A Diffusion Approach to Network Localization

Yosi Keller and Yaniv Gur

***Abstract*—The localization of nodes on a network is a challenging research topic. It arises in a variety of applications such as communications and sensor network analysis. We propose a computational approach to recovering the positions of network nodes given partial and corrupted distance measurements, and the positions of a small subset of anchor nodes. First, we show how to derive geometrically adaptive diffusion bases defined over the entire network, given only partial distance measurements. Second, we propose to utilize several diffusion bases simultaneously to derive multiscale diffusion frames. Last, we utilize the diffusion frames to formulate a L_1 regression based extension of the anchor points coordinates to the entire network. We experimentally show that under a wide range of conditions our method compares favorably with state-of-the-art approaches.**

***Index Terms*— Graph theory, machine learning, wireless sensor networks.**

I. Introduction

THE SENSOR NETWORK LOCALIZATION (SNL) problem receives a growing interest over the last few years due to its applicability to a gamut of fields such as wireless networks [33] and environmental monitoring (e.g., [2]). A sensor network is comprised of a large number of sensors scattered at positions $S = \{\mathbf{x}_i\}_1^n$ over a large area, where each sensor communicates with its neighbors within a limited sensing range R_0. The locations are often geographically pseudo-uniform and unknown *a priori.* For instance, consider a set of sensors dispensed from an aircraft over a disaster-struck area. The sensing range limitation R_0 is due to energetic constraints induced by the small size and price of each sensor. Hence, although such sensors can be made location-aware by recovering their coordinates independently using a GPS receiver, that would make each sensor pricier and larger in size. For a large number of sensor-types, this approach might prove too expensive and energy consuming. Thus, there is a growing need for fast and efficient algorithms to localize all the sensors in the network.

The SNL problem is to localize these sensors given the $K \ll n$ positions of some of the sensors denoted as *anchors*, and a noisy estimate of a small subset (down to 5%) of the pairwise Euclidean distances $\{d_{ij}\}$ between each sensor and some of its neighbors. The common distance measurement model [5], [7], [8], [15], [35], [36], [38] is given by

$$d_{ij} = d_{ij}^0(1+f), f \sim N\left(0, \sigma_n^2\right) \tag{1}$$

where d_{ij}^0 is the true (unknown) distance

$$d_{ij}^0 = \|\mathbf{x}_i - \mathbf{x}_j\|_2 \tag{2}$$

and negative noisy distance values $d_{ij} < 0$ are truncated to zero. This measurement model corresponds to RF based range measurements and its derivation is detailed in [28]. Due to energetic constraints, we are only given short distances for which $d_{ij} < R_0$. Equivalently, we are given a sparse noisy distance matrix (d_{ij}) and aim to recover its missing entries. We further sparsify the distance matrix by retaining Q edges at most for each node. In some setups [15] $Q = \infty$, implying that the edges to all nodes within R_0 are used.

Let $\{\mathbf{x}_i\}_1^n$ be the set of estimated sensor locations. In the SNL problem we aim to recover $\{\mathbf{x}_i\}_1^n$ such that their distances adhere to the noisy measured distances. One possible penalty function for the distance discrepantly is given by

$$\{\mathbf{x}_i\}_1^n = \arg\min_{\{\mathbf{x}_i^*\}_1^n} \sum_{i,j \ s.t. \ d_{ij}>0} \left(\left\|\mathbf{x}_i^* - \mathbf{x}_j^*\right\|_2^2 - d_{ij}^2\right)^2 \tag{3}$$

where (3) is evaluated over the subset $\{d_{ij}\}_{d_{ij}>0}$ of measured noisy distances. Biswas *et al.* [7] relaxed (3) into a SDP problem and also proposed [6] to use

$$\{\mathbf{x}_i\}_1^n = \arg\min_{\{\mathbf{x}_i^*\}_1^n} \sum_{i,j \ s.t. \ d_{ij}>0} \left|\left\|\mathbf{x}_i^* - \mathbf{x}_j^*\right\|_2^2 - d_{ij}^2\right|. \tag{4}$$

Another variation of the penalty function is given by

$$\{\mathbf{x}_i\}_1^n = \arg\min_{\{\mathbf{x}_i^*\}_1^n} \sum_{i,j \ s.t. \ d_{ij}>0} \left(\left\|\mathbf{x}_i^* - \mathbf{x}_j^*\right\|_2 - d_{ij}\right)^2 \tag{5}$$

and solved by the stress majorization algorithm (SMACOF) [9].

In practice, the noisy input distances d_{ij} between two different sensor nodes can be determined using approaches such as time of arrival (TOA) and time difference of arrival (TDOA). Some works deal with the *anchor-free* formulation (e.g., [34]), where no anchors are given.

In this work we aim to solve the location-aware SNL problem by formulating it as a regression problem over adaptive bases we denote *diffusion bases*. We model the network as a graph and compute the Diffusion Maps embedding [13], whose spectral embedding vectors constitute a geometrically adaptive basis. We extend previous results by Naoki Saito [30] that showed how to derive diffusion bases over generalized domains. But, in the SNL problem we are only given a sparse subset of the graph, consisting of less than 5% of the graph edges, on average. Yet, we show that this subset suffices to approximate the adaptive

Manuscript received December 11, 2010; accepted February 19, 2011. Date of publication March 03, 2011; date of current version May 18, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Lawrence Carin.

Y. Keller is with the School of Engineering, Bar Ilan University, Ramat Gan, Israel (e-mail: yosi.keller@gmail.com).

Y. Gur is with the SCI Institute, University of Utah, SLC, UT 84112 USA (e-mail: yaniv.gur1@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2011.2122261

bases computed using the entire graph (network). The spectral regression is followed by steepest descent refinement. The proposed scheme is shown to outperform state-of-the-art methods, when the noise is significant. We apply recent results in random graph theory to justify the robustness of our scheme and interpret the crossover phenomenon it exhibits.

The paper is organized as follows: we review previous results on the SNL problem in Section II and discuss the Diffusion Framework for dimensionality reduction in Section III. We continue to present Naoki Saito's [30] work on Diffusion based basis construction in Section IV, and show how to approximate them in the SNL problem. The proposed SNL scheme is derived in Section V, and we analyze and discuss its properties in Section VI. It is experimentally verified and compared to contemporary state-of-the-art schemes in Section VII, while concluding remarks and future extensions are discussed in Section VIII.

II. Background

Various solutions to the SNL problem have been proposed over the last few years [3], [7], [16], [17], [23], [27], [28], [31], [32], [34]. When *all* of the n^2 distances between the sensors are known, aunique solution (up to a rigid transformation) can be computed by Classical Multidimensional Scaling (MDS) [9]. The MDS minimizes (3) directly. But, when only a fraction (typically 3%–5%) of the distances is given, this problem becomes nonconvex and a more elaborate scheme is required. The algebraic properties of full (non-sparse) Euclidean Distance Matrices (rank, eigenspace) were analyzed by Gower [19], [20] who pointed out some connections between these properties and the configurations of points that generate the matrices.

Costa *et al.* [14] propose a distributed weighted-MDS approach to localization. They define a stress minimization objective function, that does not require the knowledge of all the pairwise dissimilarities. The distance measurements are weighted to reflect their accuracy, such that less accurate measurements are down-weighted. The resulting cost function is minimized iteratively, while the weights are also iteratively updated.

Anchor-free localization was discussed by Gao *et al.* [22], [37]. They show that one of the fundamental difficulties in Anchor-free localization is to avoid flip ambiguities, as parts of the network may fold on top of the others, thus, distorting their constructed network's topology. For that, Gao *et al.* propose to compute the combinatorial Delaunay complex on a selected subset of landmarks nodes in the sensor network [37]. These are used to recover the sensor network's shape by positioning the anchor nodes on its boundary. An improved rigid Delaunay complex can be computed by incrementally selecting landmark nodes.

A novel approach to anchor-free localization was suggested by Amit Singer [34]. In this work the localization is computed by agglomerating the results of a set of locally rigid embeddings, one per sensor node. The agglomeration is achieved by deriving a global weight matrix W that describes the relations between the different local embeddings. It is shown that the eigenvectors of W, corresponding to the eigenvalues $\lambda = 1$, are the coordinates of the network nodes. Due to the measurements noise the coordinates are given by linear combinations of the eigenvectors.

A related approach was suggested by Zhang *et al.* [38] who presented the *As-Rigid-As-Possible* (ARAP) SNL approach. They start by localizing small patches using either SDP or the stress minimization approach of Gotsman and Koren [18]. The 1-hop neighborhood of each node is rigidly reconstructed and denoted a patch. This better preserves the local relationships between patches and results in a system of sparse nonlinear equations that is solved by least-squares. They also suggest the *As-Affine-As-Possible* (AAAP) algorithm. The ARAP was experimentally shown to be robust to sparsity and noise.

The initial localization of small patches was also used by Cucuringu *et al.* [15]. The reconstructed patches are related to the global network via inversions, rotations and translations. Rather than recovering the set of all transformation parameters for all patches simultaneously, as in the ARAP [38] approach, Cucuringu *et al.* decompose this problem and estimate each of the transformations separately, yielding a robust anchorless SNL scheme.

A different approach to agglomerating the local localizations is the robust quadrilaterals method of Moore *et al.* [24], where quadruples of nodes are first localized and then iteratively merged. Care is taken such that the local localization of the quadruples is geometrically robust. Hence, in contrast to the work of Singer [34], the agglomeration step is iterative and greedy, which might result in accumulations of reconstruction errors.

