Sensor Network Localization Using Sensor Perturbation

Sensor Network Localization Using Sensor Perturbation

Yuanchen Zhu
Harvard University
yzhu@fas.harvard.edu

Steven J. Gortler
Harvard University
sjg@cs.harvard.edu

Dylan Thurston
Columbia University
dpt@math.columbia.edu

***Abstract*—Sensor network localization is an instance of the NP-HARD graph realization problem. Thus, methods used in practice are not guaranteed to find the correct localization, even if it is uniquely determined by the input distances.**

In this paper, we show the following: if the sensors are allowed to wiggle, giving us perturbed distance data, we can apply a novel algorithm to realize arbitrary generically globally rigid (GGR) graphs (or maximal vertex subsets in non-GGR graphs whose relative positions are fixed). And this algorithm works in any dimension.

In the language of structural rigidity theory, our approach corresponds to calculating the approximate kernel of a generic stress matrix for the given graph and distance data. To make our algorithm suitable for real-world application, we present techniques for improving the robustness of the algorithm under noisy measurements, and a strategy for reducing the required number of measurements.

I. Introduction

In this paper, we revisit the problem of determining the 2D geometric coordinates (up to an unknown Euclidean transform) of a set of sensors in a network, where we have the ability to measure the distances between only certain pairs of sensors. This can be modeled as a graph realization problem.

Due to the importance of the problem, many effective heuristic strategies and numerical algorithms have been proposed [1], [2], [3], [4], [5], [6], [7], [8]. But graph realization is, in general, NP-HARD, as is its decision problem Graph-Embeddability [9]. Thus, such methods are not guaranteed to find the correct (or even approximate) localization, even if it is uniquely determined by the input distances. We do note that some methods are guaranteed to work on special families of input, for example trilateration will work on "trilateration graphs" [5], and methods based on semi-definite programming such as [3] will work on input that is "universally globally rigid" [10]. Both types of input are proper subsets of "globally rigid" input.

In this paper we investigate a variant of the problem, where we are able to gather more data, in the form of perturbation data. In particular, we assume that the sensors are repeatedly able to move by small amounts, and the distances along the graph edges are repeatedly measured. We do not need to know anything about the directions or specific distances of the movement, as long as the motions are not too large or too small.

Our main theoretical result is that with enough perturbation data, of high enough accuracy, we are guaranteed to be able to localize any "generically globally rigid" (GGR) graph up to an unknown Euclidean transform. The main theoretical tool we use is the "stress normal theorem" from the structural rigidity literature. This theorem tells us that the space of "equilibrium stresses" of an embedding of a graph is the orthogonal complement to the tangent space of the image of a certain map. From the perturbation data, we can estimate this tangent space, and thus the space of equilibrium stresses. From these stresses, we are then able to correctly localize the graph. In this paper, we also discuss the class of subgraphs that we will be able to localize with our approach, even when the graph is not GGR.

On the practical side, we discuss numerical methods for estimating the relevant quantities in order to deal with measurement noise, as well as strategies for reducing the required measurements.

Although our approach requires many measurements, with faster measurement technology, this novel approach may become more practical in the future. Moreover, our approach can also be used to augment current methods, by applying our method only on the difficult sub-sections of a network. Finally, our method immediately extends to 3 and higher dimensions.

We refer interested readers to [11] for a more detailed discussion of our method.

II. Theory

In this section we first fix some notations. Most can be found in [12], [13]. Then we describe the theory behind our approach.

A. Definitions

A *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a pair consisting of a set of v *vertices*, $\mathcal{V} = \{1, \ldots, v\}$, and a set of e undirected *edges*, $\mathcal{E}$. Elements in $\mathcal{E}$ are two-element (unordered) subsets of $\mathcal{V}$. Given $i \in \mathcal{V}$, we define its *neighboring vertices* $\mathcal{N}(i) = \{j : \{i, j\} \in \mathcal{E}\}$.

