Dictionary Design for Sensor Network Localization via Block-Sparsity

Dictionary Design for Sensor Network Localization via Block-Sparsity

Alessandro Bay #1, Diego Carrera *2, Sophie M. Fosson #3, Pasqualina Fragneto †4, Marco Grella †5, Chiara Ravazzi #6, Enrico Magli #7

Department of Electronics and Telecommunications (DET), Politecnico di Torino, Italy

{1 alessandro.bay, 3 sophie.fosson, 6 chiara.ravazzi, 7 enrico.magli}@polito.it

* *Department of Electronics, Information Science and Bioengineering (DEIB) Politecnico di Milano, Italy*

2 diego.carrera@unimi.it

† *Advanced System Technology (AST) STMicroelectronics, Italy*

{4 pasqualina.fragneto, 5 marco.grella}@st.com

***Abstract*—In this paper, we consider the problem of RSS-fingerprinting localization in wireless sensor networks. In particular, inspired by the recent advances in sparse approximation and compressive sensing theory, we propose a localization scheme based on the dictionary design of block-sparse signals. We show via numerical simulations and real experiments that the proposed technique outperforms traditional fingerprinting methods.**

I. Introduction

Localization has recently taken advantage of the development of wireless sensor networks (WSNs), which are a valuable alternative to satellite-based technologies in indoor environments or in outdoor adverse conditions [1], [2], [3], [4], [5].

The localization methods based on the measurement of the received signal strength (RSS), i.e. the power transmitted by the device to be localized, have been widely considered, in particular combined with fingerprinting, which is a map-based approach that creates a dictionary in order to represent the physical space capturing possible variations. More precisely, RSS measurements are collected off-line at some known locations in the area and then stored in a dictionary; the unknown location can then be obtained on-line from the current RSS measurements which are compared with those in the dictionary.

The drawback of RSS-fingerprinting is the large number of data that have to be exchanged between the receiver and the transmitter to achieve desired performance. This issue has been recently addressed by recasting localization as a sparse recovery problem [6], [7], [8]. The rationale is the following: assuming that the physical space is discretized in a grid of D cells, each of them associated with a prescribed position, the device position can be represented by a vector of length D that has nonzero entries only where a device occupies that position. Typically this vector is sparse, that is, has few nonzero entries, provided that there are few devices with respect to the domain dimensions. Localization can then be performed by searching for the sparsest solution.

MMSP'15, Oct. 19 - Oct. 21, 2015, Xiamen, China.

Studying localization is motivated by several safety and monitoring tasks, such as detecting medical equipment or products in hospitals or warehouses, or, more generally, tasks like indoor guidance for example in huge shopping malls, museums, or airports, where mobility is fundamental. In all these cases, we are in a wide environment, which can be discretized in many cells, and, if we want to find few targets among all possible positions, then the vector is sparse.

In this paper, we present new methods, based on block-sparsity, that aim at improving the localization accuracy. Block-sparsity assumes that the nonzero entries of the sparse signal to be reconstructed are concentrated in some patterns; recent work [9], [10] provides theoretical guarantees for their reconstruction in a compressed sensing framework. Leveraging these general results, we explore RSS-fingerprinting methods based on block-sparsity and we prove their efficiency.
In particular, we rearrange the blocks by exploiting intra-block correlation of the dictionary, which improves recovery performance. We evaluate the performance of the proposed localization system through a number of simulations and practical experiments, showing the accuracy in terms of localization distance error with respect to existing systems.

The paper is organized as follows. In Section II we introduce the localization scenario, defining the setting of our model; in Section III we explain the proposed algorithm based on block-sparsity. Section IV is devote to numerical simulations, that validate the choice of a kind of block and illustrate results obtained in different noise settings. In Section V we present results of on-field real experiments, both in outdoor and indoor environments, and we compare our method to the related literature ones. Finally, in Section VI we close with some concluding remarks and future developments of our work.

