

Fig. 1. Error in the reconstruction of the sum of the three sine functions used by [1] and [3] in their Fig. 1, respectively. The S-transform has been computed by discretizing its time-domain expression rather than its frequency-domain expression.

Fig 1(b) of [3] shows the error in the reconstruction of the data from Fig. 1(d) of [1] (sum of three sine functions at 0.3, 2, and 3.5 Hz). Reference [3] uses the time-IST and freq-IST in combination with the freq-ST. In analogy, Fig. 1 shows the measured error of the time-IST and freq-IST in combination with the time-ST of the same data. By comparison with Fig. 1(b) of [3], it can be seen that the freq-IST and time-IST perform best on spectra obtained with the freq-ST and time-ST, respectively. The error is largest for the mixed combinations, that is, time-IST and freq-ST and freq-IST and time-ST. Furthermore, the error at the fundamental frequency vanishes for the time-IST in combination with the time-ST [Fig. 1(b)]. It seems from this example that the original signal is better retrieved using the freq-IST. Nevertheless, the time-IST provides a good approximation as also noticed in [3]. The S-transform, both inverse S-transforms, and the discretization errors are further discussed and illustrated in [2]. They show also examples where the time-IST performs better than the freq-IST. Furthermore, it is shown why the time-IST works.

As observed by Pinnegar, there is a mistake in the plots concerning the traces obtained with the freq-ST ($u_{\text{filt}_1}[t]$). This has no influence on the time-IST by [1] and does not change the physics, our understanding, and conclusions concerning our new approach. Pinnegar writes, “The long and short of this is that the new filtering technique, in its corrected form, does still appear to give a faster time taper than the older method, but the improvement is not as marked as implied by [1].” Reference [2] shows that the improvement depends on the filter and data as also on how one has discretized the S-transform.

As already mentioned in [1], the S-transform is a powerful method used in several motivating studies to analyze time-varying signals. As with any method, one should be aware of limitations, and the choice of the proper approach depends on the data and application.

REFERENCES

- [1] M. Schimmel and J. Gallart, “The inverse S-transform in filters with time-frequency localization,” *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4417–4422, Nov. 2005.

- [2] C. Simon, S. Ventosa, M. Schimmel, A. Helderling, J. Dañobeitia, J. Gallart, and A. Manuel, “The S-transform and its inverses: Side effects of discretizing and filtering,” *IEEE Trans. Signal Process.*, 2006, submitted for publication.
- [3] C. R. Pinnegar, “Comments on ‘The inverse S-transform in filters with time-frequency localization’,” *IEEE Trans. Signal Process.*, vol. 55, no. 10, Oct. 2007.
- [4] R. G. Stockwell, L. Mansinha, and R. P. Lowe, “Localization of the complex spectrum: The S transform,” *IEEE Trans. Signal Process.*, vol. 44, no. 4, pp. 998–1001, Apr. 1996.

A Fast and Effective Multidimensional Scaling Approach for Node Localization in Wireless Sensor Networks

Georgios Latsoudas and
Nicholas D. Sidiropoulos, *Senior Member, IEEE*

Abstract—Given a set of pairwise distance estimates between nodes, it is often of interest to generate a map of node locations. This is an old nonlinear estimation problem that has recently drawn interest in the signal processing community, due to the emergence of wireless sensor networks. Sensor maps are useful for estimating the spatial distribution of measured phenomena, and for routing purposes. We propose a two-stage algorithm that combines algebraic initialization and gradient descent. In particular, we borrow an algebraic solution known as *Fastmap* from the database literature, adapt it to the sensor network context, and motivate the placement of anchor/pivot nodes on the edges of the network. When all nodes can estimate their distance from the anchors, the overall algorithm offers very competitive performance at low complexity (quadratic in the number of nodes).

Index Terms—Multidimensional scaling, node localization, sensor networks.

I. INTRODUCTION

The problem of node localization from pairwise distance estimates has recently attracted interest in the signal processing community, owing to the growing interest in wireless sensor networks [2], [3], [5], [8], [9]. Given a matrix of pairwise distances, the localization problem aims to determine the (*relative*) node locations that generate these distances. In other words, one seeks a map of node locations with a given (approximate) distance structure. This is a classic problem originating in psychometrics [10], [11], known as *multidimensional scaling* (MDS) [6]. There are many MDS flavors and variants; perhaps the single most important one is *metric MDS*.

The classical approach to solving MDS is based on computing the principal components of a double-centered version of the matrix of squared distances. This works reasonably well (albeit not optimally in the least squares sense, due to the double centering), but its complexity is cubic in the number of nodes, and thus does not scale well

Manuscript received August 8, 2006; revised February 8, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mounir Ghogho. This work was supported by the U.S. ARO under ERO Contract N62558-03-C-0012. An earlier version of part of this work appeared in conference form at the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), Puerto Vallarta, Mexico, December 12–14, 2005.

The authors are with the Department of Electronic and Computer Engineering, Technical University of Crete, 73100 Chania—Crete, Greece (e-mail: latsoud@telecom.tuc.gr; nikos@telecom.tuc.gr).

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.2007.896101

with network size. A popular alternative to principal component analysis (PCA) is the use of gradient descent or other numerical optimization tools that aim to optimize a *stress function* of the error between the measured distances and those reproduced by a given configuration of points. The drawback of gradient descent and related approaches is that they require accurate initialization, due to the multimodal nature of the stress function.

