Topology Preserving Maps—Extracting Layout Maps of Wireless Sensor Networks From Virtual Coordinates

Dulanjalie C. Dhanapala and Anura P. Jayasumana, *Senior Member, IEEE, Member, ACM*

***Abstract*—A method for obtaining topology-preserving maps (TPMs) from virtual coordinates (VCs) of wireless sensor networks is presented. In a virtual coordinate system (VCS), a node is identified by a vector containing its distances, in hops, to a small subset of nodes called anchors. Layout information such as physical voids, shape, and even relative physical positions of sensor nodes with respect to x-y directions are absent in a VCS description. The proposed technique uses Singular Value Decomposition to isolate dominant radial information and to extract topological information from the VCS for networks deployed on 2-D/3-D surfaces and in 3-D volumes. The transformation required for TPM extraction can be generated using the coordinates of a subset of nodes, resulting in sensor-network-friendly implementation alternatives. TPMs of networks representing a variety of topologies are extracted. Topology preservation error $(E_{\rm TP})$, a metric that accounts for both the number and degree of node flips, is defined and used to evaluate 2-D TPMs. The techniques extract TPMs with $(E_{\rm TP})$ less than 2%. Topology coordinates provide an economical alternative to physical coordinates for many sensor networking algorithms.**

***Index Terms*—Localization, routing, Singular Value Decomposition (SVD), topology preserving map (TPM), virtual coordinates (VCs), wireless sensor network (WSN).**

I. Introduction

VIRTUAL coordinates (VCs) provide an economical alternative to geographical coordinates for routing and self-organization of large-scale wireless sensor networks (WSNs). Geographical coordinate-based protocols such as Geographical Routing (GR) require physical location of nodes, which may be obtained by GPS or a localization algorithm. Use of GPS is infeasible or too costly for many applications, while localization using analog measurements such as signal strength and time delay is difficult and prone to errors [19], [25], [26], [30]. Signal strength is susceptible to noise, fading, and interferences due to multipath and other devices. Need for accurate power control and signal strength measurements contribute to increased hardware complexity as well as cost. Routing is carried out using directional information derived from geographic coordinates (GCs), and hence concave physical voids in the network degrade the performance of GR schemes. Anchor-based virtual coordinate systems (VCSs) characterize each node by a coordinate vector consisting of the shortest-path hop distances to a set of anchors [5], [6], [23], [24], [27]. These anchors are a set of ordinary sensor nodes with no additional capabilities. Coordinates can be obtained using a controlled/organized flooding mechanism [20] initiated by the anchors. VCS is a higher-dimensional abstraction of a partial connectivity map of sensors. It has several properties, such as ease of generation and facilitating connectivity-based routing without the need for geographical information [5]–[8], [25], that make it attractive for large-scale or resource-starved WSNs. The number of anchors becomes the network's dimensionality in the VC space. As the network's connectivity information is embedded in VCs, the physical voids become transparent in the virtual space (VS). However, VCs lose the directional information related to node positions. The number of anchors required and their placement for a given network play a crucial role in the performance of VC-based routing. However, identification of the optimal number of anchors and proper anchor placement remain major challenges. Underdeployment of anchors causes identical node coordinates, while their overdeployment and improper placement worsen the local minima problem, causing logical voids [6]

Many disadvantages associated with VCS in comparison to geographical coordinate systems are due to the lack of information about the physical network topology and layout. As each virtual ordinate propagates radially away from an anchor, the directional information of a node with respect to the anchor is lost. Thus, the physical layout information such as physical voids, relative physical direction information of sensor nodes with respect to x-y positions, and even explicit connectivity information among pairs of nodes are absent in a VCS description. The above information can be revealed if the physical map is available. Having both, partial connectivity information that is embedded in VCs and position or direction information as in geographical coordinates can be used to overcome the disadvantages in each other's domains. However, physical topological information has to be generated without inheriting the disadvantages associated with obtaining physical location information or localization.

Obtaining a topology map resembling the physical layout topology of a network from the set of VCs that is based only on hop distances to a small set of anchors has not been possible up to now. In this paper, we present a technique to obtain topology preserving maps (TPMs) that contain the topology

Manuscript received December 25, 2012; accepted April 04, 2013; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C. Westphal. Date of publication June 18, 2013; date of current version June 12, 2014. This work was supported in part by the NSF under Grant CNS 0720889.

The authors are with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523 USA (e-mail: dulanjalie.dhanapala@colostate.edu; anura.jayasumana@colostate.edu).

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/TNET.2013.2263254

of a network and physical features, including its geographical voids, boundary profiles, and relative Cartesian directional information. TPMs overcome many of the disadvantages of VCS compared to geographical coordinate systems, but without inheriting its disadvantages, while preserving all the advantages of connectivity-based VCs.