A *framework* is a graph $\mathcal{G}$ together with a mapping from $\mathcal{V}$ to $\mathbb{R}^d$, assigning a position in $\mathbb{R}^d$ to each vertex. We use $C^d(\mathcal{G})$ to denote the space of frameworks for a given graph $\mathcal{G}$ and a given dimension d. Given a framework $\rho \in C^d(\mathcal{G})$, $\rho(i)$ is then the position of vertex i in $\mathbb{R}^d$. Two frameworks $\rho, \rho' \in C^d(\mathcal{G})$ are *congruent*, denoted by $\rho \cong \rho'$, if there exists $R \in \text{Euclid}(d)$ such that $\rho = R \circ \rho'$, where $\text{Euclid}(d)$ is the space of Euclidean transforms in $\mathbb{R}^d$. We also give $C^d(\mathcal{G})$ a metric by observing that trivially $C^d(\mathcal{G}) \cong \mathbb{R}^{vd}$.

The *length-squared function*, $l : C^d(\mathcal{G}) \to \mathbb{R}^e$, is a function that takes in a framework ρ and outputs a e-dimensional vector, assigning to each edge in $\mathcal{G}$ its squared length, i.e., $l(\rho)^{\{i,j\}} = \|\rho(i) - \rho(j)\|^2$ for $\{i,j\} \in \mathcal{E}$. Note that we index elements of vectors in $\mathbb{R}^e$ by edges in $\mathcal{E}$.

A framework ρ is *rigid* if there exists $\delta > 0$ such that $\|\rho' - \rho\| < \delta$ and $l(\rho) = l(\rho')$ implies $\rho \cong \rho'$. This captures the idea that the graph cannot continuously flex. A framework ρ is *globally rigid* if $l(\rho) = l(\rho')$ always implies $\rho \cong \rho'$. In graph realization, we want to compute ρ up to a Euclidean transform given $l(\rho)$. Thus the graph realization problem is well-defined if and only if ρ is globally rigid.

A subset of vertices $\mathcal{S}$ is called *globally linked* in a framework ρ if, in all frameworks of $\mathcal{G}$ with the same edge lengths as ρ, the embeddings of $\mathcal{S}$ are congruent to each other [14]. Note, a vertex subset can be globally linked in a framework, even if they do not induce a globally rigid subgraph of G; because *all* of the edges of $\mathcal{E}$ can indirectly constrain the embedding of $\mathcal{S}$ (ee Figure 1).

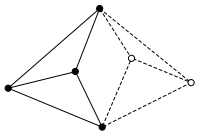

Fig. 1. The filled vertices are globally linked in the framework, but the filled vertices and solid edges alone do not form a globally rigid framework.

A framework $\rho \in C^d(\mathcal{G})$ is *generic* if the coordinates of its vertex positions, i.e., elements of the matrix $(\rho(1), \ldots, \rho(v))$, do not satisfy any algebraic equation with rational coefficients. Focusing on generic frameworks allows one to avoid various rare singularities and coincidences.

Gortler et al. [13] showed that a generic framework of a graph is globally rigid if and only if it has a "minimal stress kernel" (defined below). Moreover, all generic frameworks of a given graph in a given dimension d behave identically with respect to global rigidity. With this in mind, a graph that is globally rigid for all generic frameworks is called *generically globally rigid* (GGR).

B. Measurement Tangent Space

We denote by $l(C^d(\mathcal{G})) \subset \mathbb{R}^e$ the set of all length squared measurements of all frameworks of the graph in R^d. This set is semi-algebraic and thus, is a manifold at $l(\rho)$ for almost every, (and for all generic) ρ. The exceptions form a measure zero set, which satisfies some low order polynomial equation, so such singular frameworks are exceedingly rare. For non-singular ρ, there is a well defined *measurement tangent space* of $l(C^d(\mathcal{G}))$ at $l(\rho)$, which we will denote by $T(\rho)$.