We propose a two-stage MDS algorithm that employs an algebraic initialization procedure followed by gradient descent. The algebraic initialization step is based on the Fastmap algorithm [4], borrowed from the database literature. Fastmap is a linear-complexity mapping tool, which is, however, sensitive to measurement errors. This is not particularly relevant in the database context; therein, actual distances are computed from complete representations. Noise sensitivity is an important issue in wireless ranging applications, due to shadowing, fading, and the use of approximate path loss models.

Due to the fact that distances are invariant to coordinate frame transformations (rotation, reflection, shift), there is a need to employ three so-called *anchor nodes*, whose position is accurately known (e.g., via GPS) in order to fix a desired coordinate frame. Unfortunately, Fastmap is very sensitive to coordinate alignment, because the estimated position of every node (and thus anchor nodes as well) is only based on distances to selected *pivot* nodes—there is no averaging. In order to mitigate this problem, we advocate a judicious choice of anchor/pivot nodes, placed at the outer edges of the network. This placement also bypasses the need for alignment and thus alignment errors, thereby providing a higher quality initialization to the gradient descent. When all network nodes can estimate their distance from the anchors, the overall algorithm affords better localization accuracy than PCA-based MDS, at substantially lower complexity cost (quadratic in the number of nodes). Our algorithm is also competitive with respect to recent solutions of the same complexity order, developed specifically for node localization in sensor networks [3]. For relatively dense networks, our algorithm yields comparable estimation performance at a significantly reduced complexity relative to [3], even when the latter is initialized using our adaptation of Fastmap. An exception is the case of very sparse networks, wherein [3] with Fastmap initialization may be preferable when accuracy is more important than complexity.

We remark that there are other algorithms in the recent literature that assume a different measurement model (e.g., 0–1 node connectivity information only, as in [9]), or propose solutions of considerably higher complexity (e.g., as in [2]). We aim for the low-complexity regime, for simplicity and scalability considerations.

II. MULTIDIMENSIONAL SCALING

We denote the dissimilarity measure (the estimated distances in our case), between objects i and j as d_{ij} . The set of dissimilarities yields a measured distance matrix D . We also let \hat{d}_{ij} denote the Euclidean distance between (generated by) two points $X_i = (x_{i1}, x_{i2}, \dots, x_{im})$ and $X_j = (x_{j1}, x_{j2}, \dots, x_{jm})$, i.e.,

$$\hat{d}_{ij} = \sqrt{\sum_{k=1}^m (x_{ik} - x_{jk})^2}. \quad (1)$$

In classical metric MDS [10], [11], [6], we estimate the node coordinates by computing the m principal components of an elementwise squared and double-centered version of the matrix D , denoted by B , as follows:

$$B = -\frac{1}{2}JPJ \quad (2)$$

where $P = D \odot D$ is the matrix of squared distances (\odot denotes the elementwise matrix product), and J is the centering operator

$$J = I - ee^T/N \quad (3)$$

with N denoting the number of objects (sensors/nodes), and e denoting the $N \times 1$ vector of all 1's. For an $N \times N$ matrix D and for m dimensions, it can be shown that

$$-\frac{1}{2} \left(d_{ij}^2 - \frac{1}{N} \sum_{j=1}^N d_{ij}^2 - \frac{1}{N} \sum_{i=1}^N d_{ij}^2 + \frac{1}{N^2} \sum_{j=1}^N \sum_{i=1}^N d_{ij}^2 \right) = \sum_{k=1}^m x_{ik}x_{jk} \quad (4)$$

thus the node coordinates can be estimated from the m principal eigenvectors of the matrix B , scaled by the square roots of the corresponding eigenvalues. That is, with U_r containing the m principal eigenvectors and V_r diagonal containing the corresponding eigenvalues, $B_r = U_r V_r U_r^T$ is an optimal least-squares approximation of B , and $X_r = U_r V_r^{1/2}$ is an approximation of the node coordinates in m -dimensional space, up to a common coordinate rotation, reflection, and shift. An alignment procedure is necessary to transform the estimated node locations to a desired frame of reference.

It is important to note that, due to the preprocessing steps prior to PCA, this approach is not equivalent to nonlinear least-squares parameter fitting using the original measurements.

Direct minimization of a suitable *stress function* is an alternative to PCA-based MDS [10]. A common¹ stress function is

$$\text{stress}^2 = \sum_{i,j} w_{ij} (\hat{d}_{ij} - d_{ij})^2. \quad (5)$$

where $[w_{ij}]$ is the weight matrix, whose elements are equal to 1 if node j is in the measurement range of node i and 0 otherwise. Minimization starts with an initial guess of the node positions, followed by gradient descent iterations. Initialization matters a lot in this context, because the stress function is multimodal. Furthermore, the number of iterations required for convergence depends on the quality of the initialization.