A TPM is a rotated and/or distorted version of the real physical node map to account for connectivity information inherent in VCs. The topological coordinates provided by the proposed method are a good substitute for geographical coordinates for many applications that depend on connectivity and location information. In fact, the topological coordinates (TCs) in conjunction with VCs from which they are derived have been demonstrated to be better than geographical coordinates for routing due to significantly enhanced routing performance [12]. Topology coordinate space provides an alternative that is different from virtual and physical coordinates, yet preserving the advantages of the two. Boundary node identification, event region and void detection [10], and nodes gaining network awareness [11], i.e., finding the overall shape of the network and its place in the network, are among examples of techniques that have been demonstrated to benefit from the TCs. The results presented here demonstrate the ability to determine and visualize the structural characteristics of large-scale WSNs in both 2-D and 3-D. Ability to do such visualization without the need for analog measurement capability at nodes will be invaluable for networks whose nodes are extremely limited in capability, e.g., large-scale nanosensor networks [1]. Even though we focus on WSN context here, the technique is applicable to a broader class of networks.

Next, Section II reviews the background. After presenting the Singular Value Decomposition (SVD)-based method for obtaining TPMs in Section III, we also refine the method to reduce its complexity. A performance evaluation metric for topology maps is presented in Section IV. In Section V, we discuss the results of three alternatives for TPM generation, with different computational and communication complexities. Section VI addresses implementation issues. Finally, Section VII discusses the future work and concludes our work.

II. Background

We briefly review the related work on coordinate systems and localization techniques for generating GCs and maps, for which proposed TPMs are a competitive, economical alternative. The term TPMs has been used in contexts outside sensor networking, such as multidimensional data organization. Though some of them are not directly applicable to WSNs, we review the most relevant ones to place the proposed scheme in context.

A. Geographic Routing (GR) Versus Virtual Coordinate Routing (VCR)

In geographic routing, the physical location of nodes is used for node addressing as well as for routing. A packet is forwarded in the direction of the destination, and thus GR gets disrupted by geographical voids. Concave voids are especially difficult to overcome. Greedy Perimeter Stateless Routing (GPSR) [14] makes greedy forwarding decisions until it fails, for example due to a geographical void, and attempts to recover by routing around the perimeter of the void. Greedy Other Adaptive Face Routing (GOAFR) [16] is a geometric *ad hoc* algorithm combining greedy forwarding and face routing to overcome the local minima issue. Greedy Path Vector Face Routing with Path Vector Exchange (GPVFR/PVEX) [18] is similar to [16], but it requires the network's planar graph.

VC-based schemes, where each node is characterized by a coordinate vector corresponding to hop distances to a set of anchors, uses a distance measure in VCS to identify the node for packet forwarding. VCR scheme in [27], e.g., uses all the perimeter nodes as anchors. When a packet reaches a local minima, an expanding ring search is performed until a closer node is found or time to live (TTL) expires. In VC assignment protocol (VCap), the coordinates are defined based on hop distances [5]. At local minima, VCR causes a packet to follow a rule called "local detour." In Logical Coordinate-based Routing (LCR) [6], backtracking is used when greedy forwarding fails at a local minimum. Aligned VC system (AVCS) [21] reevaluates VCs by averaging a node's own coordinate with neighboring coordinates in an attempt to overcome local minima. Convex Subspace Routing [8] overcomes the local minima by using a subset of anchors for routing and by dynamically changing the subset to provide a convex distance surface for routing. In Axis-Based VC Assignment Protocol (ABVCap) [33], each node extracts a 5-tuple VC corresponding to longitude, latitude, ripple, up, and down. Existing VCR protocols rely mainly on Greedy forwarding, followed by a backtracking scheme to overcome the local minima issue. Geo-Logical Routing (GLR) [12] is a novel scheme that combines the advantages of VCS and TPM proposed in this paper to overcome disadvantages of each other's domain, thus impressively outperforming existing VCR schemes as well as GPSR, which requires physical coordinates.

B. Localization

We focus on relative localization techniques, as global localization is realizable through relative localization and the actual positions of a subset of nodes or physical anchors. Centralized and distributed algorithms are available for relative localization. Distributed algorithms use received signal strength indication (RSSI), radio hop count, time difference of arrival, and/or angle of arrival for relative localization. RSSI uses signal strength to estimate the distance between nodes, while radio hop count uses hop distance. The latter uses a probabilistic correction equation to approximate the hop distance to real distance [2], [32]. Disadvantages of RSSI measurements include sensitivity to terrain [26] and large variations due to fading and interference. Relationship between RSSI and distance is very difficult to predict indoors [19] as well as in complex outdoor environments due to absorption and reflection of signals and propagation characteristics over different terrains. No robust and scalable algorithms are available for localization of nodes deployed on surfaces of complex 3-D structures. An RSSI measurement-based distributed algorithm using triangulation for localization of 2-D and 3-D WSNs is proposed in [35].

Centralized algorithms for localization of 2-D networks include Semidefinite Programming (SDP) and MDS-MAP [2], [29]. The former algorithm develops geometric constraints between nodes, represents them as linear matrix inequalities (LMIs), and then simply solves for the intersection of the constraints. Unfortunately, not all geometric constraints can be expressed as LMIs, which precludes the algorithm's use in practice. MDS-MAP uses Multi-Dimensional Scaling (MDS) based on connectivity information.