A Kernel based pattern recognition approach was proposed by Nguyen *et al.* [26]. This is a location aware method that utilizes an input set of nodes with known positions, and the distances between all of the nodes. The localization consists of two stages. In the first, denoted as *coarse-grained localization*, the domain is divided into subregions and each node is classified as belonging (or not) to a particular region. This decision is derived using a SVM classifier. In the second stage (*fine-grained localization*), the node is localized as the weighted average of the locations of the nodes in its region whose positions are known.

Many of the recent solutions involve semidefinite programming (SDP) relaxation [5], [8]. These algorithms use SDP relaxation to estimate the locations of the sensors, where in some cases the estimation is improved by using a refinement step that is based on gradient descent methods. This refinement step cannot be applied directly to the original SNL problem as it is nonconvex. The works by Biswas *et al.* [5], [8] are of particular interest, as they combine the SDP solver with the use of anchor nodes.

However, the SDP relaxation has its downsides. First, in SDP the rank of the solution is higher than the dimension of the original problem, where the dimensionality reduction process leads to high estimation error. This problem may be solved by using a rank constraint such as the regularization term that was proposed in [5]. Since these rank constraints are nonconvex, it is not guaranteed that the SDP relaxation will eventually converge to the low dimensional solution. Second, solving the SDP problem on very large matrices, or subject to many constraints, is computationally expensive.

Saul *et al.* [36] propose an anchor-free SNL scheme thatsolves the first problem by using the maximum variance unfolding (MVU) technique. They show that low rank solutions

emerge naturally by computing the maximal trace solutions that respect the local distances, and can thus avoid the need for explicit rank constraints. Practically, this is done by maximizing the trace of the distance matrix which is defined via inner products between the locations, $(\Delta_{ij}) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. This leads to maximizing the variance of the low dimensional data representation. In the same paper, a solution to the second problem was suggested by factorizing the $n \times n$ distance matrix Δ such that $\Delta \approx QYQ^T$, where Q is of size $n \times m$, and where $m \ll n$. Thus, instead of operating on a large matrix Δ, the SDP is solved for the smaller matrix Y of dimension $m \times m$. This leads to a fast and accurate approximation of the solution to the original problem. This approximation is used to initiate a refinement step using a conjugate gradient descent method.

A distributed SNL approach was proposed by Khan *et al.* [21]. The core of their approach is to select a minimal number of anchor nodes such that the network could be localized. They assume that the sensing range R_0 can be enlarged to guarantee a certain triangulation, thus only three anchor nodes are needed to localize all sensors in the 2D plane. They introduce the Distributed Iterative LOCalization algorithm (DILOC) that is distributed and iterative with a guaranteed convergence. Each sensor performs local triangulation and passes the information to its neighboring sensors. Our approach differs as it is centralized and global.

The work of Coates *et al.* [11] applied diffusion wavelets to estimate network-related quantities. They represent the network as a graph based on the routing matrix, and aim to estimate the delay functions defined over a communications network. The input to their algorithm is the *entire* network and they show results on a network consisting of 11 and 30 nodes. This is due to the significant computational complexity required to compute the diffusion basis even for such small networks. In contrast, in our work we reconstruct the network itself, given a subset of its edges. Our schemes requires just a few seconds to reconstruct a network consisting of 1000 nodes, where only 5% of the connectivity is given.

The work of Gotsman and Koren [18] also utilizes the eigenvectors of the Graph–Laplacian. They formulate the SNL as a spectral graph drawing algorithm and utilize spectral embeddings. Their first step is to compute an initial estimate of the network locations by computing the two leading eigenvectors of the graph corresponding to the input distances. This is based on a distributed Laplacian eigenvector computation. In the second stage the result of the first stage is refined by iteratively minimizing the discrepancy in distance measurements. The eigenvectors are computed in this framework using a distributed computational algorithm.

Our work, as well of most of the ones mentioned above, deals with abstract localization problems, where one is given sparse noisy distance measurements. Hence, the works of Patwari *et al.* [28], [29] are of particular interest as they derive statistical models for actual range measurement systems based on wireless sensor networks [TOA, angle-of-arrival (AOA), and received-signal-strength (RSS)]. They model the probability density function of d_{ij} with respect to the real distance d_{ij}^0 as a lognormal distribution [29], and propose a probabilistic localization scheme denoted bias-reduced maximum likelihood.

III. The Diffusion Framework

In this section, we recall the diffusion framework as described in [13]. Let $\Omega = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of n data points, such that $\mathbf{x}_i \in \mathbb{R}^d$. We view the points Ω as being the nodes of an undirected graph with symmetric weights, in which any two nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected by an edge that is quantified by the affinity between $\mathbf{x}_i$ and $\mathbf{x}_j$, $w(\mathbf{x}_i, \mathbf{x}_j)$, which is application-specific. In cases where each data point is a point in a Euclidean feature space, the affinity can be measured in terms of closeness in that space, and it is common to weight the edge between $\mathbf{x}_i$ and $\mathbf{x}_j$ by

$$w(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\varepsilon^2\right) = \exp(-d_{ij}^2/\varepsilon^2), \quad (6)$$

where $\varepsilon > 0$ is a scale parameter. This choice of weight corresponds to the notion that local distance measurements are the only ones relevant. Thus, two points $\mathbf{x}_i$ and $\mathbf{x}_j$ will have nonzero affinity $w(\mathbf{x}_i, \mathbf{x}_j)$ if their distance $d_{ij} < 3\varepsilon$, and the kernel bandwidth ε allows to set the notion of closeness.

Belkin and Niyogi [4] showed that in the case of a data set approximately lying on a submanifold, this choice corresponds to an approximation of the heat kernel on the submanifold, while Coifman and Lafon [13] proved that any weight of the form $h(\|\mathbf{x}_i - \mathbf{x}_j\|)$ (where h decays sufficiently fast at infinity) allows to approximate the heat kernel.

The weight function or kernel describes the first-order interaction between the data points as it defines the nearest neighbor structures in the graph. It should capture a notion of affinity as meaningful as possible with respect to the application, and therefore could very well take into account any type of prior knowledge on the data.

The diffusion framework has an elegant probabilistic interpretation that paves the way for the spectral decomposition scheme. We induce a random walk on the data set Ω by forming the following kernel:

$$p_1(\mathbf{x}, \mathbf{y}) = \frac{w(\mathbf{x}, \mathbf{y})}{d(\mathbf{x})}$$

where $d(\mathbf{x}) = \sum_{z \in \Omega} w(\mathbf{x}, \mathbf{z})$ is the degree of node $\mathbf{x}$.

As $p_1(\mathbf{x}, \mathbf{y}) \geq 0$ and $\sum_{y \in \Omega} p_1(\mathbf{x}, \mathbf{y}) = 1$, the quantity $p_1(\mathbf{x}, \mathbf{y})$ can be interpreted as the probability for a random walker to jump from $\mathbf{x}$ to $\mathbf{y}$ in a single time step. If P is the $n \times n$ matrix of transition of this Markov chain, then taking powers of this matrix amounts to running the chain forward in time. Let $p_t(\mathbf{x}, \mathbf{y})$ be the kernel corresponding to the t^{th} power of the matrix P, thus, $p_t(\mathbf{x}, \mathbf{y})$ describes the probabilities of transition from $\mathbf{x}$ to $\mathbf{y}$ in t time steps. This Markov chain is governed by a unique stationary distribution $\boldsymbol{\phi}_0$ that is the top left eigenvector of P with eigenvalue $\lambda_0 = 1$, i.e., $\phi_0^T P = \phi_0^T$, and it can be verified that $\phi_0(\mathbf{y})$ is given by

$$\boldsymbol{\phi}_0(\mathbf{y}) = \frac{d(\mathbf{y})}{\sum_{z \in \Omega} d(\mathbf{z})}.$$

The preasymptotic regime (with respect to the Markovian time variable t) is governed according to the following eigendecomposition:

$$p_t(\mathbf{x},\mathbf{y}) = \sum_{l\geq 0} \lambda_l^t \boldsymbol{\psi}_l(\mathbf{x})\boldsymbol{\phi}_l(\mathbf{y}) \tag{7}$$

where $\{\lambda_l\}$ is the sequence of eigenvalues of P (with $\lambda_0 = 1$) and $\{\phi_l\}$ and $\{\psi_l\}$ are the corresponding biorthogonal left and right eigenvectors. Furthermore, because of the spectrum decay, only a few terms are needed to achieve a given relative accuracy $\delta > 0$ in (7).