II. Model setting

In this section, we introduce the localization scenario that we consider throughout the paper. Let $\mathcal{A} \subset \mathbb{R}^2$ be a two-dimensional region representing the area of interest where a device to be localized is placed. We then consider a discretization of $\mathcal{A}$ by setting $D \in \mathbb{N}$ reference points (RPs) in $\mathcal{A}$, whose coordinates are indicated by ξ_i, $i \in \{1, \dots, D\}$, and partitioning $\mathcal{A}$ into D subsets (or cells), each of them containing a RP. For simplicity of exposition, in this work we consider a rectangular $\mathcal{A}$ with a uniform grid of R rows and C columns, and $D = RC$ RPs.

The localization task consists in detecting the cell occupied by the device. To this purpose, we set J base stations (BSs), inside or outside $\mathcal{A}$, which collect measurements from the RPs and transmit information to a central unit that processes the data and performs the localization task. In our model, the BSs are assumed to acquire the RSS of the signals, which is a measure of the received power. RSS methods are widely used for localization as they are among the most inexpensive and power saving [6], [11]. Taking into account the dissipation due to the distance and obstacles between the transmitting device and the receiver (*e.g.*, walls, furniture, and people) the RSS at distance d is modeled as follows [6]:

$$P_r(d) = P_t - \overline{PL}(d_0) - 10n \log_{10}(d/d_0) - \eta_\sigma, \quad (1)$$

where P_t is the transmitting power, $\overline{PL}(d_0)$ is the average path loss value at a reference distance $d_0 = 1$ m; n is an attenuation parameter (generally $2 \leq n \leq 4$), η_σ is a zero-mean Gaussian noise with standard deviation σ, that we denote as $\eta_\sigma \sim \mathcal{N}(0, \sigma^2)$.

A. Fingerprinting techniques

RSS-fingerprinting consists in creating a dictionary (or signature map) that represents the RPs of $\mathcal{A}$ and then comparing RSS measurements with such fingerprints. More precisely, we distinguish two phases, known as training (or off-line) phase and runtime (or on-line) phase.

During the *training phase*, a dictionary is created as follows. As depicted in Figure 1, a transmitting device is set in turn in each RP and broadcasts a signal $T \geq 1$ times. Each BS, labeled with $j \in \{1, 2, \dots, J\}$, stores the value $\psi^j_{t,i}$ representing the t-th measure of RSS from i-th RP. Each BS is then associated with a map Ψ^j that can be written as the measurements matrix

$$\Psi^j = \begin{pmatrix} \psi^j_{1,1} & \psi^j_{1,2} & \cdots & \psi^j_{1,D} \\ \vdots & \vdots & \ddots & \vdots \\ \psi^j_{T,1} & \psi^j_{T,2} & \cdots & \psi^j_{T,D} \end{pmatrix} \in \mathbb{R}^{T \times D}.$$

Finally, each BS sends its own Ψ_j to the central unit, which stores the global dictionary

$$\Psi = \left(\Psi^{1\top}, \Psi^{2\top}, \dots, \Psi^{J\top}\right)^\top \in \mathbb{R}^{TJ \times D}.$$

The localization is performed in the *runtime phase*. As illustrated in Figure 2, for all $j \in \{1, \dots, J\}$, the j-th BS receives the RSS measurements $z_j \in \mathbb{R}^{\widetilde{T}}$, and transmits them to a central unit that collects them into the global measurement vector.

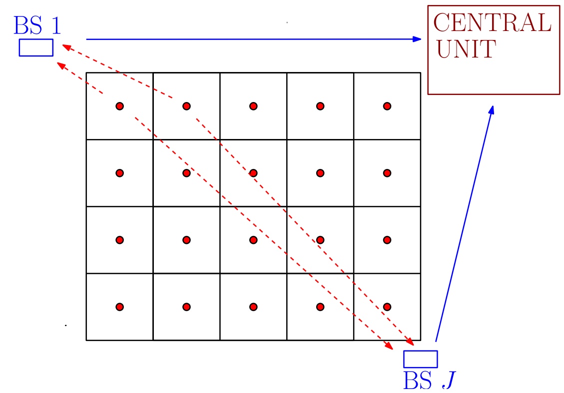


Fig. 1. Illustration of training phase: each RP transmits T times to the BSs, which in turn send the data to the central unit, which collects the signature map.