The localization scheme in [17] first selects a subset of boundary nodes as landmarks. Next, Delaunay triangles are generated based on Voronoi cells formed with landmarks. Finally, the network layout is discovered based on the landmarks' locations. Boundary nodes need to be identified accurately without physical information, and an incremental algorithm is required to combine the Delaunay triangles.

Factors that contribute to errors in localization include inaccuracies in distance estimate, the position calculation, and the localization algorithm [25]. How the localization error propagates and accumulates in a network is illustrated in [25] in terms of geographic distribution of the error, correlation, mean error, and probability distribution of the error. This study shows that routability of GR with GEAR [34] falls significantly, and the percentage of deliveries to wrong destinations increases as the error in localization increases.

As both VCS and topology maps are generated based on the hop distances, they are not affected by fading or signal strengths. Furthermore, they do not rely on analog measurements such as RSSI or time delay, and thus do not have cumulative errors that affect the performance as the networks scale.

C. Mapping Schemes for Networks and Data

TPMs discussed in this paper deviate from the localization maps. The relative localization schemes expect the relative distances to be accurate. Thus, given the absolute position of a subset of nodes, global localization is realizable. In contrast, in topology maps, what is important is the topology preservation, not the physical distances. The derived topology should be homeomorphic (topologically isomorphic) to the physical layout of the sensor network—i.e., between two topological spaces, there has to be a continuous inverse function. In our case, it is a mapping, which preserves the topological properties of the physical network topology.

In the context of analysis of high-dimensional data, unsupervised learning algorithms have been proposed that use eigenvalue decomposition for obtaining a lower dimensional embedding of the data. Here, we discuss four such schemes: MDS, Local Linear Embedding (LLE), Isomap, and Laplacian Eigenmaps (LE) [3]. None of these methods is designed for, nor is suitable for, resource-starved WSNs.

MDS [29], [31] is a commonly used statistical technique in information visualization for exploring similarities or dissimilarities in higher-dimensional data from the complete distance matrix (similarity matrix) D, which is defined as the matrix of all the pairwise distances between points/nodes. $D = [d_{ij}]_{N \times N}$, where N is the number of nodes in the network and d_{ij} is the distance from node i to node j with $d_{ij} = d_{ji}, \geq 0$, and $d_{ij} = 0$. In general d_{ij} can be any distance metric, but there is a possibility for the algorithm to fail if d_{ij} is not the Euclidean distance. Generating D based on hop distances requires all the nodes in a WSN to serve as anchors, an extremely expensive proposition that calculates and stores information about the distances between each pair of nodes. If such information were available at each node, 100% routing can be achieved just by following the ordinate corresponding to the destination, i.e., without the need for the topology map. MDS is therefore not practical or applicable for generating TPMs of WSN. Our novel method, based on SVD, generates topology maps of 2-D and 3-D networks, using a set of M anchors, where $M \ll N$, and N is the number of nodes.

Isomaps [32] is an extension of MDS to geodesic distance-based topology map generation. Again, the geodesic distances are actual distances among nodes, which require expensive error prone distance estimators such as RSSI or time of arrival (TOA). Furthermore, if a node has the information of entire network, 100% routability is achievable without need for a topology map. Moreover, LLE and LE both use an iterative approach to preserve the neighborhood distances, the realization of which is infeasible in energy-limited WSNs.

All the four schemes rely highly on physical distances between all the possible pairs of nodes, and thus require localization approaches. Accuracy of both central and distributed implementations of localization is highly sensitive to channel fading and signal-to-noise ratio (SNR).

III. Topology Preserving Maps for 2-D and 3-D WSNs

A novel technique for obtaining a TPM representation of a sensor network from its VC set is presented next. The objective is to characterize each node with an (x, y) coordinate pair, or (x, y, z) in the case of 3-D WSNs, that results in a TPM that is homeomorphic to the network's physical layout and preserves information about node connectivity, physical layout, and physical voids. We emphasize that the map so obtained is not the physical map, but is a distorted version resembling it, which takes the connectivity into account. The metromap of a metro system versus its actual physical map drawn to scale can be considered analogous to the TPM-versus-physical-map relationship. The metromap, though it does not have the exact physical dimensions, is in fact much more useful for the purpose of navigation. Similarly, the TPMs have been shown elsewhere to be much more effective for many sensor-network-related functions, e.g., routing [12], boundary detection [10], and achieving node awareness [11]. In fact, the TCs of TPM can be used as a substitute for GCs in many GC-based algorithms.

Section III-A develops the technique by starting with the VCs of all the nodes to obtain a TPM. Section III-B discusses the extension of TPMs to 3-D networks. A significantly more efficient version of the technique that uses information of only a small subset of nodes to evaluate the transformation matrix is presented in Section III-C. Finally, Section III-D proposes a method of calculating a node's Cartesian coordinates with lower computational complexity. Notations used in the text are summarized in Table I.