III. FASTMAP AND PROPOSED APPROACH

The basic element of Fastmap [4] is the projection of the nodes on a properly selected line. This is achieved by selecting two objects O_a, O_b , called *pivots*, and projecting all other objects on the line that passes through them. A pair of pivots is chosen for each of the m dimensions. The coordinates (i.e., projections on the pivot line) of the objects are found by employing the *cosine law* [4]. Thus, the first coordinate for object O_i (measured from O_a , on the line connecting O_a and O_b) is given by

$$x_i = \frac{d_{ai}^2 + d_{ab}^2 - d_{bi}^2}{2d_{ab}} \quad (6)$$

where d_{ij} is the measured distance between nodes i and j and a, b are the pivot objects. After computing these coordinates for each object O_i , we consider a hyperplane that is orthogonal to the pivot line. We

¹The negative log-likelihood of the observed data under a suitable measurement noise model would seem to be the natural choice of stress function. This is not fortuitous in our context, however, because the resulting function is not only multimodal, but also leads to numerical difficulties. For this reason, a least squares criterion is preferred. While still multimodal, the adopted least-squares criterion is much more benign from a numerical optimization viewpoint, and it often yields performance close to the pertinent CRB, as will be seen in the simulations.

then project the objects on this hyperplane, and repeat the process, this time using

$$\tilde{d}_{ij}^2 = d_{ij}^2 - (x_i - x_j)^2, \quad i, j = 1, \dots, N. \quad (7)$$

Fastmap works for arbitrary placement of the three pivot nodes in the noiseless case—the formula in (6) can be derived from two applications of the Pythagorean Theorem and subtraction of the resulting equations. In noisy scenarios, the placement of pivot and anchor nodes matters. A heuristic method is proposed in [4] for choosing the pivots as far as possible from one another.

In database applications, there is no “natural” or preferred coordinate frame of reference; thus, the final alignment step is not used, and anchors are not needed. In the context of sensor networks, however, obtaining absolute position estimates is important. In Fastmap, the estimated position of every node (and thus anchor nodes as well) is only based on distances to the chosen pivot nodes—there is no averaging. The alignment of anchors is inaccurate due to noise in the anchor positions estimated from the distance information, leading to error amplification in the final estimates.

If the three pivots are chosen to form an orthogonal triangle, and the orthogonal sides of this triangle are chosen as coordinate basis vectors (which should of course be known, hence pivots = anchors), then all projections in Fastmap are computed directly onto the “native” coordinate basis. Thus, there is no need to estimate the positions of anchors and subsequently align (rotate, shift, reflect) the entire set of point estimates. If an orthonormal basis is required, then simple scaling of one coordinate suffices. Therefore, any orthogonal placement of anchors/pivots (whether inside or outside the sensor distribution area) is advantageous, because it avoids alignment errors.

Despite its conceptual and computational simplicity, Fastmap is a highly nonlinear estimator, and its performance analysis is complicated, perhaps more than one would expect at first sight. While analysis is difficult, simulations suggest that placing the anchors/pivots on the edge of the network is optimal. This is illustrated in Fig. 1, for a network of 10^3 nodes uniformly spread over a square of side 1 centered at the origin, and a multiplicative Gaussian noise model for the measured distances [see (9) in Section IV]. Note that the minimum mean-squared error (MSE) occurs at baseline = 1 (the same holds for other values of noise variance), meaning that anchors should be on the edges of the network. Also note that when estimating the x coordinate, it is preferable to place the anchors halfway through the distribution area in the y coordinate; but the difference is small. We are typically interested in estimates of equal accuracy in both axes; this implies a symmetric configuration, i.e., placing the anchors/pivots on three vertices on the edges of a minimal square that covers the sensor distribution area, as illustrated in Fig. 2.

We assume that the anchor/pivot nodes that are used by Fastmap can take distance measurements from all the sensor nodes (even if we do not have full connectivity for the rest of the nodes). This is reasonable if the anchor/pivot nodes are airborne, in higher ground, or on a mast. If the anchors are randomly dropped along with the rest of the nodes, Fastmap still applies, but its performance will be worse.

In our approach, Fastmap estimates are used as initialization for gradient descent. Denoting by (x_i, y_i) the estimated position of node i , the partial derivative of the stress function in (5) is given by

$$\frac{\partial \text{stress}}{\partial x_i} = \sum_{j \neq i} w_{ij} \frac{(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - d_{ij})(x_i - x_j)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}. \quad (8)$$

with a similar expression for the partial derivative with respect to y_i . Each step of gradient descent costs $\mathcal{O}(N^2)$. For simplicity, but also

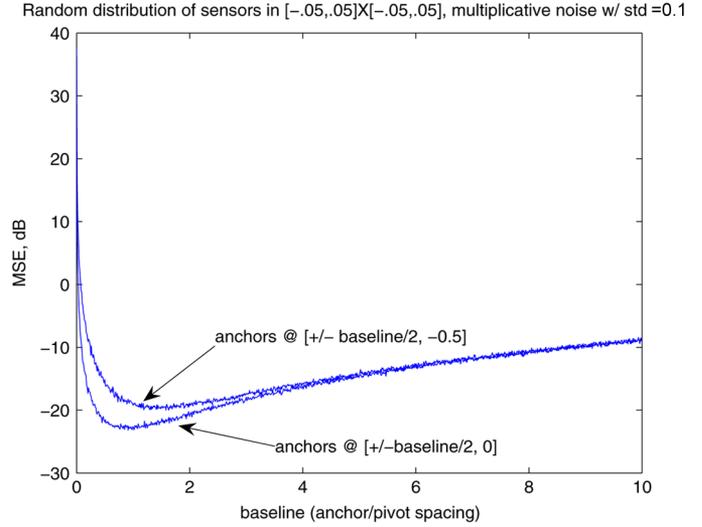


Fig. 1. MSE in Fastmap x -coordinate estimate as a function of (exactly known) anchor/pivot baseline. Sensors are uniformly spread over a square of side 1 centered at the origin. Zero-mean Gaussian multiplicative noise model for the measured distances. The minimum MSE point is stable with respect to noise variance.