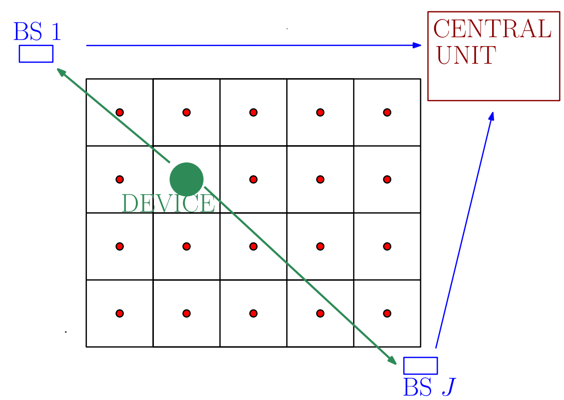


Fig. 2. Illustration of runtime phase: a device is placed in cell and transmits to the BSs, which forward the information to the central unit, which tries to localizes the device using the database.

At this point, given Ψ and z, the central unit estimates the position of the device comparing z with Ψ. Intuitively, if the device is in the ℓ-th cell, each BS j is expected to receive a signal z_j *close* to $\psi^j_{t,\ell}$, $t = 1, \dots, T$. Based on the notion of distance that one considers, different algorithms can be developed [12].

III. Block-sparsity based localization

We now introduce some elements of spatial sparsity localization; further we introduce block-sparsity, which is the theoretical basis of our novel approach to localization.

A. Localization via spatial sparsity

Localization via spatial sparsity builds on the key idea that one or few devices positions can be represented with a sparse vector, whose components represent the cells and are null at empty cells. The mathematical problem [6], [13] consists in finding the sparsest $x \in \mathbb{R}^D$ such that $z = \Psi x + \eta$ where

$\eta \in \mathbb{R}^{TJ}$ is a small error, subject to the constraint that just one position is occupied by each device. A common formulation to this problem is the following:

$$\min \|z - \Psi x\|_2 \qquad \text{s. t. } \|x\|_0 \leq k \tag{2}$$

where $\|x\|_0 = |\{i \in \{1, \ldots, D\} | x_i \neq 0\}|$ and k is the number of devices to be localized.

B. Localization via block-sparse representations

Our block-sparsity approach refines the spatial sparsity methods by grouping the cells and proposing a hierarchic search of the device position.

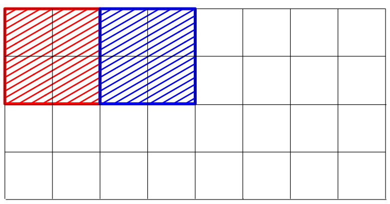
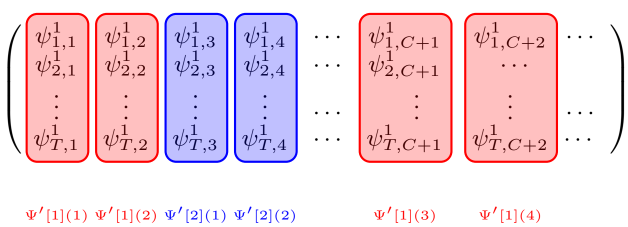


Fig. 3. An example of block partitioning.

The idea is as follows. In the training phase some adjacent cells, which form a partition of the ground floor, are grouped into N macroblocks. The dictionary Ψ is then rearranged accordingly. For example, let us consider a 4×8 grid and $N = 8$ macroblocks composed by 2×2 blocks of adjacent cells as shown in Figure 3.