In our sensor network localization approach, we will assume that we can measure not only $l(\rho)$ but also $l(\rho + \Delta_i)$ for some sufficient number of Δ_i's. With such measurements we can then use PCA to fit an approximate $T(\rho)$ to these points. In the limit, with smaller and smaller perturbations, and no noise, this will converge to the correct linear space for all generic ρ.

C. The Stress Normal Theorem

The linearization (Jacobian) of l at point ρ is $l^*_\rho : \mathbb{R}^{vd} \to \mathbb{R}^e$, and its matrix representation with respect to the standard bases in $\mathbb{R}^{vd}$ and $\mathbb{R}^e$ is called the *rigidity matrix*. For non singular ρ, we have $T(\rho) = l^*_\rho(\mathbb{R}^{vd})$, i.e., the tangent space is equal to the column span of the Jacobian.

A *stress vector* of a framework $\rho \in C^d(\mathcal{G})$ is a vector $\omega \in \mathbb{R}^e$ such that for each $i \in \mathcal{V}$,

$$\sum_{j \in \mathcal{N}(i)} \omega^{\{i,j\}}(\rho(j) - \rho(i)) = 0. \tag{1}$$

In other words, the position of each vertex is described as the weighted average of the positions of its neighbors, with weights given by ω. (And recall that our edges are undirected, so $\omega^{\{i,j\}} = \omega^{\{j,i\}}$.) We denote the vector space of all stress vectors of the framework ρ by $S(\rho)$ and call it the *stress space* at ρ.

The connection between the measurement tangent at $l(\rho)$ and the stresses $S(\rho)$ is described by what we will call "the stress normal theorem".

Theorem 1: For all ρ, $S(\rho)$ is the orthogonal complement to $l^*_\rho(\mathbb{R}^{vd})$ in $\mathbb{R}^e$.

This theorem can be proved with straightforward calculation. It has been used explicitly (see for example Lemma 2.5 of [12]) and implicitly (see for example page 186 of [15]) in the structural rigidity literature for many years. And thus, for generic ρ, $S(\rho)$ is orthogonal complement to $T(\rho)$ in $\mathbb{R}^e$.

We can now conclude that, from the measurement tangent space, we can compute the stress space for a generic framework.

D. Positions (Almost) from Stresses

A *stress matrix* of a framework ρ is a rearrangement of a stress vector into a matrix form, with suitably chosen diagonal entries. It is a $v \times v$ symmetric real matrix Ω satisfying: (i) For all $i,j \in \mathcal{V}$: $\Omega_{i,j} = \omega^{\{j,i\}}$ if $i \neq j$ and $\{i,j\} \in \mathcal{E}$; (ii) For all $i,j \in \mathcal{V}$: $\Omega_{i,j} = 0$ if $i \neq j$ and $\{i,j\} \notin \mathcal{E}$; (iii) $\Omega\,(1, \ldots, 1)^T = 0$. Because of the bijective correspondence between stress vectors and stress matrices, we will write $\Omega \in S(\rho)$.

The *stress kernel* $K(\rho)$ of a framework ρ is the intersection of the kernels of all its stress matrix, i.e., $K(\rho) = \bigcap_{\Omega \in S(\rho)} \ker \Omega$. From the stress space, we can easily compute the stress kernel. (Moreover, the stress kernel will exactly match the kernel of almost any randomly selected stress matrix from the stress space.)

From the definition, we can state that $K(\rho)$ contains $\mathbf{1} = (1, \ldots, 1)^T$ and the column space of $(\rho(1), \ldots, \rho(v))^T$. This corresponds to the fact that every stress matrix in $S(\rho)$ is also a stress matrix for any ρ' related to ρ by some d-dimensional affine transform.