Unifying ideas from Markov chains and potential theory, the *diffusion distance* between two points $\mathbf{x}$ and $\mathbf{z}$ was introduced in [13] as

$$D_t^2(\mathbf{x},\mathbf{z}) = \sum_{y\in\Omega} \frac{\left(p_t(\mathbf{x},\mathbf{y}) - p_t(\mathbf{z},\mathbf{y})\right)^2}{\boldsymbol{\phi}_0(\mathbf{y})}. \tag{8}$$

This quantity is a weighted L_2 distance between the conditional probabilities $p_t(\mathbf{x},\cdot)$, and $p_t(\mathbf{z},\cdot)$. These probabilities can be thought of as features attached to the points $\mathbf{x}$ and $\mathbf{z}$, that measure the interaction of these two nodes with the rest of the graph. The connection between the diffusion distance and the eigenvectors is given by

$$D_t^2(\mathbf{x},\mathbf{z}) = \sum_{l\geq 1} \lambda_l^{2t} \left(\boldsymbol{\psi}_l(\mathbf{x}) - \boldsymbol{\psi}_l(\mathbf{z})\right)^2. \tag{9}$$

Note that ψ_0 does not appear in the sum as it is a constant. This identity implies that the right eigenvectors can be used to compute the diffusion distance. Due to the spectrum decay, only a few terms are needed to achieve a given relative accuracy $\delta > 0$ in (9). Let $m(t)$ be the number of terms retained, and define the diffusion map

$$\Psi_t : x \longmapsto \left(\lambda_1^t \boldsymbol{\psi}_1(\mathbf{x}), \lambda_2^t \boldsymbol{\psi}_2(\mathbf{x}), \ldots, \lambda_{m(t)}^t \boldsymbol{\psi}_{m(t)}(\mathbf{x})\right)^T. \tag{10}$$

This mapping provides coordinates on the data set Ω, and embeds the n data points into the Euclidean space $\mathbb{R}^{m(t)}$. This method constitutes a universal and data-driven way to represent a graph or any generic data set as a cloud of points in a Euclidean space, and we derive a data parametrization that captures relevant modes of variability.

IV. Diffusion Bases and Frames

In this section we utilize the diffusion framework discussed in the previous section to derive data adaptive bases and frames adapted to the SNL problem. By *Diffusion basis* we refer to using the set of Diffusion embedding vectors (as defined in (7)) $\Psi_\varepsilon = \{\psi_l\}_1^L$ as a basis for a vector space. By construction, these eigenvectors are orthogonal and form a basis.

By using several Diffusion bases simultaneously with respect to the same vector space we derive a *Diffusion frame*. For instance, given a range of embedding bandwidth $S_\varepsilon = \{\varepsilon_m\}_1^M$, one can derive a corresponding set of embeddings

$$\{\Psi_{\varepsilon_m}\}_1^M = \{\Psi_{\varepsilon_1}, \ldots, \Psi_{\varepsilon_m}, \ldots, \Psi_{\varepsilon_M}\} \tag{11}$$

and the union of their eigenvectors forms a *Diffusion frame*.

While such constructions can be used in general to form multiscale diffusion embeddings, our work on the SNL problem was inspired by the results of Naoki Saito [30], who showed that the eigenfunctions of the Graph Laplacian can be used as basis functions for signals sampled on *nonuniform domains*. There are a gamut of data sources that are defined on generally shaped domains, on which one cannot define the conventional Fourier basis. In our approach the SNL problem is formulated as the estimation of the function of coordinates, defined over the network nodes that constitute a nonuniform two-dimensional domain.

Recalling that the (noise-free) Laplacian is a positive semidefinite (p.s.d) matrix, its eigenvectors (7) form an orthonormal basis, that can be used to expand functions defined on S, where its ambient geometry can be arbitrary [30]. In the SNL problem we are given the *noisy* distance measurements d_{ij} rather than a set of (possibly noisy) points Ω, as in the general Diffusion framework presented in Section III. Thus, the Graph-Laplacian we compute, is manifested by a noisy affinity matrix $A = (a_{ij})$

$$a_{ij} = \exp\left(-d_{ij}^2/\varepsilon^2\right). \tag{12}$$

that is an approximation of the true Laplacian over the domain. As such, the SNL problem differs from the one discussed by Saito [30] as there, the given manifold was noise-free. It follows that the resulting Graph Laplacian matrix is not guaranteed to be p.s.d.[1] Fortunately, due to Wigner's Semicircle Law [10], its eigenvectors approximate well those of the Laplacian computed over the noise-free domain. This is further discussed in Section VI.

The core of our SNL approach is to utilize the inherent similarity in the construction of the Laplacian and sensor networks. Both, utilize measurements (affinities) to close graph nodes, while the affinity between distant pairs is negligible. Namely, when computing the affinity using an exponential kernel as in (12), limiting the sensing range to R_0 is equivalent to setting $\varepsilon \approx \frac{R_0}{3}$

$$w_{i,j} = w(\mathbf{x}_i,\mathbf{x}_j) = \exp\left(-d_{ij}^2/\varepsilon^2\right) \approx 0, \quad \forall d_{ij} > R_0 \tag{13}$$

and the affinity computed using the partial data ($d_{ij} < R_0$) approximates the one computed with the full data set.

A function defined over a manifold might contain multiscale diffusion structures and thus a multiscale diffusion framework is required. For that we compute a Diffusion Frame as in (11), by computing a set of approximate Laplacians with varying bandwidths.

V. Sensor Localization by Spectral Regression

In order to solve the SNL problem we start by computing an adaptive diffusion basis $\Psi_\varepsilon = \{\psi_l\}_1^L$ or diffusion frame $\{\Psi_{\varepsilon_m}\}_1^M$ as in (11). These bases are both adaptive to the geometric network structure and defined over the entire domain. Hence, we propose to recover the location of the unknown nodes by considering each coordinate (x,y) as a *separate* function defined over the sensor network. Such functions can be estimated

[1]Communicated by one of the anonymous reviewers.

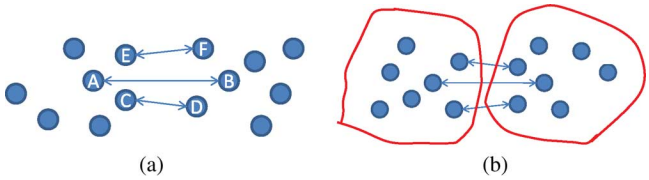

Fig. 1. Graph construction from direct distance measurements (a), and graph construction via patches (meta-nodes) in (b).

by linear regression using the diffusion bases/frames, and the given set of anchor points.

Using a set of K anchors points $S_a = \{x_k, y_k\}_1^K$, we fit a linear model to each coordinate using S_a. For instance, for the y coordinate

$$[\widehat{\boldsymbol{\psi}}_1 \quad \dots \quad \widehat{\boldsymbol{\psi}}_L]\mathbf{a} = \begin{bmatrix} y_1 \\ \vdots \\ y_K \end{bmatrix} \tag{14}$$

where the vectors $\{\widehat{\psi}_l\}_1^L$ are the L leading basis functions $\{\psi_l\}_1^L$ restricted to the set of anchors S_a. Namely, the elements of $\{\psi_l\}_1^L$ corresponding to the anchor points, and $\mathbf{a} \in \mathbb{R}^L$ is the vector of mixture coefficients. Due to the decay of the eigenvalues of the Markov matrix representing the network, only a few basis function (usually less than 10) are used. As it is common to have $K = 20 \div 40$ anchor points, $L < K$ when using a single diffusion basis and the linear regression used to solve (14) is overdetermined and robust. When a diffusion frame consisting of M bases is used, we have $L \cdot M$ basis functions that might be of the same order as the number of anchor points K. This reduces the robustness of the regression and hence we use L_1 regression instead of the linear one

$$\begin{aligned} &\mathbf{a} = \arg\min_{\mathbf{a}^*} \|\mathbf{a}^*\|_{L_1}, \\ &s.t. \begin{bmatrix} \widehat{\boldsymbol{\psi}}_1^1 & \dots & \widehat{\boldsymbol{\psi}}_L^1 & \dots & \widehat{\boldsymbol{\psi}}_1^M & \dots & \widehat{\boldsymbol{\psi}}_L^M \end{bmatrix} \mathbf{a}^* \\ &\qquad = \begin{bmatrix} y_1 \\ \vdots \\ y_K \end{bmatrix}. \end{aligned} \tag{15}$$

The use of L_1 regression also provides a best-basis regression solution, where the coordinates are regressed by the eigenvectors that best fit the data. Thus, eigenvectors that do not fit the data well will be automatically discarded, providing an additional measure of robustness. It also implies that we do not have to fine-tune the bandwidth ε, as a range of ε values can be used.

Given the regression coefficients, $\mathbf{a}$, we estimate the coordinates over the entire domain by

$$\mathbf{y}^* = \begin{bmatrix} \boldsymbol{\psi}_1^1 & \dots & \boldsymbol{\psi}_L^1 & \dots & \boldsymbol{\psi}_1^M & \dots & \boldsymbol{\psi}_L^M \end{bmatrix} \mathbf{a}. \tag{16}$$

where $\mathbf{y}^*$ and $\{\psi_l\}_{\mathbf{1}}^L$ are defined over the *entire* network. The same method mutatis mutandis is applied to the x coordinates. We denote our approach as *Spectral Regression (SR)* and it is summarized in Algorithm 1.

Algorithm 1 Spectral Regression

1: Given the sets of anchors S_a, relative distances S_d, and kernel bandwidth S_ε
2: For each kernel bandwidth ε_m, compute the corresponding embedding $\Psi_{\varepsilon_m} = \{\psi_{\boldsymbol{l}}^m\}_{\mathbf{1}}^L$
3: Form an overdetermined equation set using the y-coordinates of the set of anchors S_a

$$\begin{bmatrix} \widehat{\boldsymbol{\psi}}_1^1 & \dots & \widehat{\boldsymbol{\psi}}_L^1 & \dots & \widehat{\boldsymbol{\psi}}_1^M & \dots & \widehat{\boldsymbol{\psi}}_L^M \end{bmatrix} \mathbf{a} = \begin{bmatrix} y_1 \\ \vdots \\ y_K \end{bmatrix} \tag{17}$$

4: Solve (17) using L_1 regression.
5: Compute the extended set of coordinates $\mathbf{y}^*$

$$\mathbf{y}^* = \begin{bmatrix} \boldsymbol{\psi}_1^1 & \dots & \boldsymbol{\psi}_L^1 & \dots & \boldsymbol{\psi}_1^M & \dots & \boldsymbol{\psi}_L^M \end{bmatrix} \tag{18}$$

6: Compute the extended set of coordinates $\mathbf{x}^*$ mutatis mutandis.
7: Use the computed coordinates $\{\mathbf{x}_i\}_1^n = \{\mathbf{x}^*, \mathbf{y}^*\}$ as an initial estimate for gradient based optimization scheme.