Ψ has the following structure:



where the different colors represent different blocks, composed by the cells indicated in Figure 3, i.e. $\Psi'[i](j)$ represent the RSS value from j-th cell of the i-th macroblock. Thus, Ψ is then rearranged to form a new dictionary, denoted by

$$\Psi' = \Big[\Psi'[1], \Psi'[2], \ldots, \Psi'[N]\Big].$$

We also rearrange $x \in \mathbb{R}^D$ into a concatenation of N blocks $x = (x[1]^\top, x[2]^\top, \ldots, x[N]^\top)^\top$. For simplicity, let us suppose that the k devices to be localized occupy k different macroblocks. Thus, each of the k nonzero components of x belongs to a different block and x has k nonzero blocks. Therefore, we can see the k-sparse vector x as k-block sparse. In the runtime phase, first, we can estimate the macroblocks occupied by the devices by solving the optimization problem

$$\min \|z - \Psi' x\|_2 \qquad \text{s. t. } \|x\|_{2,0} \leq k \tag{3}$$

where $\|x\|_{2,0} = |\{i \in \{1, \ldots, N\} : \|x[i]\|_2 > 0\}|$. The problem (3) can be solved via a modification of the Orthogonal Matching Pursuit for the sparse vectors [10]. Second, if $x^\star$ is the solution of (3) and $\ell^\star$ the index such that $x^\star[\ell^\star] \neq 0$, the cell occupied by the device in the selected macroblock is estimated by solving (2), reducing the dictionary to the columns of $\Psi'[\ell^\star]$ and applying the Orthogonal Matching Pursuit (OMP) algorithm [14]. We choose OMP since it is a greedy algorithm and it computes a number of iterations depending on the sparsity and thus it is much faster than a convex optimization routine, such as the interior point or the simplex method [15].

C. Dictionary design via block-coherence

We now introduce the notion of block-coherence of a dictionary, which we will use to provide theoretical guarantees for our method.

The *block-coherence* μ_B of a dictionary Θ measures the similarity between dictionary elements belonging to different blocks, and is defined as

$$\mu_B(\Theta) := \frac{1}{T} \max_{i \neq j} \rho(\Theta[i]^\top \Theta[j]),$$

where $\rho(\Theta[i]^\top \Theta[j])$ is the spectral norm of the matrix $\Theta[i]^\top \Theta[j]$. Conversely, the *sub-coherence* is a measure of the correlation of the atoms belonging to the same block, and is defined as

$$\nu(\Theta) := \max_i \min_{j \neq h} \left| \theta[i]_j^\top \theta[i]_h \right|,$$

where $\theta_j[i]$ denotes the j-th column of the i-th block. Theorem 3 in [10] provides a sufficient condition for successful recovery in absence of noise. More precisely, let $x \in \mathbb{R}^{LW}$ be a block k-sparse vector (*i.e.*, $\|x\|_{2,0} \leq k$) and $y = \Theta x$, where $\Theta \in \mathbb{R}^{N \times LW}$. If the following inequality is satisfied

$$kL < \frac{1}{2} \left(\frac{1}{\mu_B(\Theta)} + L - (L-1) \frac{\nu(\Theta)}{\mu_B(\Theta)} \right), \tag{4}$$

then x can be exactly recovered from y. In practice, this condition requires a small correlation between the atoms belonging to the same block (*i.e.*, a small value of $\nu(\Theta)$), and a small correlation between the different blocks (*i.e.*, a small value of $\mu_B(\Theta)$).

In this paper, we are going to compare our approach with the classical Nearest Neighbor [12], which can localize only one device. Thus, from now on, we assume $k = 1$.

IV. Simulations

We now present the results of some numerical simulations, which are useful to test the probability that a dictionary satisfies the block-coherence condition and the behavior of the theoretical model for different level of noise.