Suppose now $\dim K(\rho) = d + 1$ (i.e. it contains only the vectors spanned by the above statement). Then we say that the framework has a *minimal stress kernel*. Suppose we have computed a basis for such a minimal stress kernel $K(\rho)$: $\{\mathbf{1}, \mathbf{x}_1, \ldots, \mathbf{x}_d\}$. Define $\rho' \in C^d(\mathcal{G})$ such that $\rho'(i)$ is the

ith column of $(\mathbf{x}_1, \ldots, \mathbf{x}_d)^T$. Then ρ must be related to ρ' by a d-dimensional affine transformation. Let us denote by L the d by d matrix representing the linear part of this affine transform.

Thus if the framework has a minimal stress kernel, then from the computed stress space (or even a single random stress matrix) we can compute the coordinates of the vertices in $\mathcal{V}$ up to an affine transform.

If the framework does not have a minimal stress kernel, all is not lost. Suppose there is a subset $\mathcal{S} \subset \mathcal{V}$, such that $K(\rho)$ is in fact of dimension $d+1$ when projected onto the coordinates corresponding to $\mathcal{S}$. (The natural inclusion $\mathcal{S} \subset \mathcal{V}$ induces a projection $\pi_{\mathcal{S}} : \mathbb{R}^{|\mathcal{V}|} \to \mathbb{R}^{|\mathcal{S}|}$.) Then, using the same reasoning, from the *projected stress kernel* $\pi_{\mathcal{S}} K(\rho)$, we can compute the coordinates of the vertices in $\mathcal{S}$ up to an affine transform.

To find maximum subsets with a minimal projected stress kernel, we start by finding a seed component $\mathcal{S}_0$ with $d+1$ vertices satisfying the condition, and then we sequentially (order does not matter for linear dependence) add any vertex i to the component as long as $\dim \pi_{\mathcal{S}_0 \cup \{i\}} K(\rho)$ remains $d+1$, until there is none that can be added without increasing the dimension. Conveniently, all generic frameworks of a given graph in a given dimension behave identically with respect to this property. So instead of using the noisy input distance data, we find such subsets using exact distances on a random test framwork of the graph that we generate ourselves.

E. Removing the Affine

We will now use the edge lengths to remove this free affine transform in the coordinates of $\mathcal{V}$ (resp. of $\mathcal{S}$). Let ρ' and L be as defined in the previous section. Consider the following system of equation, in the unknown $d \times d$ matrix L.

$$\forall \{i,j\} \in \mathcal{E} : \|L(\rho'(i) - \rho'(j))\|^2 = l(\rho)^{\{i,j\}} \tag{2}$$

We know there must be some solution L.

To solve this system let $M = L^T L$ giving us the following set of linear equations (in the $\frac{d(d+1)}{2}$ unknowns of M):

$$\forall \{i,j\} \in \mathcal{E} : (\rho'(i) - \rho'(j))^T M (\rho'(i) - \rho'(j)) = l(\rho)^{\{i,j\}}. \tag{3}$$

(For the case of a subset of vertices, we define $\mathcal{E}$ to be the set of edges with both vertices in the subset $\mathcal{S}$).

Using Cholesky decomposition on M then yields L.

The only remaining concern is whether, this system has more than one solution. Fortunately, it was proven in Prop 4.3 of [12] that if ρ is a generic framework of a graph where each vertex has degree at least d, then the solution to (3) must be unique. Moreover in his proof for his Theorem 1.3, Connelly shows that if generic framework of graph with $d+2$ or more vertices, has a minimal stress kernel, then each vertex must have degree at least $d+1$.

When dealing with a subset of the vertices, and a projected stress kernel, we do not know of any specific guarantees of the vertex degrees, and must check this condition manually.

F. When Will We Succeed?

In summary, if a generic framework has a minimal stress kernel, then our algorithm will correctly localize the graph up to a Euclidean transform. Moreover, as mentioned above, this condition is in fact equivalent to the generic global rigidity of the underlying graph [13].

If the graph is not GGR, then our algorithm will correctly localize any vertex subset $\mathcal{S}$ with $\dim \pi_{\mathcal{S}} K(\rho) = d+1$ and vertex degree at least d in the induced subgraph. In fact, this condition is sufficient for $\mathcal{S}$ to be globally linked in all generic frameworks, although whether it is a necessary condition is still an open question.