A. Refinement Procedure

We apply a steepest descent refinement to improve the localization computed in the regression step. For that we use the spectral regression result as an initial estimate. In contrast to the spectral regression step, this step utilizes both the x and y coordinates simultaneously and minimizes (3) directly. We use the same conjugate gradient method as in [36].

VI. Discussion and Analysis

The proposed scheme is based on the spectral embedding of a noisy distance matrix, that is a low-rank matrix, as the intrinsic manifold of the data described by the distances is a two-dimensional plane. The eigendecomposition of such matrices is known to be robust due to Wigner's Semicircle Law [10], which implies that adding noise to p.s.d. matrices (such as the affinity matrix) results in a semicircle shaped noise spectrum. As the noise increases, the semicircle noise spectrum moves towards the leading eigenvalues, and when the noise energy reaches a certain threshold, the semicircle crosses into the leading eigenvalues.

Thus, we expect to encounter a *crossover* phenomenon in the computation of the eigenvectors $\{\psi_l\}_1^L$, where below a certain level of noise energy, the noise does not influence the computation of $\{\psi_l\}_1^L$ as the eigenvalues corresponding to the noise are not among the L leading eigenvectors. However, beyond a certain threshold a crossover occurs and the noise eigenvectors are recovered among the L leading eigenvectors, causing our scheme to fail. This is a known phenomenon in spectral algorithms such as PCA [25] and spectral embedding, and is evident in our experimental results in Section VII and depicted in Fig. 3(c).

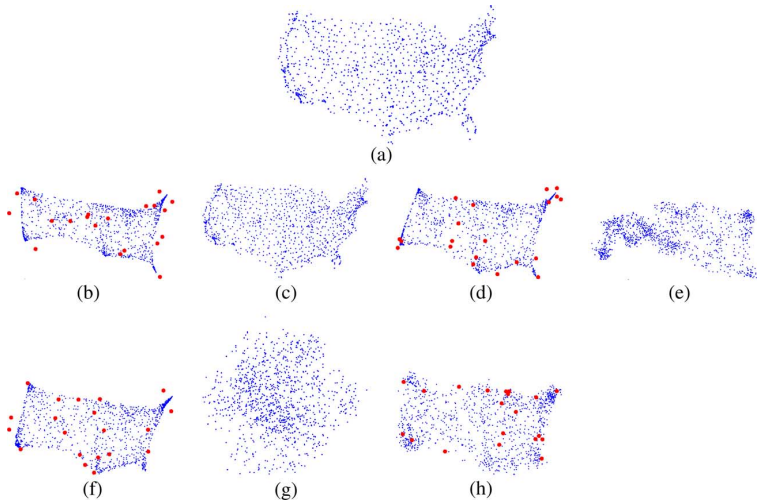

Fig. 2. The map of the US cities and its reconstructions for varying noise levels. The $K = 20$ anchors are marked by the red dots. For low levels of noise (Figs. (b) and (c) the MVU is superior, while for higher levels of noise [Figs. (d)–(h)] the MVU breaks down. The corresponding numerical error values are depicted in Fig. 3(a). The reconstructions are shown without the refinement step. (a) Original; (b) SR, $\sigma_n = 0$; (c) MVU, $\sigma_n = 0$; (d) SR, $\sigma_n = 0.2$; (e) MVU, $\sigma_n = 0.2$; (f) SR, $\sigma_n = 0.3$; (g) MVU, $\sigma_n = 0.3$; (h) SR, $\sigma_n = 1$.

Moreover, the affinity matrix used in our formulation, is a sparsified replica of the one computed given the entire domain, as it is common to limit the number of neighbors of each sensor (and the maximal rank of each node) to Q at most. Sparsification of p.s.d matrices, such as the affinity matrix was studied by Achlioptas and McSherry [1] that provide the mathematical foundations for approximating the spectrum of a matrix given its sparsified replica. The core idea is that sparsifying a large low dimensional matrix A is equivalent to adding a random matrix E whose entries are independent random variables with zero-mean and bounded variance

$$A_{\text{sparse}} = A + E.$$

The larger the sparsification the larger the variance of the elements of E. Achlioptas and McSherry utilize Wigner's Semicircle Law to show that the sparsification does not influence the computation of the leading eigenvalues and eigenvectors of A up to a certain threshold, where a crossover phenomenon occurs.

The leading eigenvectors of the Laplacian are the analogue of the low frequency basis functions on the Cartesian domain. Indeed, by applying the proposed scheme to a regular two-dimensional grid we would rediscover the two-dimensional Fourier basis. It follows that our approach captures the *global* structure of the graph (network), while its fine details can be recovered by steepest descent based refinement.

The eigenvectors satisfy the boundary condition $\frac{\partial \psi_l}{\partial n} = 0$, while the expanded functions, namely the x and y coordinates, do not satisfy this condition. This may lead to Gibbs-like phenomenon near the boundary of the domain.[2] A possible solution was suggested by Coifman [12] that noted that given an estimate of the localization, one can compute the normal components to the basis functions in the vicinity of the boundary, and add them as basis functions to the L_1 minimization in (15).

[2]Communicated by one of the anonymous reviewers.

The proposed scheme is essentially a data interpolation/extrapolation scheme for data points given on generalized domains. The SR scheme in Section V does not assume or utilize the specialized structure (rigidity) of the localization problem, as in the SDP based schemes, ARAP, ASAP, and the MVU. It only assumes a proper definition of the graph corresponding to the data, that is used for the interpolation phase, while the stress function ((3)–(5)) is only utilized in the refinement step. Thus, the SR can be applied to the interpolation of any function given on a graph in general, and on a network in particular. A possible application we aim to pursue in the future is the interpolation of sensor measurements, where the corresponding stress function will be related to the particular data. For instance, the stress function for temperature measurements might relate to physical conservation law or smoothness constraints.

Both the ARAP [38] and ASAP [15] schemes start by rigidly reconstructing the 1-hop neighborhood of each node. In our approach, [see Fig. 1(a)] the construction of the graph edge connecting the points (network nodes) A and B is based on *single* measured noisy distance d_{ab}. Using the reconstructed patches (Fig. 1(b)) we can better estimate the distance $\widehat{d}_{ab}$ based on the *multiple* measured distances between the corresponding patches, (d_{ab}, d_{cd} and d_{ef}), rather than a single measured distance d_{ab}. This implies that local reconstructions can be used to improve the computation of the graph representing the network. As both the ARAP and ASAP used $Q \approx 20$ 1-hop neighborhoods to reconstruct the patches, it follows by Fig. 1(b) that these are connected via an effective measurement radius $\widehat{R}_0 > R_0$. An approximate upper bound can be derived by assuming that the nodes within each patch are uniformly spread within a rectangular, each being R_0 away from its nearest neighbors

$$\widehat{R}_0 = 2\sqrt{K}R_0$$

and for $Q \approx 20$ we get $\approx 8.9R_0$. Thus, the effective sensing range of metanodes (patches) is larger, and the corresponding

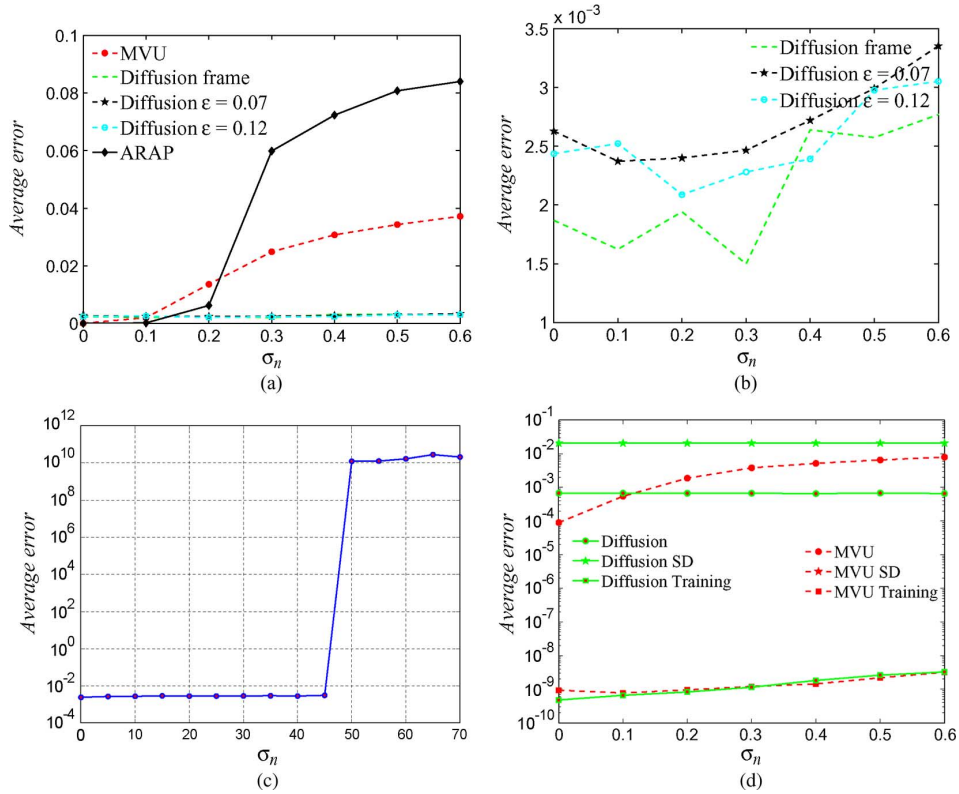

Fig. 3. The sensitivity of the reconstruction of the US map to varying degrees of noise. The accuracy rates are shown in (a) while we focus on the SR results in (b). (c) The reconstruction error of the SR over a wide range of noise levels. The reconstructions are shown without the refinement step. (d) The results of applying iterative steepest descent (SD) refinement to the SNL. The *Training* error is the average error over the given set of distances, while the *SD* error is the average error over the entire network.

graph would be based on more robust and longer range measurements.