A. Probability of block-coherence condition

We performed a Monte Carlo simulation (over 10^4 runs) in order to estimate the probability that a dictionary Ψ' verifies the block-coherence condition in (4). We remark that the block coherence condition for exact reconstruction holds only in the noise-free case, and no guarantees are given for the noisy case. However, we expect the estimation in (3) to be robust against bounded noise (see [16]): small perturbations in the observations should cause small perturbations in the reconstruction.

We build Ψ exploiting the RSS model described in Section II and we compute a new dictionary obtained by reorganizing the columns of the original matrix Ψ, as described in Section III. More precisely, we set an area of 30m $\times$ 15m, on which we build a 12×6 grid, with square cells whose side is 250 cm. We deploy $J = 8$ BSs randomly in the grid and at each BS we take $T = 5$ measurements. The dictionary Ψ is built in a noise-free case, following the model (1), where we set $P_t - PL = -50$ and $n = 2.5$. We choose several partitions of the ground floor, corresponding to different choices of the block size. For each of these choices we evaluate the condition in (4).
The estimated probability for our block-based approach is reported in Table I.

2×1	2×2	3×2	3×3	4×3	6×3	6×6
1.00	0.98	0.80	0.62	0.46	0.11	0.05

TABLE I
EMPIRICAL PROBABILITY FOR VALIDITY OF (4)

We observe that the best probability is obtained for blocks of size 2×1. This happens because the value of the sub-coherence ν increases as the block size increases, therefore the best value of ν is reached when the block size is the smallest considered. This could suggest the choice of blocks of size 1, which leads to a ν equal to 0. However, we expect that a large block size increases the robustness to noise, since a large number of atoms of the dictionary is exploited in the approximation of the measurement vector z. Therefore, in the designing of an optimal dictionary for localization we have to find a good trade off between these two aspects.

B. Simulation varying the noise level

We now propose some numerical simulations to test our algorithm, comparing it to WkNN [12] and to existing sparse methods [6].

We built the dictionary Ψ as in Section IV-A, then, for the runtime phase we generate the real-time measurements using 100 different values of standard deviation $\sigma \in (10^{-3}, 10)$, performing 200 experiments for each value of σ.

The results are shown in Figure 4. Our block-based method with blocks of 2×1 and 2×2 cells (green and red lines respectively) performs better than classical methods such as WkNN (black line) and spatial sparsity techniques (blue line): it always recovers the exact position when $\sigma < 0.1$. As expected (see Table I), the cases with 2×1 and 2×2 block-size cells outperforms the 3×2 case (orange line), which in turn is better than the spatial sparsity method when the noise-level is not too high.

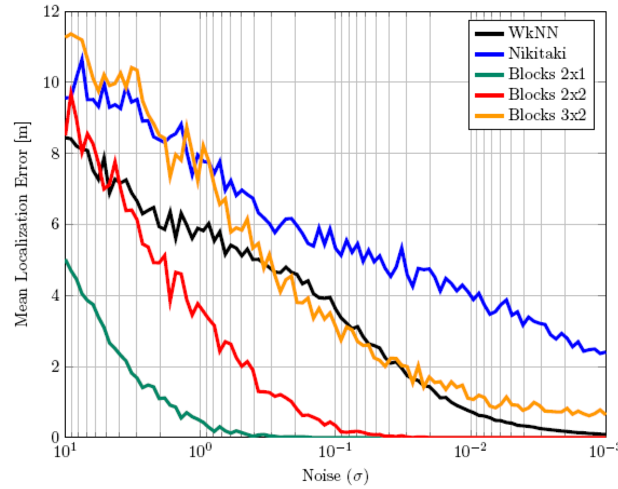


Fig. 4. Comparison of different approaches in a simulated case.

V. REAL EXPERIMENTS

In this section, we show the results of real experiments performed in two different scenarios. The experiments are carried out with SPIRIT1 evaluation boards (frequency band 915 MHz), produced by ST Microelectronics [17]. Optimized for low-power operation [18], SPIRIT1 is a radio frequency transceiver that provides digital output of the RSS for the last received packet or sequence of symbols. The firmware running on the devices has been developed from the Thingsquare [19] open source code, taking advantage of its Contiki operating system [20] core for the programming model and for the networking support.