III. Numerical Algorithm

A. Estimating the Measurement Tangent, and Stress Space

We assume that we can perturb each vertex of the graph to obtain a slightly different framework, and measure its image under l. Formally, let $\delta > 0$ be a predefined parameter which we call *perturbation radius*, and define $B_\delta(\rho) = \{\rho' \in C^d(\mathcal{G}) : \|\rho'(i) - \rho(i)\| < \delta \text{ for all } i \in \mathcal{V}\}$. We also assume that we can obtain $l(\rho_1)$, …, $l(\rho_M)$ for some fixed number, M, of random samples $\rho_i \in B_\delta(\rho)$. We then approximate $T(\rho)$ by fitting a hyperplane to these points using PCA.

Choosing M. In order to properly approximate the measurement tangent space $T(\rho)$ through PCA, we set the number of samples, M, to $\lceil \mu \dim T(\rho) \rceil$, where $\mu \geq 1$ is a *sampling factor* parameter. The exact value of μ depends on how noisy the distance measurements are. In practice, setting $\mu = 4$ often suffices for moderate level of noise. Note that $\dim T(\rho)$ is a generic property which can be calculated from almost any random framework of $\mathcal{G}$ in $\mathbb{R}^d$. Moreover, when the framework is rigid, $\dim T(\rho)$ takes its upper bound of $(vd - \binom{d+1}{2})$. Thus the number of perturbations we need scales linearly with vd. For each perturbation ρ_i, an entire set of edge length measurements then need to be made to get $l(\rho_i)$, so the total number of required pair-wise length measurements is $O(vde)$. In Section III-D, we present a strategy for reducing the number of required measurement.

Choosing δ. Ideally, δ should be as small as possible. Then in the limit, our approximated measurement tangent space will converge to the true $T(\rho)$. However, in practice, length measurement is noisy, and δ needs to be set large enough to dominate the noise. In our simulation, we find that when the noise is normally distributed with variance σ^2, setting $\delta = 20\sigma$ produces good results for a wide variety of graphs (see Section IV for details).

PCA details. Because $l(C^d(\mathcal{G}))$ is a cone, $T(\rho)$ must include the origin. Thus we use the origin $\mathbf{0} = (0, \ldots, 0)^T$ rather than the statistical mean of the $l(\rho_i)$'s as the center for PCA. We also verified experimentally that this choice often produces more accurate results in the presence of measurement noise.

B. Stress Kernel from Measurement Tangent

The stress kernel is the same as the kernel of any generic stress matrix in $S(\rho)$, so we could take a random stress matrix

from the stress space and compute its kernel. With no noise, we know that the kernel of such an Ω for a GGR graph will be well-defined and of dimension $d+1$.

However, in the presence of measurement noise, there no longer is a distinct gap in magnitude between the $(d+1)$th and the $(d+2)$th smallest eigenvalues of Ω, i.e., the kernel of Ω is no longer numerically well-defined. If we naively use the eigenvectors corresponding to the smallest $d+1$ eigenvalues to recover the graph embedding, we usually get extremely large error in the resulting realization.

We find it is possible to dramatically increase the accuracy of the computed stress kernel by taking into account more than a single stress matrix. Calculating the eigenvectors of Ω corresponding to the smallest eigenvalues is equivalent to computing orthonormal vectors $\mathbf{y}_1, \ldots, \mathbf{y}_{d+1}$ such that $\|\Omega \mathbf{y}_i\|^2$ is minimized in the order of $i = 1, \ldots, d+1$. We extend this notion to multiple stress matrices. With stress matrices Ω_1, ..., Ω_N, we thus seek orthonormal vectors $\mathbf{y}_1$, ..., $\mathbf{y}_{d+1}$ minimizing $\sum_j \|\Omega_j \mathbf{y}_i\|^2$ in the order $i = 1, \ldots, d+1$. Since $\sum_j \|\Omega_j \mathbf{y}_i\|^2 = \mathbf{y}_i^T (\sum_j \Omega_j^T \Omega_j) \mathbf{y}_i$, the $\mathbf{y}_i$'s are simply the eigenvectors of $\Sigma = \sum_j \Omega_j^T \Omega_j$ corresponding to the smallest $d+1$ eigenvalues. We call Σ the *aggregated squared stress matrix*. In practice, using even just two stress matrices produces significant better results compared with using a single stress matrix in the presence of noise.