VII. Experimental Results

We experimentally verified our proposed Spectral Regression (SR) scheme by applying it to simulated sensor networks used in previous works [15], [32], [35], [36]. Each network consists of the set $S = \{\mathbf{x}_i\}_1^n \in \mathbb{R}^2$ of points that simulates the sensors, and their noisy pairwise distances form the graph edges.

Using this graph, and a small number of K anchor points (nodes whose locations are known), we aim to infer the coordinates of all the nodes in this graph. In general, the number of anchor points is fixed, but, in one experiment we vary their number to test its influence on the localization. To test the performance of the algorithm on a noisy data set, we added various levels of nonconstant variance Gaussian noise to the original distance matrix, and used the noisy distance matrix as in (1). Hence, the regression we use is suboptimal (in the maximum-likelihood sense) for this noise model. This follows the experimental setup in previous works [5], [7], [8], [15], [35], [36], [38].

Let (d_{ij}^*) be the estimated distance matrix, then, the estimation error for a single experiment is measured by

$$\mathrm{MSE} = \frac{1}{n^2}\sum_{i,j}\left|d_{ij}^* - d_{ij}\right|^2 \tag{19}$$

where n is the number of nodes (sensors). Using this metric allows to compare our results against those of both anchor-based and anchor-free schemes.

We compare our results against those of the MVU based algorithm by Weinberger *et al.* [36], whose code was made public by the authors,[3] as this anchor-free approach utilizes spectral bedding. We also compared against the *FULLSDPD*, *ESDPD* and *ESDPDualD* SDP-based approaches of Wang *et al.* [35]. These schemes utilize anchor-nodes information and their code is publicly available, courtesy of the authors.[4] We also compared our results to the ARAP approach [38][5] that was shown to be the state-of-the-art in anchor-free localization.

Each experiment was repeated 200 times and the average error was calculated for each experimental setup. In each experiment we first computed a sparse and noisy distance matrix that was simultaneously used as input to *all* of the compared schemes. As the focal point of both the MVU and our scheme is the initial estimate of the network structure, and as both schemes use a similar refinement step, we report the localization results *without* the iterative refinement. When we also show the refinement results it is explicitly mentioned.

We start with the data set of $n = 1097$ major cities in the US that play the role of the nodes. This network is depicted in Fig. 2(a). We used $R_0 = 0.1$, $K = 20$ anchor points and

[3]http://www.cse.wustl.edu/kilian/Downloads/FastMVU.html.

[4]http://www.stanford.edu/yyye/.

[5]http://www.math.zju.edu.cn/ligangliu/CAGD/Projects/Localization/.

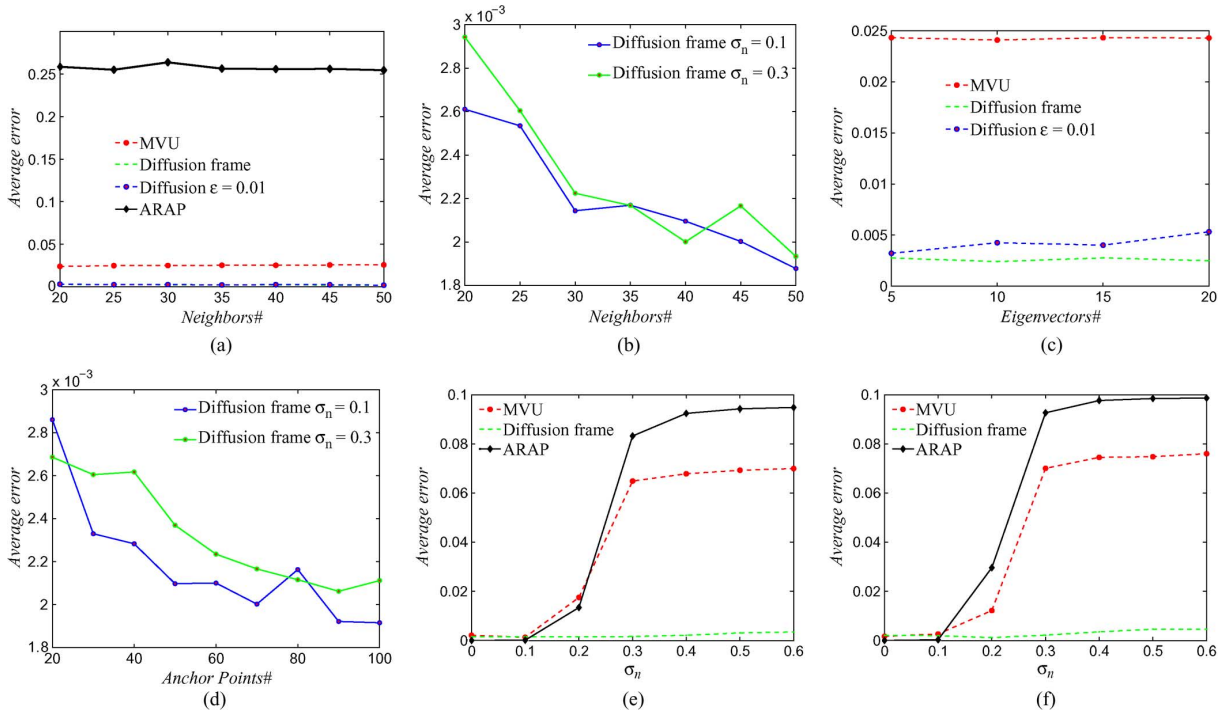

Fig. 4. The sensitivity of the reconstruction of the US cities map to different input parameters: (a) the maximal number of neighbors for each node within a radius of $R_0 = 0.1$ and $\sigma_n = 0.1$. (b) We repeat the analysis in (a) for the SR only and $\sigma_n = \{0.1, 0.3\}$.(c) The number of embedding eigenvectors. We also tested the stability to the number of anchor points in (d). In (e) and (f) we applied the different schemes with $R_0 = \{0.03, 0.05\}$, and average node rank of 39 and 18, respectively. (a) $R_0 = 0.1, \sigma_n = 0.3$; (b) $R_0 = 0.1$; (c) $R_0 = 0.1, \sigma_n = 0.3$; (d) $R_0 = 0.1$; (e) $R_0 = 0.05$; (f) $R_0 = 0.03$.

$Q = 30$ neighbors at most that resulted in ∼57 k non zero known distances (out of an ∼0.5 M edges). We were unable to run the SDP based algorithms due to memory constraints. We applied the MVU scheme using ten eigenvectors, as this was found to be optimal by Weinberger *et al.* [36]. For the proposed SR scheme we used $L = 5$ eigenvectors in all of our simulations (other than those in which we explicitly vary their number), and $M = 3$ bandwidths $\varepsilon_m = \{0.07, 0.12, 0.15\}$ for the Diffusion frame.

The reconstructed networks are depicted in Figs. 2(b)–(h) and the corresponding reconstruction error is shown in Fig. 3(a). For low noise levels the MVU and ARAP outperformed the SR providing a close to perfect reconstruction, even without the iterative refinement step. As the noise increases, the accuracy of the MVU reconstruction decreases rapidly for $\sigma_n = 0.2$ and the ARAP at $\sigma_n = 0.3$. In contrast, the reconstruction accuracy of the SR degrades gradually and provides reasonable results for $\sigma_n = 1$ [see Fig. 2(h)].

The reconstruction error computed according to (19) is shown in Fig. 3. It follows that for low noise values of $\sigma_n < 0.15$ the MVU and ARAP outperform the proposed scheme, but the proposed SR scheme was superior for $\sigma_n > 0.2$. The benefits of using the diffusion frame are depicted in Fig. 3(b), where for most noise values the Diffusion Frame is better than the single bandwidth schemes.

In Section VI we discussed the spectral properties of the Graph Laplacian and related the noise effect to Wigner's Semicircle Law. There we predicted that the proposed scheme will be robust to the noise, up to a certain amplitude of noise at beyond which it will completely fail due to the cross over of the eigenvalues. This is evident in Fig. 3(c).

Recalling that the proposed scheme and the MVU aims at recovering an initial estimate of the solution, to be used with an iterative refinement scheme (Section V-A), we tested the overall performance of the localization schemes using the iterative refinement procedure and the USA maps network. The results are reported in Fig. 3(d), where we first show the *Training error*. This is the average localization error of the refinement scheme, over the *given* set of distances. This error is directly minimized by the iterative refinement scheme, and it follows that for both schemes, it was on average of $O(10^{-9})$. As expected, the refinement schemes of both the SR and MVU converged to the same error due to the convexity of the localization problem given an initial estimate.