Experiment 1: Garden area in the university campus. We consider a 4×6 grid with 3m-side cells in a 12m $\times$ 18m open air area. We displace all $J = 7$ BSs as shown in Figure 5. At each BS we collect $T = 3$ samples of RSS both in the training and in runtime phase.

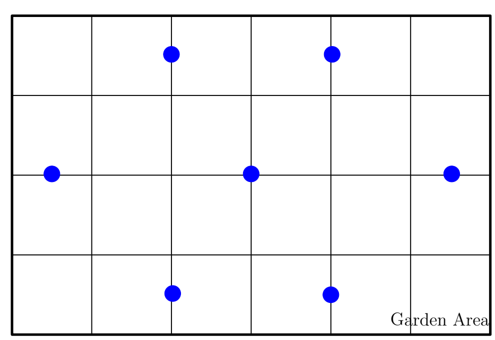


Fig. 5. Outdoor scenario: 7 BSs in a 12m $\times$ 18m grid.

The results are shown in Figure 6, where we illustrate the empirical cumulative distribution function ($\text{CDF}(x) = \mathbb{P}(X \leq x)$) of the localization error our block-sparsity method, the spatial sparsity method [6], and the well-known WkNN [12].

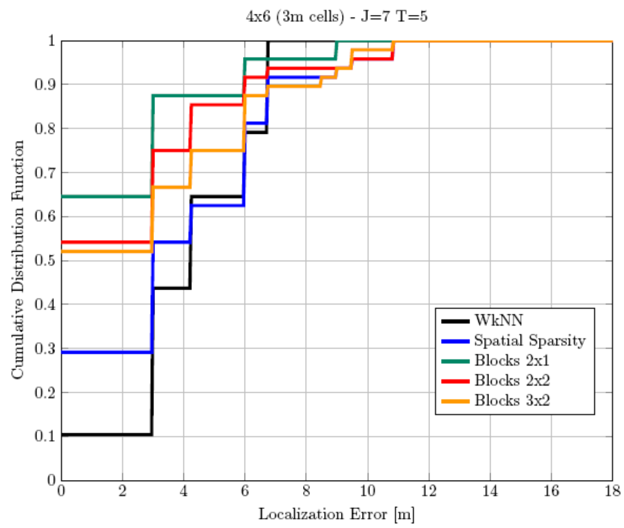


Fig. 6. CDFs in outdoor scenario: a localization error of 0m corresponds to exact localization, 3m means localization in an adjacent cell, 4.24m in a diagonal cell, and so on.

On a sample of 48 experiments, we obtained a significant reduction of the localization error with respect to WkNN and spatial sparsity methods. In about 65% of experiments we obtain perfect localization (i.e. exact identification of the cell occupied by the device) for the 2×1 block-size case (green line) and between 55% and 50% for 2×2 and 3×2 cases (red and orange curves respectively), while only 30% and 10% for CS-based approach [6] and WkNN, respectively.
Moreover for more than 85% (against 55% of CS-based approach and 43% of WkNN) we have a localization error lower than 3m, which means we have localized the device at most in the adjacent cell.

Experiment 2: Household scenario. We now consider an indoor household scenario. In Figure 7 we can see the house plan (the walls dividing the rooms are emphasized by thicker darkred lines). In each room we consider 4 cells and deploy $J = 7$ BSs (blue dots) in the middle, except for hallway and bathrooms. At each BS we take $T = 3$ measurements both in training and runtime phases. We run 60 experiments.

As depicted in Figure 8, the proposed algorithm achieves the best performance with 2×1 and 2×2 blocks (green and red lines). In this case we have a better performance of 2×2 case compared to the outdoor scenario, since the blocks correspond to the rooms. These two cases recover the exact position in over 60% of the experiments, while for competing algorithms [6] (blue line) and [12] (black line) the percentage of success is below 25%. Moreover, in the 80% of the experiments our approach localizes the device in the right room.