C. Positions from Stress Kernel

When there is noise in the measurement, we in general cannot expect (3) to be exactly solvable. Our approach is to solve the linear systems in the least square sense with the constraint $M \succ 0$, using semi-definite programming ([16], [8]). In addition, as in [8], we optionally use $D > (d+1)$ eigenvectors from Σ to recover the positions. In this case we obtain an embedding in $\mathbb{R}^{D-1}$ and must use PCA to project back down to $\mathbb{R}^d$.

D. Reducing Required Measurements

To reduce the requirement measurements, we can calculate the stress space of some overlapping subgraphs of $\mathcal{G}$: $\mathcal{H}_1, \mathcal{H}_2, \ldots$ from a fixed set of perturbed length measurements. Since these subgraphs are much smaller than $\mathcal{G}$, we can use a much smaller measurement set. In particular, the subsets we use are the 2-rings in $\mathcal{G}$, and if there is an upper limit on vertex degree, then the number of required pairwise measurement is reduced to $O(de)$, linear in the graph size.

As described in [11], we can add together these stress spaces from the subgraphs to obtain a stress subspace of the whole graph, $S^*(p) \subset S(p)$ and a enlarged stress kernel $K^*(p) \supset K(p)$. Although we have no guarantee that $\dim(K^*(p)) = \dim(K^*(p))$, in practice it often is, and in any case we can still proceed with our algorithm and localize any vertex subset where $K^*(\rho)$ projects down to the appropriate dimension.

IV. Results and Discussions

In this section we briefly evaluate our algorithm using simulation. We refer interested readers to [11] for more detailed experimental data and discussions.

Testing graphs: We test our algorithm on two types of random graphs: The first, which we call *isotropic graphs*, are constructed by placing vertices uniform-randomly in $[0,1]^2$, and putting an edge between vertices that are within a *distance threshold*, R. The second, which we call *anisotropic graphs*, are constructed by first doing the same, but then overlaying an actual floor plan and disconnecting edges that cross walls, thus making connectivity highly non-uniform. In addition, we enforce maximal vertex degree constraints: 7 for isotropic graphs and 9 for anisotropic graphs. The actual mean vertex degree for the resulting graphs are typically around 7 and 7.5 respectively. We optionally prune the graphs to make them GGR. Note that this level of connectivity usually does not allow trilateration based method to localize the entire graph. For brevity, we report simulation results on two *fixed* instances from these two categories, shown in Figure 2.

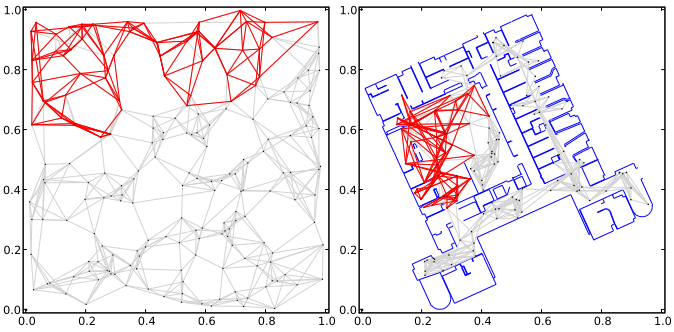

Fig. 2. Sample isotropic (left, v=200) and anisotropic (right, v=137) testing graphs. For each graph, the largest subgraph that can be localized through trilateration is shown in red. Both graphs are GGR.