A. 2-D Topology Preserving Maps From VCs

Consider a 2-D sensor network with N nodes and M anchors. Thus, each node is characterized by a VC vector of length M. Let P be the $N \times M$ matrix containing the VCs of all the nodes, i.e., the ith row corresponds to the M-long VC vector of the ith node, and the jth column corresponds to the virtual ordinate of all the nodes in the network with respect to the jth anchor. Therefore

$$P = [h_{n_i A_j}]$$

where $h_{n_i A_j}$ is the hop distance from node n_i to anchor A_j. For sensor network applications, it is generally desirable to have only a small subset of nodes as anchors, i.e., $M \ll N$,. A 2-D network has an M-dimensional representation under the VC

TABLE I
NOTATION USED IN THE TEXT

Notation	Description
$\mathcal{N}$	Set of nodes
$\boldsymbol{N}=\vert\mathcal{N}\vert$	Number of network nodes
$\boldsymbol{n_i} \in\mathcal{N}$	Node i
$\mathcal{A}\subset\mathcal{N}$	Set of anchor nodes
$\boldsymbol{M} =\vert\mathcal{A}\vert\,(\boldsymbol{M} << \boldsymbol{N})$	Number of anchors
$\boldsymbol{A_i}\ \boldsymbol{\epsilon}\mathcal{A}\,, \boldsymbol{i} = \mathbf{1}{:}\,\boldsymbol{M}$	i^{th}anchor
$\boldsymbol{h_{n_in_j}}$	Minimum hop distance between nodes n_i, n_j
$\boldsymbol{P_{N\times M}}$	Virtual coordinate matrix of the entire network
$\boldsymbol{P}_{(\boldsymbol{i})} = [\boldsymbol{h_{n_iA_1}}, \ldots, \boldsymbol{h_{n_iA_M}}]$	VCs of Node n_i
$\boldsymbol{Q_{R\times M}}\ ; \boldsymbol{R} << \boldsymbol{N}$	VCs of a subset of nodes
$\boldsymbol{P_{SVD}}^{(\boldsymbol{i})}$	i^{th} principle component of P
$[\boldsymbol{X_T}, \boldsymbol{Y_T}]$, $[\boldsymbol{X_T}, \boldsymbol{Y_T}, \boldsymbol{Z_T}]$	Topological coordinate matrix of a 2-D network Topological coordinate matrix of a 3-D network
$[\boldsymbol{X_T}, \boldsymbol{Y_T}]_{(\boldsymbol{i})} = [\boldsymbol{X_{T,i}}, \boldsymbol{Y_{T,i}}]$ $[\boldsymbol{X_T}, \boldsymbol{Y_T}, \boldsymbol{Z_T}]_{(\boldsymbol{i})} = [\boldsymbol{X_{T,i}}, \boldsymbol{Y_{T,i}}, \boldsymbol{Z_{T,i}}]$	Topological coordinates of node n_i of a 2-D network Topological coordinates of node n_i of a 3-D network
$[\boldsymbol{X_p}, \boldsymbol{Y_p}]$, $[\boldsymbol{X_p}, \boldsymbol{Y_p}, \boldsymbol{Z_p}]$	Physical coordinates of a 2-D network Physical coordinates of a 3-D network

transformation. The main goal thus is to extract the 2-D representation of the network from this M-D space. SVD [15] of P is denoted as

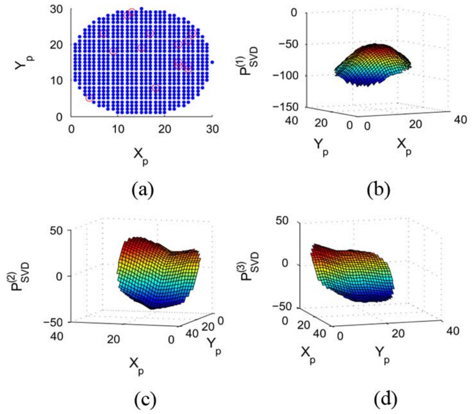
$$P = U \cdot S \cdot V^T \tag{1}$$

where U, S, and V are $N \times N, N \times M$, and $M \times M$ matrices, respectively [14]. U and V are unitary matrices, i.e., $U^TU = I_{N\times N}$ and $V^TV = I_{M\times M}$. SVD extracts and packages the salient characteristics of the dataset P providing an optimal basis for P. Moreover V is an optimal basis of P^T, i.e., V spans R^M. Let us consider the principle components (PCs) of P