Finally, for what concerns the computational time of the runtime phase, we report the mean, maximum and minimum values of 108 experiments for the considered recovery algorithms in Table II. The tests are performed using MATLAB® R2014a on a HP® Z230 Tower Workstation Intel® Xeon® processor E3-1225 v3, 3.2GHz, with 4 cores and 32GB RAM.

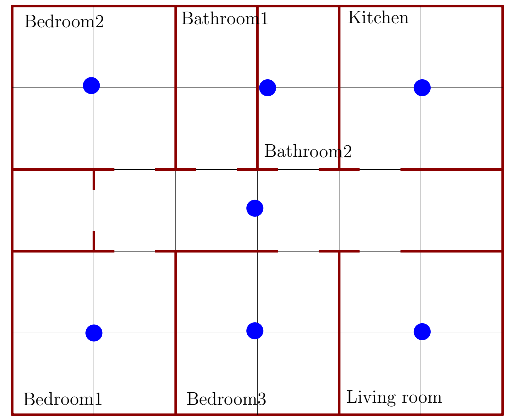


Fig. 7. Household scenario: 7 BSs in a 12.5m $\times$ 15m grid.

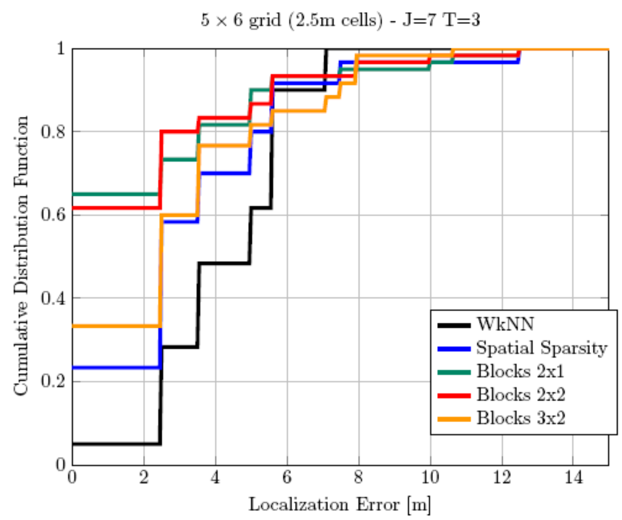


Fig. 8. CDFs in household scenario: a localization error of 0m corresponds to exact localization, 2.5m means localization an adjacent cell, 3.53m in a diagonal cell, and so on.

	Nearest Neighbor	Spatial Sparsity	Block Sparsity
mean	1.0587e-04 s	2.6059e-2 s	1.5421e-3 s
max	1.2000e-04 s	2.7455e-2 s	1.8580e-3 s
min	1.0100e-04 s	2.5830e-2 s	1.1960e-3 s

TABLE II
MEAN, MAXIMUM AND MINIMUM COMPUTATIONAL TIME OF 108 TESTS FOR THE CONSIDERED APPROACHES USING MATLAB® R2014A ON A HP® Z230 TOWER WORKSTATION INTEL® XEON® PROCESSOR E3-1225 V3, 3.2GHZ, 4 CORES, 32GB RAM.

The results show that the fastest algorithm is the WkNN, which computes a scalar product of the runtime measurements with the columns of the training matrix. However, our method based on block sparsity is also fast since the occupied position is recovered in the order of milliseconds thanks to the greedy algorithm involved, while the convex optimization of spatial sparsity method [6] is 20 times slower.

VI. Concluding Remarks

In this paper, we have introduced a new methodology for RSS-fingerprinting localization in wireless sensor networks based on block-sparsity, and we have proved that it performs better than state-of-the-art techniques through several numerical simulations and real experiments, deploying sensors in different indoor and outdoor scenarios. We achieve a precise localization in 65% of the experiments, while only 30% and 10% are the percentage of success for the considered literature methods.