Noise model: We use an additive noise model. The noise term follows a zero-mean Gaussian distribution with a standard deviation of ϵR, where ϵ is a *noise level* parameter. The commercially available UWB devices described in [17] has a maximum range of about 30 meters and its high-probability error has a standard deviation of 3cm. This corresponds to setting the noise level $\epsilon = 0.1\%$. We also present testing results of setting $\epsilon = 0.01\%$, representing a future generation of ranging devices. In our tests, We refer to these two cases as *high noise* and *low noise* settings.

Error metrics: The output of our algorithm by definition leaves free a Euclidean transform. In order to compare the computed localization, ρ', to the ground truth, ρ, we first align up the two using the least-square optimal $Q \in \text{Euclid}(d)$ minimizing $\sum_i \|\rho(i) - Q(\rho'(i))\|^2$, computed using the Procrustes method [18]. Note that this alignment step does not improve the *intrinsic* quality of the output. We then calculate two simple error metrics (which are also used in [7]): the *positional error*, σ_p, defined as

$$\sigma_p^2 = v^{-1} \sum_{i=1}^{v} \|\rho(i) - Q(\rho'(i))\|^2,$$

and the *distance error*, σ_p, defined as

$$\sigma_d^2 = e^{-1} \|l^{\frac{1}{2}}(\rho) - l^{\frac{1}{2}}(\rho')\|^2,$$

where $l^{\frac{1}{2}}(\rho)$ is simply $l(\rho)$ with every element square-rooted, i.e, it is a vector of edge lengths as opposed to squared edge lengths.

From simulation, we found that without noise ($\epsilon = 0$), and with a small perturbation radius ($\delta = 10^{-4}$), our method essentially performs as predicted by the theory, and always recovers entire GGR frameworks and any detected generically globally linked component of non-GGR frameworks with negligible error. Thus we are mostly interested in how our method performs in the presence of noise, which depends heavily on the choice of parameters described in Section III:

- μ controls how many perturbations are used.
- δ controls how much we perturb each vertex. To be relevant, we always report the ratio $\frac{\delta}{\epsilon R}$.
- N stress matrices are used to form the aggregated squared stress matrix.
- D coordinate vectors are used to reconstruct the framework.

The numerical accuracy of our algorithm for the two testing graphs and two noise levels, under suitably chosen parameters are summarized in Table I. To provide an intuitive feel for these numbers, we present corresponding plots of the error vectors $(\rho - \rho')$ in Figure 3. Note that our algorithm handles the isotropic graph very well, both at low and high noise levels. The anisotropic testing case is heavily affected by the noise level though. For more experimental data on the effects of the parameters, see [11].

TABLE I
ACCURACY OF OUR ALGORITHM UNDER SUITABLE PARAMETERS. IN ALL CASES, $\mu = 4.0$ AND $N = 200$.

		isotropic (v=200)		anisotropic (v=137)	
		high noise	low noise	high noise	low noise
ϵR	in 10^{-5}	29	2.9	29	2.9
δ	in 10^{-4}	58	23	58	23
$\frac{\delta}{\epsilon R}$		20	80	20	80
D		4	4	6	6
σ_p	in 10^{-4}	48	6.3	390	55
σ_d	in 10^{-4}	21	2.4	80	28

Experimentally we find our method produces results of similar quality on detected generically globally linked subgraphs of non-GGR graphs, though the optimal D depends on subgraph size. (Setting D proportional to the logarithm of the number of subgraph vertices produces good results for us.) Our measurement reduction strategy works very well with our isotropic testing graphs. For anisotropic testing graphs, the method can also localize a large portion of the graph. (For details, please see [11].)