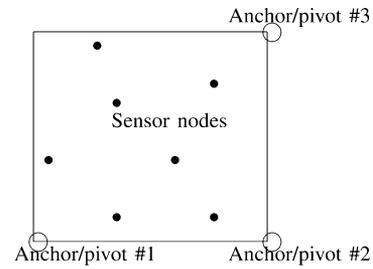


Fig. 2. Anchor/pivot node placement.

to bound complexity, a fixed small number of gradient descent steps (denoted by p) is used in our simulations.

IV. MEASUREMENT NOISE MODEL AND CRAMÉR–RAO BOUND

Pairwise distance estimates will inevitably contain measurement errors, which are typically amplified with increasing distance between nodes. The choice of measurement noise model depends on many factors, and is application specific. We shall adopt a certain multiplicative normal noise model from the recent literature on node localization in wireless sensor networks [2], [9], in which the distance measurement error is proportional to the actual distance between the pair of nodes. Thus, the measured distance d_{ij} between nodes i, j is assumed to be drawn from

$$d_{ij} \sim \delta_{ij} + \delta_{ij} \mathcal{N}(0, e_r^2) \quad (9)$$

where δ_{ij} is the actual distance between nodes i, j , and $\mathcal{N}(0, e_r^2)$ denotes a zero-mean normal random variable of variance e_r^2 (henceforth referred to as *range error variance*). We also assume that the measurements are reciprocal (or symmetrized by averaging prior to further processing); i.e., $d_{ij} = d_{ji}$.

In this section, we derive the CRB for node localization using the above multiplicative normal noise model. Analogous derivations for different noise models [3], [8] can be found in [8]. An explanation of the difference between the received signal strength (RSS) noise model described therein and our multiplicative noise model can be found in the Appendix.

Define the vector of sensor parameters $\boldsymbol{\gamma} = (\gamma_1 \gamma_2 \cdots \gamma_N)$. Each γ_i contains the location coordinates for node i , i.e., $\gamma_i = (x_i, y_i)$ in the 2-D case. The unknown parameter vector for the $N - 3$ sensors whose locations are unknown² is defined as $\boldsymbol{\theta} = (\boldsymbol{\theta}_x \boldsymbol{\theta}_y)$, with $\boldsymbol{\theta}_x = (x_1, x_2, \dots, x_{N-3})$ and $\boldsymbol{\theta}_y = (y_1, y_2, \dots, y_{N-3})$. Sensors i, j perform pairwise observations d_{ij} . We assume that the observations d_{ij} are statistically independent for $i < j$. The density function of the observations d_{ij} given the locations of nodes i, j is denoted by $f(d_{ij} | \gamma_i, \gamma_j)$. Thus, the joint log-likelihood is

$$l(\mathbf{D}, \boldsymbol{\gamma}) = \sum_{i=1}^N \sum_{j \in H(i), j < i} l_{i,j} \quad (10)$$

$$l_{i,j} = \log f(d_{ij} | \gamma_i, \gamma_j)$$

where $H(i)$ is the set of nodes that are in the range of node i .

The CRB for coordinate θ_i is $\text{cov}(\theta_i) \geq [\mathbf{F}_{\boldsymbol{\theta}}^{-1}]_{ii}$, where $\mathbf{F}_{\boldsymbol{\theta}}$ is the Fisher information matrix (FIM), given by

$$\mathbf{F}_{\boldsymbol{\theta}} = \begin{bmatrix} \mathbf{F}_{\mathbf{x}\mathbf{x}} & \mathbf{F}_{\mathbf{x}\mathbf{y}} \\ \mathbf{F}_{\mathbf{x}\mathbf{y}}^T & \mathbf{F}_{\mathbf{y}\mathbf{y}} \end{bmatrix}. \quad (11)$$

The elements for the submatrix $\mathbf{F}_{\mathbf{x}\mathbf{x}}$ are given by

$$\mathbf{F}_{\mathbf{x}\mathbf{x}}(k, l) = \begin{cases} -\sum_{j \in H(k)} E \left[\frac{\partial^2}{\partial x_k^2} l_{k,j} \right], & k = l \\ -I_{H(k)}(l) E \left[\frac{\partial^2}{\partial x_k \partial x_l} l_{k,l} \right], & k \neq l \end{cases} \quad (12)$$

where $I_{H(k)}(l)$ is the indicator function (1 if l is in the range of k , 0 otherwise). Similar expressions hold for the $\mathbf{F}_{\mathbf{x}\mathbf{y}}$, $\mathbf{F}_{\mathbf{y}\mathbf{y}}$ submatrices. For the model in (9), these specialize to

$$\mathbf{F}_{\mathbf{x}\mathbf{x}}(k, l) = \begin{cases} \frac{1}{c_r^2} \sum_{j \in H(k)} \frac{(x_k - x_j)^2}{\delta_{kj}^4} (1 + 2e_r^2), & k = l \\ -\frac{1}{c_r^2} I_{H(k)}(l) \frac{(x_k - x_l)^2}{\delta_{kl}^4} (2e_r^2 + 4 - 3), & k \neq l. \end{cases} \quad (13)$$

The expression for $\mathbf{F}_{\mathbf{y}\mathbf{y}}(k, l)$ is the same, with the x 's replaced by y 's, whereas

$$\mathbf{F}_{\mathbf{x}\mathbf{y}}(k, l) = \begin{cases} \frac{1}{c_r^2} \sum_{j \in H(k)} \frac{(x_k - x_j)(y_k - y_j)}{\delta_{kj}^4} (1 + 2e_r^2), & k = l \\ -\frac{1}{c_r^2} I_{H(k)}(l) \frac{(x_k - x_l)(y_k - y_l)}{\delta_{kl}^4} (1 + 2e_r^2), & k \neq l. \end{cases} \quad (14)$$