As future work, we will investigate the mathematical properties of the dictionary Ψ, which is a low-rank incoherent matrix, and whether the position of BSs could influence the results.
Moreover, in this paper we have presented the localization of one device, but this can be extended to $k > 1$ devices [21]. Thus, we are going to set up both simulated and real experiments localizing two or more devices simultaneously, saving time and hopefully approaching WkNN computational performance, since this method can localize only one device at a time.
Finally, we will try to perform localization in a distributed way, where the BSs exchange messages among them and recover the occupied position without the need of a central unit [22], [23], [24].

References

[1] J. Bachrach and C. Taylor, "Localization in sensor networks," *Handbook of sensor networks: Algorithms and Architectures*, vol. 1, 2005.

[2] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern., Syst., Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, 2007.

[3] A. Boukerche, H. Oliveira, E. Nakamura, and A. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Commun.*, vol. 14, no. 6, pp. 6–12, 2007.

[4] J. Wang, R. Ghosh, and S. Das, "A survey on sensor localization," *J. Control Theory Appl.*, vol. 8, no. 1, pp. 2–11, 2010.

[5] Z. Farid, R. Nordin, and M. Ismail, "Recent advances in wireless indoor localization techniques and system," *J. Comput. Netw. Commun.*, 2013.

[6] S. Nikitaki and P. Tsakalides, "Localization in wireless networks via spatial sparsity," in *Asilomar Conference on Signals, Systems and Computers*, 2010, pp. 236–239.

[7] ——, "Localization in wireless networks based on jointly compressed sensing," *Proc. of IEEE EUSIPCO*, vol. 11, 2011.

[8] ——, "Decentralized indoor wireless localization using compressed sensing of signal-strength fingerprints," in *Proc. of 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, 2012, pp. 37–44.

[9] Y. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 5302–5316, 2009.

[10] Y. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Signal Process*, vol. 58, no. 6, pp. 3042–3054, 2010.

[11] C. Feng, W. Au, S. Valaee, and Z. Tan, "Received-signal-strength-based indoor positioning using compressive sensing," *IEEE Trans. Mobile Comput.*, vol. 11, no. 12, pp. 1983–1993, 2012.

[12] B. Li, J. Salter, A. G. Dempster, and C. Rizos, "Indoor positioning techniques based on wireless LAN," in *Proc. of 1st IEEE International Conference on Wireless Broadband and Ultra Wideband Communications*, 2006.

[13] A. Bay, P. Fragneto, M. Grella, S. M. Fosson, C. Ravazzi, and E. Magli, "Sparsity-based indoor localization in wireless sensor networks," in *15th International Workshop on Multimedia Signal Processing*, 2013.

[14] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[15] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge, UK: Cambridge university press, 2004.

[16] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 6–18, 2006.

[17] http://www.st.com/web/catalog/sense_power/FM1968/CL1976/SC1845/PF253167.

[18] http://www.st.com/web/en/catalog/tools/PF259026.

[19] http://thingsquare.com/.

[20] http://www.contiki-os.org/ .

[21] H. Shen, Z. Ding, S. DasGupta, and C. Zhao, "Multiple source localization in wireless sensor networks based on time of arrival measurement," *IEEE Trans. Signal Process.*, vol. 62, no. 8, pp. 1938–1949, April 2014.

[22] N. Kantas, S. Singh, and A. Doucet, "Distributed maximum likelihood for simultaneous self-localization and tracking in sensor networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5038–5047, 2012.

[23] A. Simonetto and G. Leus, "Distributed maximum likelihood sensor network localization," *IEEE Trans. Signal Process.*, vol. 62, no. 6, pp. 1424–1437, 2014.

[24] F. Yin, C. Fritsche, D. Jin, F. Gustafsson, and A. Zoubir, "Cooperative localization in WSNs using Gaussian mixture modeling: Distributed ECM algorithms," *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1448–1463, 2015.