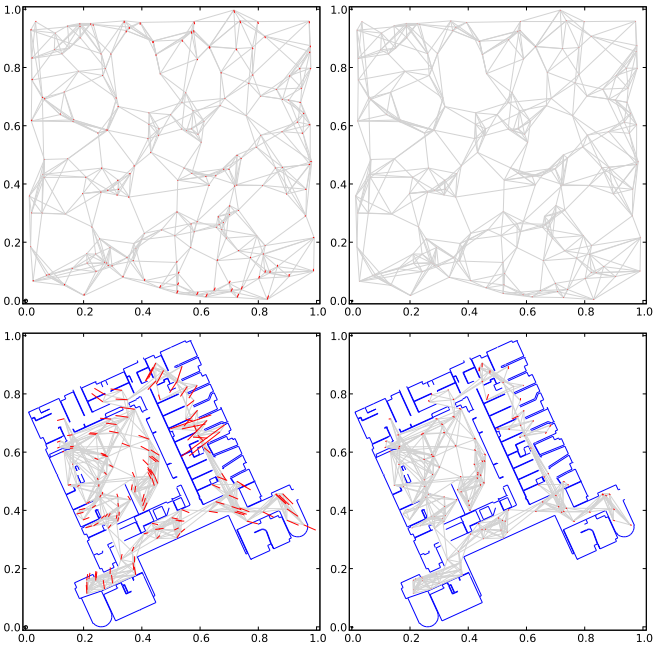

Fig. 3. Realization results for isotropic (top row) and anisotropic (bottom) testing cases under high (left) and low (right) noise condition. Drawn in red are the error vectors pointing from the ground-truth frameworks to the computed frameworks (under optimal alignment) for different testing cases.

REFERENCES

[1] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *MobiHoc '03*, 2003, pp. 201–212.

[2] L. Doherty, K. pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," *INFOCOM 2001 Proc.*, vol. 3, pp. 1655–1663 vol.3, 2001.

[3] 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.

[4] D. Niculescu and B. Nath, "Dv based positioning in ad hoc networks," *Kluwer journal of Telecommunication Systems*, pp. 267–280, 2003.

[5] T. Eren, O. Goldenberg, W. Whiteley, Y. Yang, A. Morse, B. Anderson, and P. Belhumeur, "Rigidity, computation, and randomization in network localization," *INFOCOM 2004 Proc.*, pp. 2673–2684 vol.4.

[6] B. Hendrickson, "The molecule problem: Exploiting structure in global optimization," *SIAM Journal on Optimization*, 1995.

[7] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *SenSys '04*, pp. 50–61.

[8] A. Singer, "A remark on global positioning from local distances," *Proc. of the National Academy of Sciences*, vol. 105, no. 28, pp. 9507–9511, 2008.

[9] J. B. Saxe, "Embeddability of weighted graphs in k-space is strongly np-hard," in *Proc. 17th Allerton Conf. in Communications, Control, and Computing*, 1979, pp. 480–489.

[10] B. Jackson and T. Jordán, "Operations preserving global rigidity of generic direction-length frameworks," Egerváry Research Group, Budapest, Tech. Rep. TR-2008-08, 2008, http://www.cs.elte.hu/egres.

[11] Y. Zhu, S. J. Gortler, and D. Thurston, "Sensor network localization using sensor perturbation (long version)," Harvard University, Tech. Rep., 2009.

[12] R. Connelly, "Generic global rigidity," *Discrete Comput. Geom.*, vol. 33, no. 4, pp. 549–563, 2005.

[13] S. J. Gortler, A. Healy, and D. Thurston, "Characterizing generic global rigidity," arXiv:0710.0926.

[14] B. Jackson, T. Jordán, and Z. Szabadka, "Globally linked pairs of vertices in equivalent realizations of graphs," *Discrete & Computational Geometry*, vol. 35, no. 3, pp. 493–512, 2006.

[15] L. Asimow and B. Roth, "The rigidity of graphs. ii." *J. Math. Anal. Appl.*, vol. 68, pp. 171–190, 1979.

[16] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, 1996.

[17] J. Park, E. D. Demaine, and S. Teller, "Moving-baseline localization," in *IPSN '08 Proc.*, pp. 15–26.

[18] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, pp. 1–10, 1966.