$$P_{\text{SVD}} = U \cdot S. \tag{2}$$

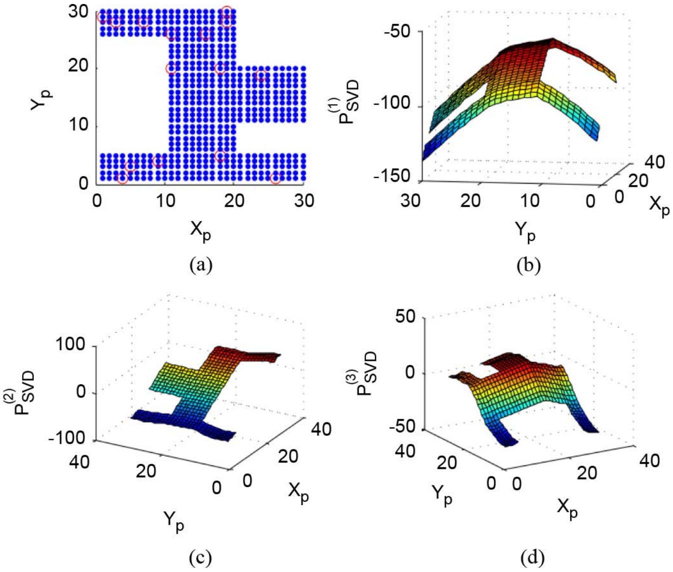
P_{SVD} is an $N \times M$ matrix that describes each node with a new set of M-length coordinate vectors. It gives the coordinates for the data in P under the new basis V. As S is a diagonal matrix with diagonal elements being the singular values of P arranged in their descending order, elements in S provide unequal weights on columns of U. Using the unitary property of V, it is also the projection of P onto V [15], i.e.,

$$P_{\text{SVD}} = P \cdot V. \tag{3}$$



(a) (b) (c) (d)

Fig. 1. (a) Circular network of 707 nodes with 15 anchors. (b)–(d) First three PCs $P^{(1)}_{\text{SVD}}$, $P^{(2)}_{\text{SVD}}$, and $P^{(3)}_{\text{SVD}}$ plotted against the physical positions. Randomly selected anchors are marked in circles.



(a) (b) (c) (d)

Fig. 2. (a) Odd-shaped network with 550 nodes with 15 anchors. (b)–(d) First three PCs $P^{(1)}_{\text{SVD}}$, $P^{(2)}_{\text{SVD}}$, and $P^{(3)}_{\text{SVD}}$ plotted against the physical positions. Randomly selected anchors are marked in circles.

The columns of P_{SVD}, i.e., the PC values of the VC set are arranged in the descending order of information about the original coordinate set. The first PC captures the highest variance of the data set, and each succeeding component has the highest variance possible under the constraint that it be orthogonal to the preceding components.

Figs. 1 and 2 show, for two different networks, the variation over the physical layout of the first three PCs, i.e., columns of P_{SVD} given by (3), plotted against the corresponding physical positions of the nodes. As observed in [9], the first three SVD components dominate in magnitude over the remaining PCs, which are similar to Fourier basis vectors.

The set of VCs has the connectivity information embedded in it, though it has no directional information. All the nodes that are h hops away from the jth anchor have h as the jth ordinate. Each ordinate propagates as a concentric circle centered at the corresponding anchor, while the angular information is completely lost. Note that the most significant ordinate based on SVD, i.e., the first column of P_{SVD} shown in Figs. 1(b) and 2(b),

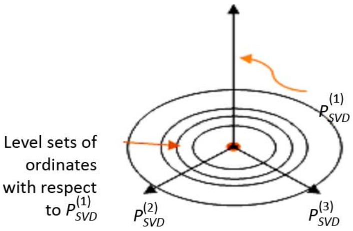
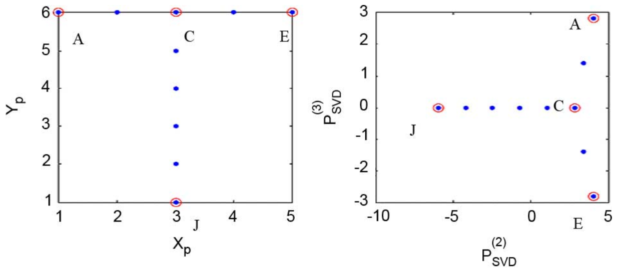


Fig. 3. Nature of principal component directions derived from VC.



(a) (b)

Fig. 4. (a) Physical map of a T-shaped example network. (b) Topology map of the network in (a).

is a convex surface centered at some point within the network, thus capturing the resultant effect of the conical propagation of different anchor coordinates. As SVD provides an orthonormal basis, second and third ordinates are orthogonal to the first ordinate while being perpendicular to each other as illustrated in Fig. 3. Plots of the variation of second and third components for the two example networks in Figs. 1 and 2 illustrate this as well. Therefore, the second and third columns of P_{SVD} provide a set of two-dimensional Cartesian coordinates for node positions, less encumbered by the dominant radial information in VCs, which was captured by the first column. Thus, instead of the M coordinates of a row of P_{SVD} to characterize a node, the second and third columns are used as Cartesian coordinates for plotting an approximate map of the network, i.e.,

$$[X_T, Y_T] = [P^{(2)}_{\mathrm{SVD}}, P^{(3)}_{\mathrm{SVD}}] \tag{4}$$

where $P^{(j)}_{\mathrm{SVD}}$ is the jth column of P_{SVD}. $[X_T, Y_T]$ is the Cartesian coordinate matrix of entire node set, i.e., its ith row, $[X_T, Y_T]_{(i)}$, is used as the Cartesian coordinates of ith node.