V. SIMULATION RESULTS

In this section, we compare various MDS algorithms in the context of node localization in sensor networks. In addition to the aforementioned approaches, we also include in the comparison an iterative algorithm recently proposed by Costa *et al.* in [3]. Costa's algorithm is based on the principle of *majorization*. The idea behind majorization is simple. Instead of directly minimizing a complicated cost/stress function, majorization uses a simpler (usually quadratic) *majorizing* function that lies over the said cost/stress function and is equal to it at the current parameter estimate. Minimizing the majorizing function thus yields a new parameter estimate whose cost/stress is lower than or equal to that of the previous one. Continuing in this fashion yields a sequence of parameter estimates of decreasing cost/stress values. When the difference between the previous and the current cost values becomes smaller than a threshold ϵ the algorithm terminates. This is guaranteed due to the fact that a single iteration can reduce or maintain, but cannot increase the cost, which is also bounded from below. Costa's algorithm can be executed in a distributed fashion [3].

Network nodes are considered to be uniformly distributed in a square with area equal to 1, i.e., the x and y coordinates of the sensor nodes are uniformly distributed in $[0, 1]$. We employ the alignment procedure

²In the 2-D case, we need three anchor nodes.

TABLE I
COMPUTATIONAL COMPLEXITY ORDERS FOR FULL CONNECTIVITY (N IS NUMBER OF NODES, m IS NUMBER OF SPATIAL DIMENSIONS)

Algorithm	Complexity
Fastmap	$\mathcal{O}(mN)$
Fastmap+SD	$\mathcal{O}(pmN^2)$, $p \ll N$
PCA	$\mathcal{O}(N^3)$
Costa's	$\mathcal{O}(kmN^2)$, $k \ll N$

TABLE II
COMPUTATIONAL COMPLEXITY ORDERS FOR PARTIAL CONNECTIVITY (s IS THE AVERAGE NUMBER OF DISTANCE MEASUREMENTS COLLECTED BY A NODE)

Algorithm	Complexity
Fastmap	$\mathcal{O}(mN)$
Fastmap+SD	$\mathcal{O}(pmsN)$, $p \ll N$
Costa's	$\mathcal{O}(kmsN)$, $k \ll N$

described in [5], where necessary, in order to estimate the absolute coordinates, and adopt root mean-squared error (RMSE) as our estimation performance metric, as follows:

$$\text{RMSE} := \frac{\sum_{i=1}^N \sqrt{(x_{ri} - x_{ei})^2 + (y_{ri} - y_{ei})^2}}{N} \quad (15)$$

where x_{ei}, y_{ei} are the estimated coordinates, and x_{ri}, y_{ri} are the actual coordinates of sensor i . The computational complexity orders of the various algorithms under consideration are listed in Tables I and II, for the case of full and partial connectivity, respectively.

The baseline³ MDS algorithm performs PCA of the doubly centered matrix of squared distances, and is henceforth referred to as *PCA-based MDS*. For Costa's algorithm, we tried both random initialization and the alternative initialization strategy suggested in [3]. The former yields unsatisfactory results that do not improve with decreasing error variance; the latter often yields complex coordinates when the triangle inequality fails due to measurement errors. It is clear that Costa's algorithm is sensitive with respect to initialization and could benefit from a better "warm start." For this reason, we also tried using our adaptation of Fastmap to initialize Costa's iteration.

Fig. 3 shows the RMSE performance of the various algorithms (PCA, Fastmap, Fastmap + SD, Fastmap + Costa, and Costa with random initialization) for a network of 80 sensors, as a function of c_r^2 . Distance measurements were drawn from the multiplicative noise model in (9). The corresponding CRB is also plotted as a benchmark. For the SD step of the proposed algorithm (Fastmap + SD), a step-size of $\lambda = 0.01$ and $p = 10$ SD iterations were used. The convergence threshold in Costa's algorithm was set to $\epsilon = 0.1$. From Fig. 3, we observe that stand-alone Fastmap exhibits poor performance, which quickly degrades with increasing range error variance. When randomly initialized, Costa's algorithm also performs poorly in this setup, and its performance does not improve with decreasing error variance. Fastmap + SD and Fastmap + Costa are the best options from the viewpoint of RMSE performance, and remain relatively close to the CRB, especially for low-range error variance. Interestingly, the proposed algorithm is not only less complex, but also more accurate than PCA. This is partially attributed to the fact that PCA uses double centering, which colors the noise, whereas the proposed algorithm directly aims to minimize the stress function.

Fig. 4 shows corresponding results for a network of 200 nodes ($\lambda = 0.005$; the remaining setup is the same as in Fig. 3). The estimation

³PCA-based MDS is not directly applicable in the case of partial connectivity.

