 2011. Low-rank optimization for distance matrix completion. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 4455–4460.

[27] Kim-Chuan Toh, Michael J Todd, and Reha H Tütüncü. 2012. On the implementation and usage of SDPT3–a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. In *Handbook on semidefinite, conic and polynomial optimization*. Springer, 715–754.

[28] Kim-Chuan Toh and Sangwoon Yun. 2010. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization* 6, 615-640 (2010), 15.

[29] Warren S Torgerson. 1952. Multidimensional scaling: I. Theory and method. *Psychometrika* 17, 4 (1952), 401–419.

[30] Tianyi Zhou and Dacheng Tao. 2011. Godec: Randomized low-rank & sparse matrix decomposition in noisy case. (2011).