We now illustrate the procedure using the T-shaped network of 10 nodes shown in Fig. 4(a) as an example. Physical coordinates $[X_P, Y_P]$ and VC matrix P with respect to anchors A, C, E, and J are given in Table II. SVD evaluation as in (1) provides V as

$$V = \begin{pmatrix} -0.55 & -0.34 & -0.71 & -0.30 \\ -0.34 & -0.25 & 0.00 & 0.91 \\ -0.55 & -0.34 & 0.71 & -0.30 \\ -0.54 & 0.84 & 0.00 & 0.03 \end{pmatrix}.$$

P_{SVD} can now be evaluated using (3), and thus topological coordinates of nodes are given by (4). $[P^{(2)}_{\mathrm{SVD}}, P^{(3)}_{\mathrm{SVD}}]$ is tabulated in Table II and plotted in Fig. 4(b).

TABLE II
COORDINATES, VCS, AND TOPOLOGICAL COORDINATES FOR THE NETWORK IN FIG. 4(a)

ID	X_P	Y_P	P				P_{SVD}			
			A	C	E	J	$P^{(1)}_{SVD}$	$P^{(2)}_{SVD}$	$P^{(3)}_{SVD}$	$P^{(4)}_{SVD}$
A	1	6	0	2	4	7	-6.62	4.06	2.83	0.82
B	2	6	1	1	3	6	-5.74	3.46	1.41	-0.12
C	3	6	2	0	2	5	-4.87	2.87	0.00	-1.05
D	4	6	3	1	1	6	-5.74	3.46	-1.41	-0.12
E	5	6	4	2	0	7	-6.62	4.06	-2.83	0.82
F	3	5	3	1	3	4	-5.76	1.10	0.00	-0.76
G	3	4	4	2	4	3	-6.66	-0.66	0.00	-0.47
H	3	3	5	3	5	2	-7.55	-2.43	0.00	-0.19
I	3	2	6	4	6	1	-8.45	-4.19	0.00	0.10
J	3	1	7	5	7	0	-9.35	-5.96	0.00	0.39

A question that naturally arises is why the removal of the largest PC and keeping less significant second and third terms, which in many applications of PCA correspond to error or deviations, yield a good layout map. First, we address why the dominant PC cannot be used. Note that our dataset consists of the VCS of a network. Therefore, if we were to reconstruct the VCS using a subset of PCs, the first component would be the most important one. However, the reconstruction space we seek is an approximation to the physical layout, and the mapping from the physical layout to the acquisition space (VCS) is highly nonlinear. Each VC propagates concentrically w.r.t. to the corresponding anchor, i.e., all the nodes at a minimum distance r from the anchor map to the value r.

The impact of conical nature of propagation of each VC from the corresponding anchor and the net effect of many such anchors completely dominate over any cardinal direction information embedded in VCS. In Appendix-A, we show the convexity of the first PC for a simple 1-D network and extend the result to a 2-D network. This convex nature is also evident from Figs. 1(b) and 2(b). As the first PC contains much of the dominant convex form of distortion introduced in going from physical layout to the VC space, using it as a major axis for mapping produces maps with a significant amount of folding. Appendix-A also illustrates the resulting folding of the map, both for a 1-D example and the simple 2-D example of Fig. 1(a), when the first PC is used for reconstruction.

Based on the assertion above that the 1st PC contains much of the radial distortion introduced in physical layout to VC mapping, it follows that the removal of the first PC from consideration removes much of this radial information from the data set. This is evident in the second and third PC plots in Figs. 1(c), 1(d), 2(c), and 2(d) for the two examples. The remaining PCs contain the information from the physical layout that was masked due to this convex distortion in the original VC set. Therefore, as specified in (4), we use the next two significant components to yield the cardinal directional information to plot the layout maps. SVD-based construction assures that these two components, i.e., second and third PCs, form an orthogonal Cartesian plane for network. Note that these maps are not physical layout maps; rather, they are distorted maps that preserve much of the topological relationships of the network layout.

These Cartesian coordinates are estimated without having any kind of physical directional or positioning information beyond the radial information (hop distance) with respect to the anchors. Results presented later demonstrate that a TPM thus obtained preserves the topological characteristics of the original network. One can even identify features such as physical voids that were not apparent in the VC-based description.

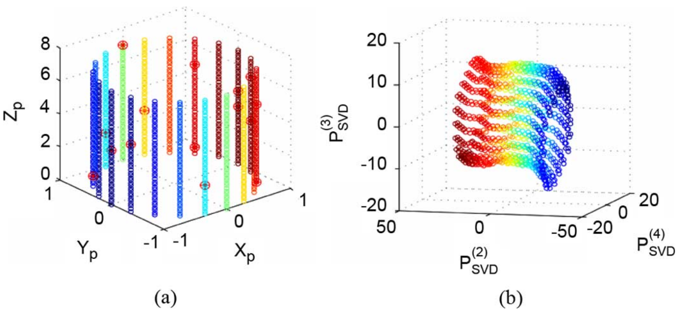


Fig. 5. (a) Network on a cylindrical surface (900 nodes). Randomly selected 20 anchors are marked in circles. (b) Topology map of (a).