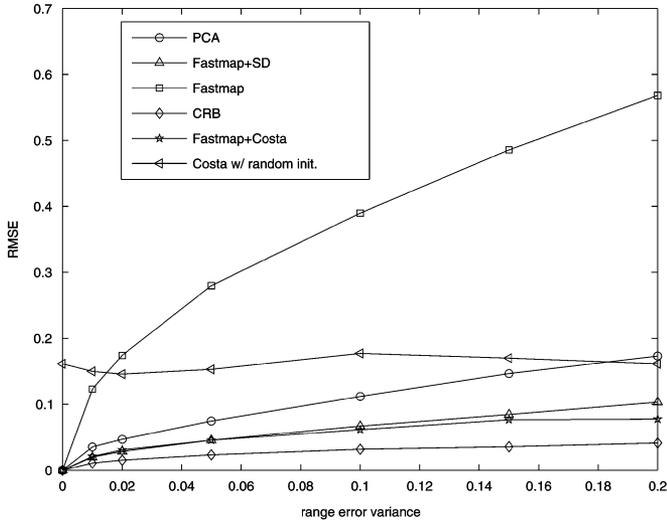


Fig. 3. RMSE performance versus measurement range error variance; $N = 80$, fully connected network, multiplicative normal measurement noise, 100 Monte Carlo runs.

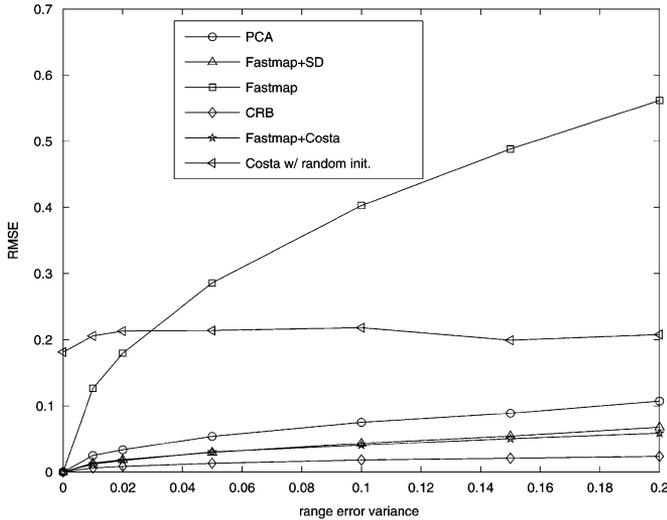


Fig. 4. RMSE performance versus measurement range error variance; $N = 200$, fully connected network, multiplicative normal measurement noise, 100 Monte Carlo runs.

accuracy of PCA, Fastmap + SD, and Fastmap + Costa, improves relative to Fig. 3, as expected. Fastmap does not benefit, due to the lack of (implicit or explicit) averaging, while Costa's algorithm with random initialization performs slightly worse than in Fig. 3.

We also tried an additive measurement noise model, and the results were qualitatively similar. The same is true when the multiplicative noise is log-normal, instead of normal.

Fig. 5 shows the average computational cost in floating-point operations (FLOPS) of Fastmap + SD and Fastmap + Costa, as a function of the number of nodes N . We observe that Fastmap + SD exhibits significantly lower complexity (up to five times lower) than Fastmap + Costa. The values of the step-size λ used for the different values of N are listed in Table III.

In all simulation results presented so far, the network was assumed to be fully connected, i.e., distance measurements were available for each pair of nodes in the network. We now switch to partially connected scenarios. We assume that nodes which are further apart than

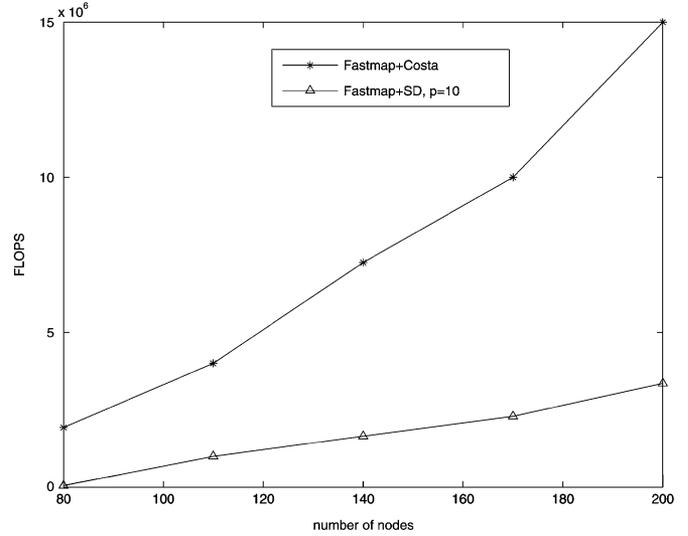


Fig. 5. Average computational cost in FLOPS versus number of nodes; fully connected network, $e_r^2 = 0.1$, 50 Monte Carlo runs. For Costa's algorithm, $\epsilon = 0.1$.

TABLE III
CHOICE OF STEP-SIZE λ AS A FUNCTION OF THE NUMBER OF NODES N

N	λ
80	0.01
110	0.0075
140	0.007
170	0.006
200	0.005

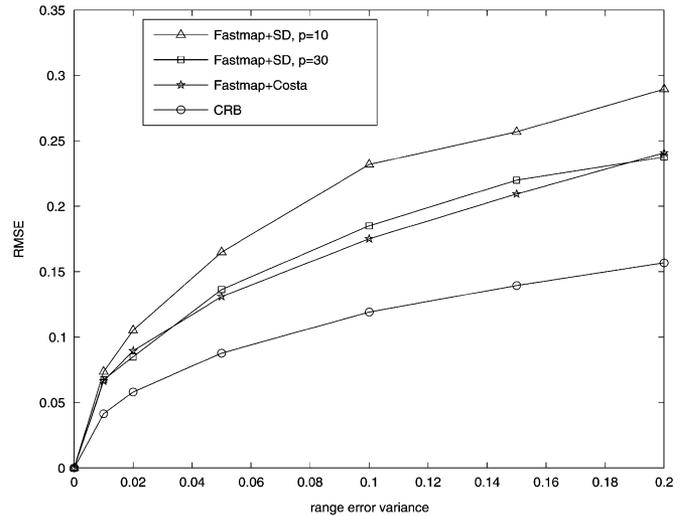


Fig. 6. RMSE performance and CRB for limited measurement range = 0.14 (weights corresponding to actual distances greater than this limit are set to zero); $N = 80$, $\lambda = 0.015$, 100 Monte Carlo runs. For Costa's algorithm, $\epsilon = 0.1$.