We further studied the sensitivity of the proposed SR scheme to its different input parameters. First we consider Q, the maximal number of edges per node these are depicted in Figs. 4(a) and 4(b), while the number of eigenvectors used for the adaptive spectral basis is tested in Fig. 4(c). The sensitivity of the reconstruction to the number of anchor points K is studied in Fig. 4(d). The sensitivity with respect to the sensing radius $R_0 = \{0.03, 0.05\}$ is studied in Figs. 4(e), 4(f), where the average node rank is 39 and 18, respectively. In these figures, we also compared our scheme and the MVU against the ARAP [38]. It follows that for low noise levels $\sigma_n < 0.1$ the ARAP outperforms both the MVU and SR. Yet, it should benoted that both the MVU and SR results are shown *without* the iterative refinement step, while the equivalent of the refine-

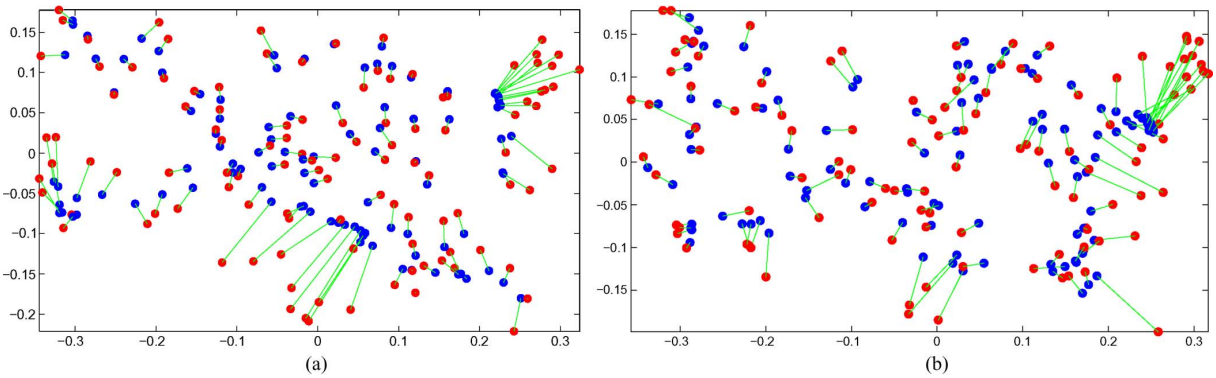

Fig. 5. Localization error visualization of the USA cities network. Corresponding true and localized nodes are connected. Note that the localization error increases as we approach the boundary. (a) $R_0 = 0.05, \sigma_n = 0$; (b) $R_0 = 0.05, \sigma_n = 0.3$.

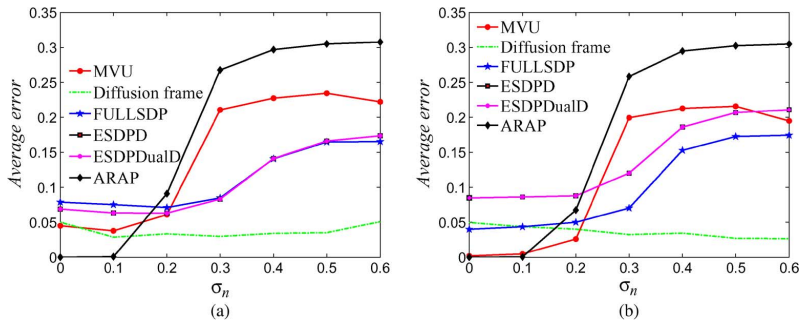

Fig. 6. Localization experiment using 500 and 1000 random points drawn randomly over $[-0.5, 0.5]^2$ and used in [35]. (a) 500 points; (b) 1000 points.

ment, the patch localization, is the first step of the ARAP. Both the ARAP and the MVU breakdown for $\sigma_n > 0.1$

We visualize the localization error in Fig. 5 for $R_0 = 0.03$ and $\sigma_n = \{0, 0.3\}$, by depicting 10% of the nodes. The localization errors increase as we approach the boundary. This follows the analysis in Section VI, where we predicted that a Gibbs-like phenomenon might occur near the boundary of the domain due to the corresponding boundary condition.

We repeated the localization experiments for the sets of 500 and 1000 uniformly spread random points over $[-0.5, 0.5]^2$ used in [35],[6] and also ran the *FULLSDPD*, *ESDPD* and *ESDPDualD* methods of Wang *et al.* [35]. The results xare depicted in Fig. 6.

We then tested our scheme by randomly varying the topology of the network. For that we drew 100 sets of 500 random points uniformly spread over the domain $[-0.5, 0.5]^2$. We added noise as before using $\sigma_n = \{0, 0.2, 0.4\}$. The experiment was repeated with ten different realizations of the noise (for each random topology/network and σ_n). The results are depicted in Fig. 7. First we limited the *maximal* rank of each node to $Q = 20$ and present the accuracy averaged over of manifestations of the network and noise (for a particular σ_n) in Fig. 7(a). The results are similar to those presented in Fig. 6. We also depict the histogram of average node rank in Fig. 7(b), where the distribution seems to be Gaussian-like, and thresholded at $Q = 20$. We repeated this experiment for a maximal node rank of 10. This is depicted in Figs. 7(c) and (d). In this case the distribution of node ranks is mostly concentrated around $Q = 10$, and while it differs from the histogram Fig. 7(b), the accuracy results (Figs. 7(a) and (c)) are similar.

We also considered a nonconvex domain as in [32] that is depictedin Fig. 8(a). We repeated the experimental set-up (number of points and random networks) used in randomly networks test previously described, and tested for varying noise levels σ_n and maximal noderanks Q. The results shown in Figs. 8(b), (c) are similar to those reported for the random graphs in Fig. 7. We also depict the reconstructions for the noise-freecase $(\sigma_n = 0)$ for the SR scheme in Fig. 8(d).

The domains depicted in Fig. 9 are of particular interest as they show special graph-like structures. These were used by Cucuringu *et al.* [15], and are essentially one-dimensional manifolds. The *SPIRAL* domain (Fig. 9(b)) was used in machine learning papers as an example of a one-dimensional manifold embedded in a two-dimensional domain. We normalized both domains to be of range $[-1, 1]$ to compare their average localization error with the domains used in the previous figures. As before, we used $R_0 = 0.1$, $K = 20$ anchor points and $Q = 30$ neighbors at most per node. The *PACM* and *SPIRAL* domains consist of 452 and 2259 points, respectively. The results are similar to ones achieved over the previous domains, where the SR outperformed the SDP and MVU. Yet, it seems that the MVU performed relatively worse than in the previous examples.

We also compared our results on these domains to the ARAP [38], and the reconstruction results are shown in Fig. 10, where we applied both the SR and a steepest-descent based refinement scheme, denoted as SD-SR. The ARAP outperformed the proposed SR and SD-SR on the PACM network, while the SR and SD-SR proved superior on the more difficult SPIRAL network.

[6]Available at: http://www.stanford.edu/~yyye/.

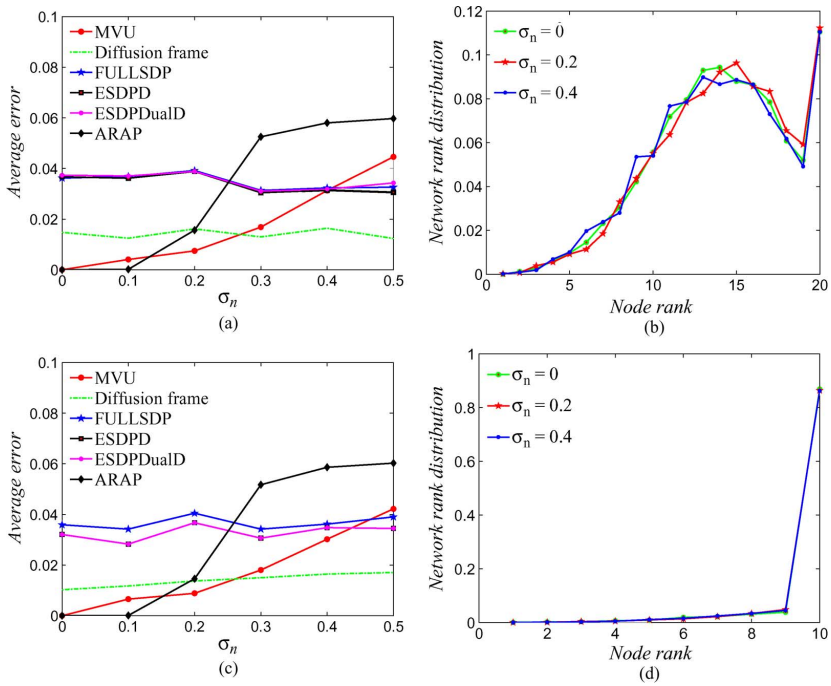

Fig. 7. The localization of random networks whose nodes are uniformly spread over the domain $[-0.5, 0.5]^2$. Each network consists of 500 points. In (b) we depict the distribution of node ranks. In (c), (d) we repeat the experiment with networks with a maximal node rank of $Q = 10$. Note the similar results in (a) and (c) despite the different nodes rank histograms. (a) 500 points, $\mathrm{Q} = 20$; (b) 500 points, $\mathrm{Q} = 20$; (c) 500 points, $\mathrm{Q} = 10$; (d) 500 points, $\mathrm{Q} = 10$.