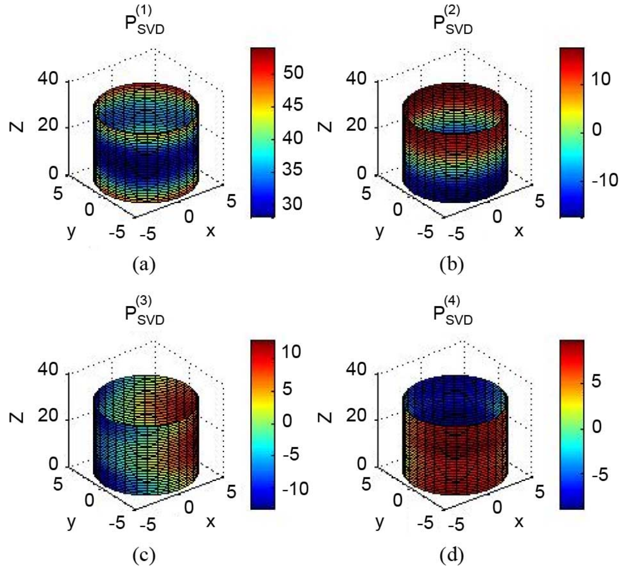


Fig. 6. First four PCs (a) $P^{(1)}_{\mathrm{SVD}}$, (b) $P^{(2)}_{\mathrm{SVD}}$, (c) $P^{(3)}_{\mathrm{SVD}}$, and (d) $P^{(4)}_{\mathrm{SVD}}$ of a cylindrical network plotted as a color map on the surface of the network.

A formal general proof as to how the second and third PCs preserve many of the physical features of the network layout has so far eluded us. The above description is the best intuitive explanation we have based on analysis of many data sets.

B. 3-D Topology Preserving Map From VCs

Sensor networks may be deployed within 3-D volumes, on 3-D surfaces, or a combination of those. Here, we consider sensors deployed on a 3-D surface, which may even wrap around, thus affecting VC propagation in complex ways. Consider the uniform cylindrical surface shown in Fig. 5(a) on which 900 nodes are deployed. Fig. 6(a)–(d) shows the plots of the first four PCs for each node in the network. They are denoted by $P^{(1)}_{\mathrm{SVD}}$, $P^{(2)}_{\mathrm{SVD}}$, $P^{(3)}_{\mathrm{SVD}}$, and $P^{(4)}_{\mathrm{SVD}}$, respectively. As SVD provides an orthonormal basis, the second, third, and fourth PCs are orthogonal to the first ordinate while being perpendicular to each other. Like with the 2-D case, the salient feature of the VCS, i.e., the radial propagation of coordinates, is captured by the first PC as seen in Fig. 6(a). The value is lowest at the center of the surface and increases toward edges, resulting in a convex variation along the height. Thus, removing it from further consideration allows us to uncover linear patterns embedded in the VC set. As seen in Fig. 6(b), the second PC varies monotonically along the height of the cylinder, thus it can be used to obtain the z-coordinate for the topology map. More interestingly, third and fourth PCs, which are taken as x- and y-coordinates, directionally distribute in such a way that they are orthogonal to each other while being normal to the second PC. The resulting TPM is illustrated in Fig. 5(b). TPM generation on 3-D surfaces can thus be done by ignoring the first PC and by taking the second, third, and fourth columns of P_{SVD} to provide a set of 3-D Cartesian coordinates.

To summarize, the topological coordinates of node n_i for the 3-D case are given by

$$\begin{aligned}[X_T, Y_T, Z_T]_{(i)} &= [P^{(2)}_{\mathrm{SVD}}, P^{(3)}_{\mathrm{SVD}}, P^{(4)}_{\mathrm{SVD}}]_{(i)} \\ &= [P_{(1)}\cdot V^{(2)}, P_{(i)}\cdot V^{(3)}, P_{(i)}\cdot V^{(4)}] \quad (5)\end{aligned}$$

where $[P^{(j)}_{\mathrm{SVD}}]_{(i)}$ is jth PCs of node n_i. Note that the above result holds for 3-D volumes as well. The first PC in that case will propagate radially outward from the center of the volume, as opposed to from the center of the area in case of 2-D networks.

C. Generation of Cartesian Coordinate Set Using VCs of a Subset of Nodes

Cartesian coordinates for 2-D TPM are obtained by multiplying the node's VC by V as in (3) and (4) [and as in (5) for 3-D TPMs]. V is based on P, the $N \times M$ matrix that consists of VCs of all the nodes. In sensor networks, it is crucial to reduce communication and computation overheads. This section presents a process to generate the transformation matrix V with only a small subset of rows of P, thus significantly reducing the computation overhead. Let Q be the submatrix of P corresponding to an appropriately selected set of R nodes (rows). Let the SVD of Q be

$$Q = U_Q \cdot S_Q \cdot V_Q^T. \quad (6)$$

Q is $R \times M$, where M is the number of anchors. U_Q, S_Q, and V_Q are $R \times R$, $R \times M$, and $M \times M$ matrices, respectively. If Q is selected appropriately, V_Q can serve as a substitute, or at a minimum a good approximation, for V for TPM generation. Note that V_Q has the same size as V in (1) and is also unitary. Following the same procedure as earlier, we use

$$P_{\mathrm{SVD}} = P \cdot V_Q. \quad (7)$$

The Cartesian coordinates for TPMs of 2-D and 3-D networks can be written as

$$\begin{aligned}[X_T, Y_T] &= [P^{(2)}_{\mathrm{SVD}}, P^{(3)}_{\mathrm{SVD}}] \\ [X_T, Y_T, Z_T] &= [P^{(2)}_{\mathrm{SVD}}, P^{(3)}_{\mathrm{SVD}}, P^{(4)}_{\mathrm{SVD}}] \quad (8)\end{aligned}$$

respectively.