a certain threshold (radio range) cannot hear each other, the corresponding distance measurement is marked as unavailable, and the associated weight in the stress function is set to zero. An exception is that every node is assumed to be within range from each of the three anchor/pivot nodes. We adopt the multiplicative normal noise model in (9), and consider two cases: in the first, the measurement range is 0.14, and in the second it is 0.3. Figs. 6 and 7 show the RMSE performance of

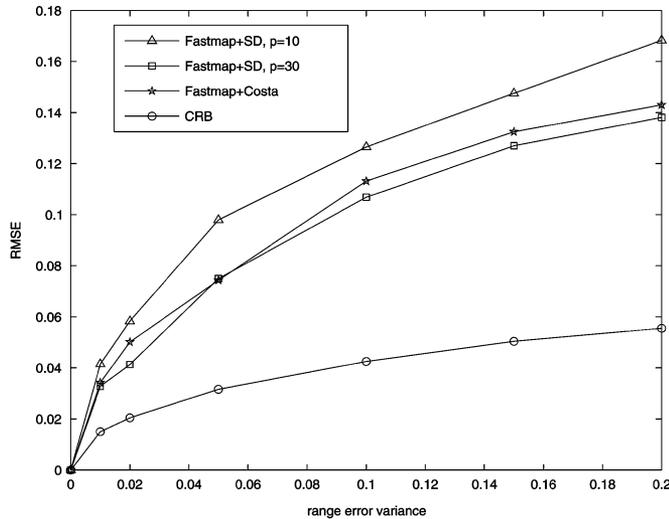


Fig. 7. RMSE performance and CRB for limited measurement range = 0.3; $N = 80$, $\lambda = 0.013$, 100 Monte Carlo runs. For Costa's algorithm, $\epsilon = 0.1$.

TABLE IV
CHOICE OF STEP-SIZE λ AS A FUNCTION OF
MEASUREMENT RANGE ($N = 80$ NODES)

Measurement Range	λ
Infinite	0.01
0.3	0.013
0.14	0.015

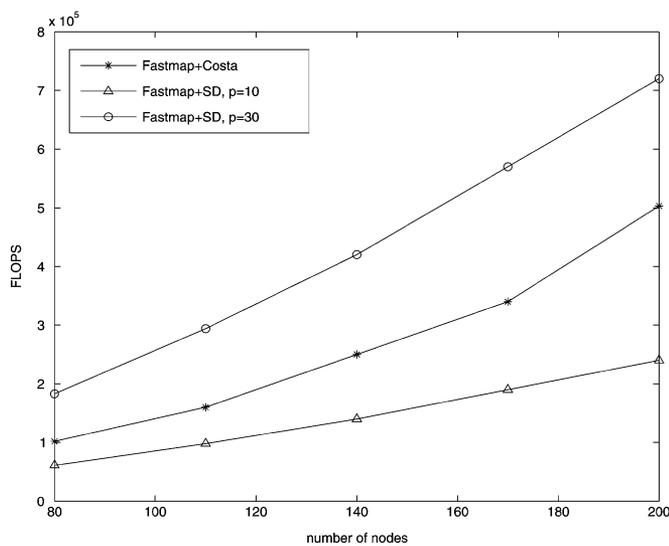


Fig. 8. Average computational cost in FLOPS versus number of nodes for limited measurement range = 0.14 (measurements corresponding to actual distances greater than 0.14 are missing); multiplicative normal measurement noise, $e_r^2 = 0.1$, 50 Monte Carlo runs.

Fastmap + SD, Fastmap + Costa, and the CRB (which accounts for the missing data) for the two cases, as a function of range error variance, for $N = 80$ nodes. Table IV lists the values of λ used in the SD iteration for the three different connectivity scenarios (fully connected, partially connected with measurement range equal to 0.3, or 0.14) and $N = 80$. For Fastmap + SD, we tried two different values for the number of SD iterations: $p = 10$ and $p = 30$. From Figs. 6 and 7, we observe

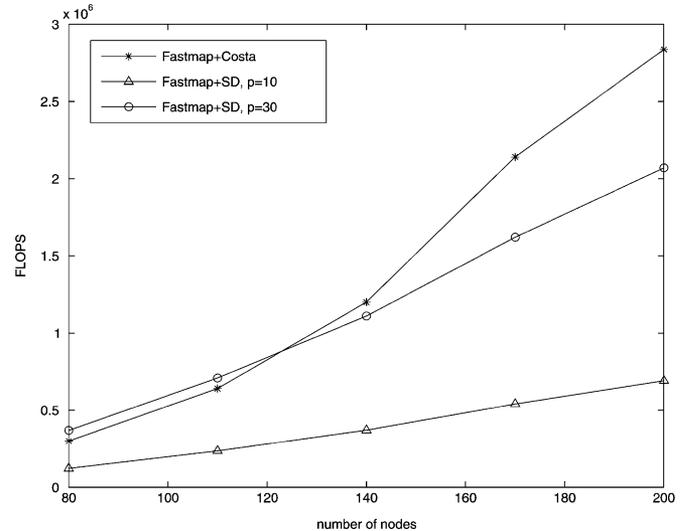


Fig. 9. Average computational cost in FLOPS versus number of nodes for limited measurement range = 0.3; multiplicative normal measurement noise, $e_r^2 = 0.1$, 50 Monte Carlo runs.