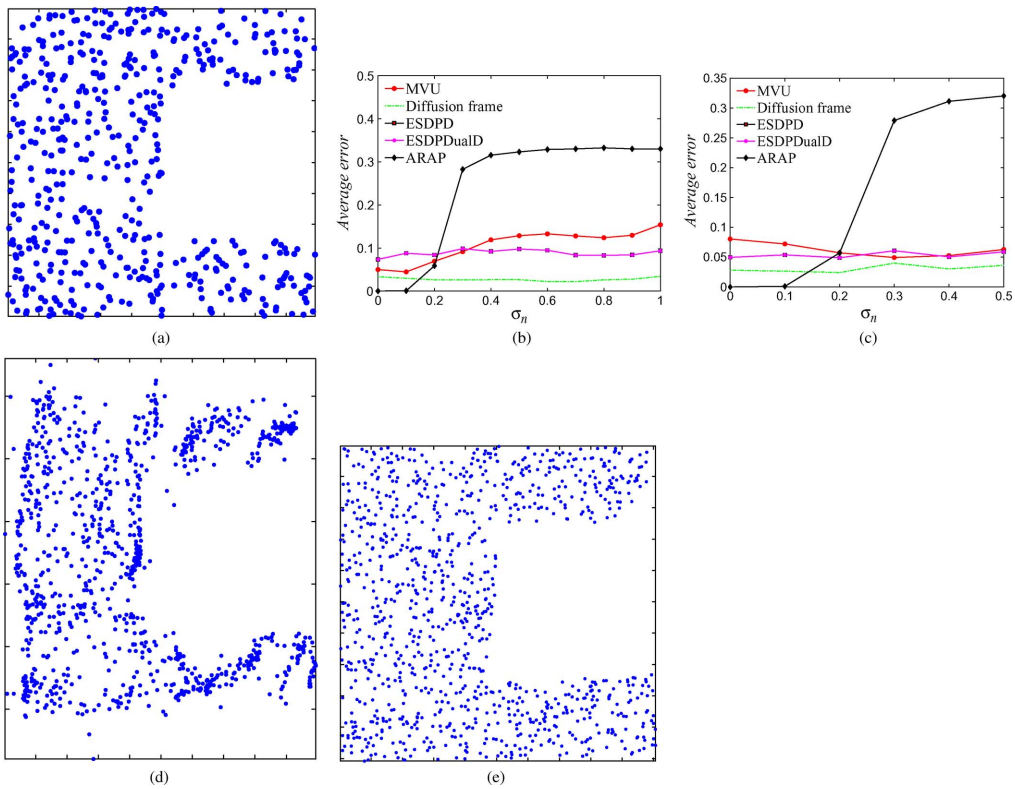

Fig. 8. The localization of random networks defined over nonconvex domains. In each simulation we uniformly drew 500 random points within the domain shown in (a). (b) and (c) report accuracy rate simulations with respect to varying noise levels. (d) depicts the reconstruction of the SR, while (e) shows the final reconstruction after SD refinement was applied to (d). (a) 500 points; (b) 500 points, $Q = 20$, $K = 20$; (c) 500 points, $Q = 10$, $K = 20$; (d) SR, $\sigma_n = 0$, $\mathrm{MSE} = 0.0044$; (e) SD-SR, $\sigma_n = 0$, $\mathrm{MSE} = 0.00007$.

The networks differ in their number of nodes: 452 for the *PACM*, versus 2259 points for the *SPIRAL*. This implies that the *SPIRAL* is better represented by a two-dimensional manifold yielding the improved results of our scheme. We believe that such networks would be better localized by first localizing small network patches as discussed in Section VI.

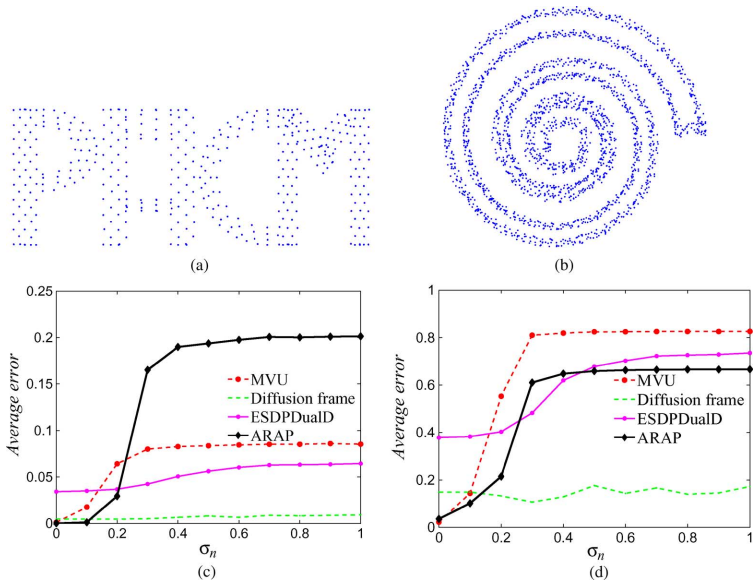

Fig. 9. Localization of the domains with special topology used by Cucuringu et. al. [15]. The localization results of the *PACM* (a) and *SPIRAL* (b) networks are depicted in Figs. (c) and (d), respectively. (a); (b) 500 points; (c) PACM; (d) SPIRAL.

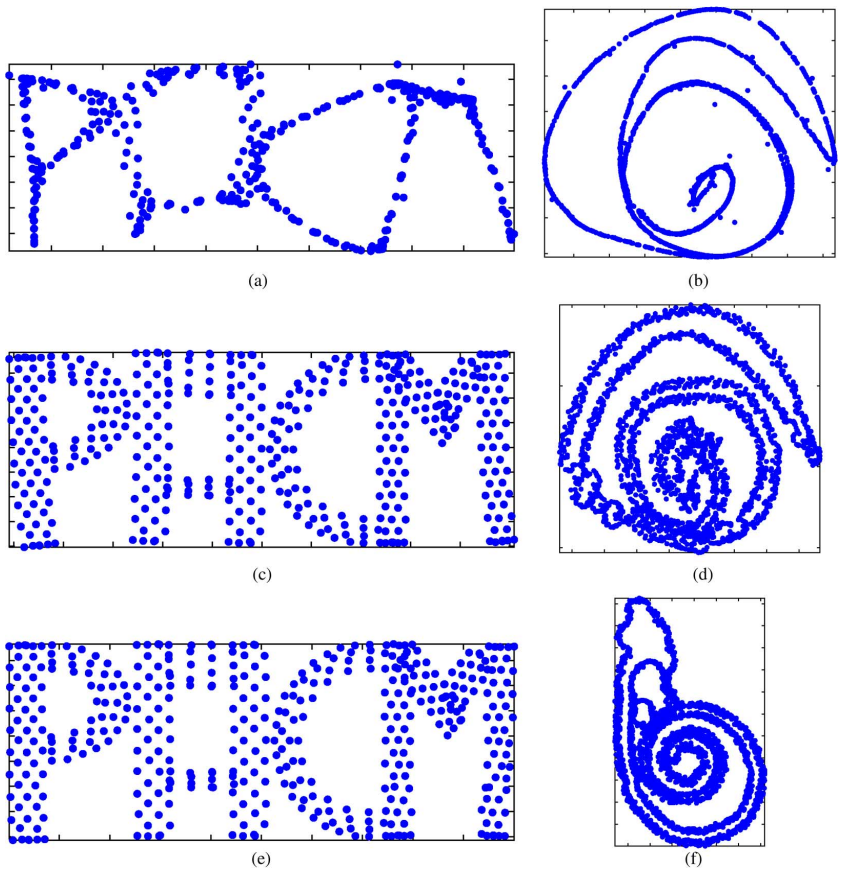

Fig. 10. Localization results for PACM and *SPIRAL* domains. For the PACM we normalized the largest dimension to [0,1] and set $R_0 = 0.1$ and $\sigma_n = 0$. For the *SPIRAL* we set $R_0 = 0.1$ and $\sigma_n = 0$. (a) SR, $\sigma_n = 0$, $\mathrm{MSE} = 0.0017$; (b) SR, $\mathrm{MSE} = 0.0269$; (c) SD-SR, $\sigma_n = 0$, $\mathrm{MSE} = 0.00004$; (d) SD-SR, $\mathrm{MSE} = 0.0116$; (e) ARAP, $\sigma_n = 0$, $\mathrm{MSE} = 0.00002$; (f) ARAP, $\mathrm{MSE} = 0.1067$.