that the RMSE performance of Fastmap + Costa is better than that of Fastmap + SD with $p = 10$ iterations, and comparable to Fastmap + SD with $p = 30$ iterations. The corresponding FLOP counts in Figs. 8 and 9 show that Fastmap + SD with $p = 10$ maintains its computational complexity advantage compared to Fastmap + Costa, although the gap is somewhat smaller than in the case of full connectivity. Increasing p improves the RMSE performance of Fastmap + SD, but at the cost of computational complexity. For very sparse networks, it appears that Fastmap + Costa is preferable to Fastmap + SD if accuracy is the prime consideration.

VI. CONCLUSION

We have proposed a hybrid two-stage (Fastmap + SD) node localization algorithm that offers good performance at low complexity when every node can estimate its distance from the anchor nodes. The new algorithm employs Fastmap, coupled with judicious selection of anchor nodes that double as pivots, to generate a computationally cheap yet sufficiently accurate initialization for gradient descent. We also proposed using our adaptation of Fastmap as initialization for Costa's algorithm. Extensive simulations indicate that, in the context of our present application: i) Fastmap + SD uniformly outperforms the classical PCA-based MDS, both in terms of complexity and in terms of estimation accuracy; ii) our adaptation of Fastmap is an effective initialization for Costa's algorithm, leading to a substantial reduction in RMSE; and iii) Fastmap + SD and Fastmap + Costa yield comparable RMSE performance, but Fastmap + SD affords considerable reduction in computational complexity, especially for dense networks. For very sparse networks, it appears that Fastmap + Costa is preferable to Fastmap + SD if accuracy is the prime consideration. We have also derived the pertinent CRB for the multiplicative noise model in [2], [9], which was adopted for most of our simulations. Fastmap + SD and Fastmap + Costa operate close to the CRB for dense networks, but there is a measurable performance gap for sparse networks.

Wireless sensor nodes are battery-operated, so energy is an issue. Energy spent in communication typically far outweighs that spent in processing. Energy costs associated with communication between nodes can be controlled by restricting communication to nearby neighbors, and developing distributed implementations. The Fastmap step of the proposed algorithm is naturally distributed, whereas

distributed implementation of gradient descent can be developed along known lines [1], [7].

APPENDIX
NORMAL VERSUS LOG-NORMAL
MULTIPLICATIVE NOISE MODELING

In [3] and [8], the power received at node i from node j , measured in decibels (dB), is modeled as $P_{ij} = \bar{P}_{ij} + v$, where \bar{P}_{ij} is the mean power, and v is a zero-mean Gaussian random variable of standard deviation σ . The mean power is modelled as $\bar{P}_{ij} = P_0 - 10n_p \log_{10}(\delta_{ij}/\delta_0)$, where P_0 is the mean power for a reference distance, δ_0 , and n_p is the path loss exponent. It follows that

$$P_0 - P_{ij} = P_0 - \bar{P}_{ij} - v = 10n_p \log_{10} \frac{\delta_{ij}}{\delta_0} - v \quad (16)$$

and the associated distance estimate is given by [3]

$$d_{i,j} = \delta_0 10^{(P_0 - P_{ij})/10n_p}. \quad (17)$$

Substituting $P_{ij} = \bar{P}_{ij} + v$ and $\bar{P}_{ij} = P_0 - 10n_p \log_{10}(\delta_{ij}/\delta_0)$ yields

$$d_{ij} = \delta_{i,j} 10^{-v/10n_p}. \quad (18)$$

Notice that the noise factor is *log-normal*, whereas in the model of [2] and [9] (also adopted herein), the noise factor is normally distributed.

REFERENCES

- [1] D. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd ed. Belmon, MA: Athena Scientific, 1997.
- [2] P. Biswas, T.-C. Liang, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sensor Netw.*, vol. 2, no. 2, pp. 188–220, May 2006.
- [3] J. A. Costa, N. Patwari, and A. O. Hero, "Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks," *ACM Trans. Sensor Netw.*, vol. 2, no. 1, pp. 39–64, Feb. 2006.
- [4] C. Faloutsos and K.-I. Lin, "FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," in *Proc. ACM SIGMOD*, 1995, vol. 24, no. 2, pp. 163–174.
- [5] X. Ji and H. Zha, "Sensor positioning in wireless ad hoc sensor networks using multidimensional scaling," in *Proc. Infocom*, 2004, pp. 2652–2661.
- [6] J. B. Kruskal and M. Wish, *Multidimensional Scaling*. Beverly Hills, CA: Sage, 1978.
- [7] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 1439–1451, Aug. 2006.
- [8] N. Patwari, A. Hero, 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.
- [9] 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, Nov. 2004.
- [10] W. S. Torgerson, "Multidimensional scaling: I. Theory and method," *Psychometrika*, vol. 17, pp. 401–419, 1952.
- [11] W. S. Torgerson, "Multidimensional scaling of similarity," *Psychometrika*, vol. 30, pp. 379–393, 1965.