One of the upsides of the proposed scheme is its relatively low computational time and complexity. The running time of the compared algorithms are depicted in Fig. 11, and it follows that the SDP based schemes are slower by a few orders of magnitude, and the same applies to the ARAP. We run the timing simulations on a 2.8 GHz Intel Quad computer. Our SR scheme

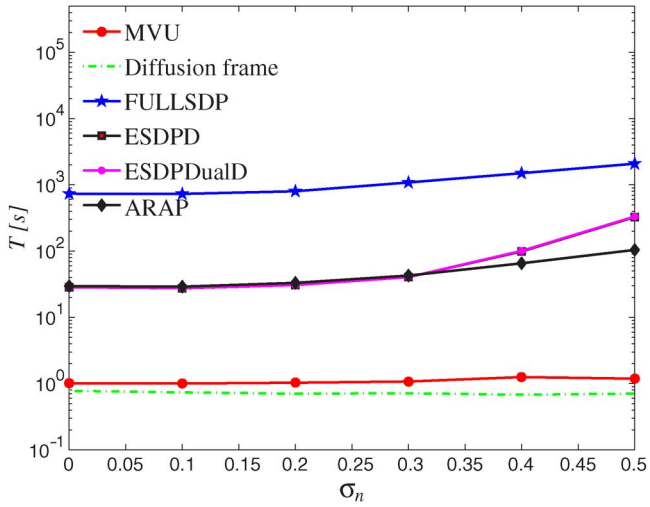

Fig. 11. The running time of the localization schemes when applied to the set of 500 random points drawn randomly over $[-0.5, 0.5]^2$. The proposed SR scheme and the MVU are on par, while the SDP based approaches are slower by a few orders of magnitude.

was implemented in Matlab and was not optimized for performance. We used Romberg's l_1-Magic package[7] as our L_1 minimization algorithm.

To conclude, we compared the proposed scheme to prior state-of-the-art schemes using the test cases used by the authors of these algorithms. Our scheme is shown to be more robust to noise that is a significant advantage for real applications. It is faster by a few orders of magnitude compared to the SDP based schemes [35] and is on par with the MVU based approach [36].

VIII. Conclusion

We presented a sensor networks localization scheme based on geometrically adaptive diffusion bases. These bases were derived by modeling the sensor network by its sparse connectivity graph. We extend previous results by Saito on such bases, by showing that the affinity matrix based on the full graph can be approximated by the one computed using the sparse connectivity graph, produced by only retaining close nodes. In order to resolve the embedding scale issue, we introduced the diffusion frames and applied L_1 minimization instead of linear regression. The resulting scheme is shown to out perform previous state-of-the-art approaches [35], [36], [38] in terms of accuracy, when the input distance measurements are noisy.

Acknowledgment

The authors would like to thank R. Coifman and A. Singer for their thorough and constructive suggestions and discussions during this work.

References

[1] D. Achlioptas and F. Mcsherry, "Fast computation of low-rank matrix approximations," *J. ACM*, vol. 54, no. 2, p. 9, 2007.
[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, pp. 393–422, 2002.
[3] J. Aspnes, W. Whiteley, and Y. R. Yang, "A theory of network localization," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1663–1678, 2006.
[4] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 6, no. 15, pp. 1373–1396, Jun. 2003.
[5] P. Biswas, H. Aghajan, and Y. Ye, "Semidefinite programming algorithms for sensor network localization using angle information," in *Pro. 39th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, 2005, pp. 220–224.
[6] P. Biswas, T. C. Liang, K. C. Toh, Y. Ye, and T. C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, 2006.
[7] P. Biswas, T. C. Lian, T. C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sen. Netw.*, vol. 2, no. 2, pp. 188–220, 2006.
[8] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proc. 3rd Int. Symp. Inform. Process. Sensor Netw.*, New York, 2004, pp. 46–54, ACM.
[9] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Berlin, Germany: Springer-Verlag, 2005.
[10] F. Chung, Spectral Graph Theory CBMS-AMS, May 1997, no. 92.
[11] M. Coates, Y. Pointurier, and M. Rabbat, "Compressed network monitoring," in *Proc. 2007 IEEE/SP 14th Workshop Statist. Signal Process.*, Washington, DC, 2007, pp. 418–422, IEEE Computer Society.
[12] R. Coifman, private communication, 2010.
[13] R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmonic Anal.: Special Issue Diffusion Maps Wavelets*, vol. 22, pp. 5–30, Jul. 2006.
[14] J. A. Costa, N. Patwari, and A. O. Hero, III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sensor Netw.*, vol. 2, no. 1, pp. 39–64, 2006.
[15] M. Cucuringu, Y. Lipman, and A. Singer, "Sensor network localization by eigenvector synchronization over the Euclidean group," *ACM Trans. Sensor Netw.*, 2011.
[16] Y. Ding, N. Krislock, J. Qian, and H. Wolkowicz, "Sensor network localization, euclidean distance matrix completions, and graph realization," in *MELT '08: Proc. 1st ACM Int. Workshop Mobile Entity Localization Tracking in GPS-Less Environ.*, New York, 2008, pp. 129–134, ACM.
[17] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. IEEE 20th Annu. Joint Conf. IEEE Comput. Commun. Soc. (IEEE INFOCOM)*, 2001, vol. 3, pp. 1655–1663.
[18] C. Gotsman and Y. Koren, "Distributed graph layout for sensor networks," *J. Graph Algorithms Appl.*, vol. 9, no. 3, pp. 327–346, 2005.
[19] J. C. Gower, "Euclidean distance geometry," *Math. Sci.*, vol. 7, pp. 1–14, 1982.
[20] J. C. Gower, "Properties of euclidean and non-Euclidean distance matrices," *Linear Algebra Its Appl,*, vol. 67, pp. 81–97, 1985.
[21] U. Khan, S. Kar, and J. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Trans Signal Process.*, vol. 57, no. 5, pp. 2000–2016, May 2009.
[22] S. Lederer, Y. Wang, and J. Gao, "Connectivity-based localization of large scale sensor networks with complex shape," in *Proc. 27th Annu. IEEE Conf. Comput. Commun.*, May 2008, pp. 789–797.
[23] K. W. K. Lui, W. K. Ma, H. C. So, and F. K. W. Chan, "Semi-definite programming approach to sensor network node localization with anchor position uncertainty," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 2245–2248.
[24] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. 2nd Int. Conf. Embed. Netw. Sensor Systems (SenSys'04)*, Baltimore, MD, 2004, pp. 50–61.
[25] B. Nadler, "Finite sample approximation results for principal component analysis: A matrix perturbation approach," *Annal. Stat.*, vol. 36, no. 6, pp. 2791–2817, 2008.
[26] X. Nguyen, M. I. Jordan, and B. Sinopoli, "A kernel-based learning approach to ad hoc sensor network localization," *ACM Trans. Sen. Netw.*, vol. 1, no. 1, pp. 134–152, 2005.
[27] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proc. GLOBECOM*, 2001, pp. 2926–2931.

[7]Available at: http://www.acm.caltech.edu/l1magic/.

[28] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.* vol. 22, no. 4, pp. 54–69, 2005 [Online]. Available: http://dx.doi.org/10.1109/MSP.2005.1458287

[29] N. Patwari, I. Hero, A. O. , M. Perkins, N. Correal, and R. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[30] N. Saito, "Data analysis and representation on a general domain using eigenfunctions of Laplacian," *Appl. Comput. Harmonic Anal.*, vol. 25, pp. 68–97, 2007.

[31] C. Savarese, J. M. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *Proc. General Track Annu. Conf. USENIX Annu. Techn. Conf.*, Berkeley, CA, 2002, pp. 317–327, USENIX Association.

[32] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 11, pp. 961–974, 2004.

[33] J. P. Sheu, P. C. Chen, and C. S. Hsu, "A distributed localization scheme for wireless sensor networks with improved grid-scan and vector-based refinement," *IEEE Trans. Mobile Comput.*, vol. 7, no. 9, pp. 1110–1123, 2008.

[34] A. Singer, "A remark on global positioning from local distances," *Proc. Nat. Acad. Sci.*, vol. 105, no. 28, pp. 9507–9511, 2008.

[35] Z. Wang, S. Zheng, Y. Ye, and S. Boyd, "Further relaxations of the semidefinite programming approach to sensor network localization," *SIAM J. Optimization*, vol. 19, no. 2, pp. 655–673, 2008.

[36] K. Q. Weinberger, F. Sha, Q. Zhu, and L. K. Saul, "Graph laplacian regularization for large-scale semidefinite programming," in *Proc. NIPS*, 2006, pp. 1489–1496.

[37] J. G. Yue Wang and S. Lederer, "Connectivity-based sensor network localization with incremental Delaunay refinement method," in *Proc. 28th Annu. IEEE Conf. Comput. Commun.*, 2009.

[38] L. Zhang, L. Liu, C. Gotsman, and S. J. Gortler, "An as-rigid-as-possible approach to sensor network localization," *ACM Trans. Sensor Netw.*, 2010.

Yosi Keller received the B.Sc. degree in electrical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, in 1994. He received the M.Sc. and Ph.D. degrees in electrical engineering from Tel-Aviv University, Tel-Aviv, in 1998 and 2003, respectively.

From 1994 to 1998, he was a R&D Officer in the Israeli Intelligence Force. From 2003 to 2006, he was a Gibbs Assistant Professor with the Department of Mathematics, Yale University. He is currently a Senior Lecturer at the Electrical Engineering Department in Bar Ilan University, Israel. His research interests include graph based data analysis, optimization, and spectral graph theory-based dimensionality reduction.

Yaniv Gur received the BS.c. and M.A. degrees in physics from the Technion-Israel Institute of Technology, Haifa, Israel, in 1999 and 2004, respectively. He received the Ph.D. degree in applied mathematics from Tel-Aviv University, Tel-Aviv, Israel, in 2008.

From 2008 to 2010, he was a Postdoctoral Research Associate at the School of Engineering in Bar Ilan University, and a Researcher at the Technion Research and Development Foundation in Haifa. Since October 2010, he is a Postdoctoral Research Associate at SCI Institute, University of Utah, Salt Lake City. His research interests include variational and PDE methods in image processing, shape analysis, optimization, and assignment problems.