While the there are many possible ways to select the subset of nodes, we use the following two simple options in this paper:

1) use the set of M anchor nodes ($Q = Q_A$);
2) use a set of R randomly selected nodes ($Q = Q_R$).

As $N \gg M, R$, significant savings in overhead can be achieved, and results presented later demonstrate that the impact on accuracy is negligible.

V is a basis of R^M. V_Q is also a basis for R^M even though it is based on a subset of coordinates. Therefore, we can write $V = V_Q \cdot \phi$, where ϕ is a rotation matrix. If the selected subset of coordinates is a good representation of the entire P, similar TPMs can be generated as demonstrated in Section V, with significantly lower computational, memory, and communication complexities.

TABLE III
COMPLEXITY AND MEMORY USAGE COMPARISON

Method	**Full SVD implementation with *P***	**EVD method of estimating *V* of *P***
# Computations N ≫M	$(4N^2M + 8M^2 + 9M^3)$ [13]	*Upper bounded by* $(4M^2N + 8M^3)$ [13]
Memory usage	$(M \times M) + (N \times N) + (N \times M)$	*Upper bounded by* $(M \times M) + (1 \times M)$

D. Computationally Efficient Implementation

Computational power and memory available at a sensor node is limited. Conventional SVD calculation of $P_{N\times M}, (N \gg M)$, which involves computing U, S, and V, has approximately $(4N^2M + 8NM^2 + 9M^3)$ operations [13]. Also, the memory requirement is approximately the sizes of V, U, and S that are $(M \times M + N \times N + N \times M)$. In this section, we present a technique for further enhancing the efficiency of the computation necessary for 2-D and 3-D TPM generation. Note that U is a by-product of SVD and is not necessary for topology map computation. The eigenvalue decomposition (EVD)-based approach [15] to evaluate matrix V not only allows us to implement the TPM generation in a distributed manner, but also completely avoids generating matrix U, thus reducing the computational complexity and memory requirement compared to those for SVD. From (1), (3), and (4) $P_{\text{SVD}}^{(j)}$, the jth column of P_{SVD} is given by

$$P_{\text{SVD}}^{(j)} = [h_{(n_iA_1)}, \ldots, h_{(n_iA_M)}] \cdot V^{(j)}, \qquad i = 1 : N. \tag{9}$$

$[h_{(n_iA_1)} \ldots, h_{(n_iA_M)}]$ is the coordinate vector of the node i. Also, $V^{(j)}$ is the jth basis vector/column of V. $j = 2, 3$ for 2-D networks, while $j = 2, 3, 4$ for 3-D networks. Thus, $V^{(2)}$ and $V^{(3)}$ are sufficient to evaluate 2-D Cartesian coordinates $[X_{T,i}, Y_{T,i}]$ of node i. Three-dimensional networks require $V^{(2)}, V^{(3)}$ and $V^{(4)}$. Define C as

$$\begin{aligned} C &= P^T \cdot P = V \cdot S^2 \cdot V^T \\ C \cdot V &= V \cdot S^2. \end{aligned} \tag{10}$$

C is an $M \times M$ symmetric matrix. This is an eigenvalue problem [15]. Therefore, let us solve

$$C \cdot v = \lambda \cdot v. \tag{11}$$

v is an eigenvector of C that is a column of V. Eigenvalues λ can be found by solving

$$|C - \lambda \cdot I| = 0. \tag{12}$$

The eigenvectors corresponding to second and third *largest* eigenvalues provide the second and third columns of V. Now, $[X_{T,i}, Y_{T,i}]$ ($[X_{T,i}, Y_{T,i}, Z_{T,i}]$ for 3-D case) can be evaluated locally without calculating the entire V matrix. Also, $U_{N\times N}$ is not evaluated at all, which reduces the memory consumption significantly. Therefore, the memory consumption is upper-bounded by $(M \times N) + (1 \times M)$. Number of computations required for this method of calculating V is upper-bounded by $(4M^2N + 8M^3)$ [13], which is the computations associated with calculation of entire V and S. Since $N \gg M$, this method is significantly less complex compared to the full SVD implementation (see Table III). For example, if the number of anchors in the network is set to $M \leq 0.01N$, which is reasonable based on our experience, the upper bound of computations required with this method is only 0.99% of the computations required for a full SVD-based calculation with (3) and (4), indicating a significant reduction in complexity.

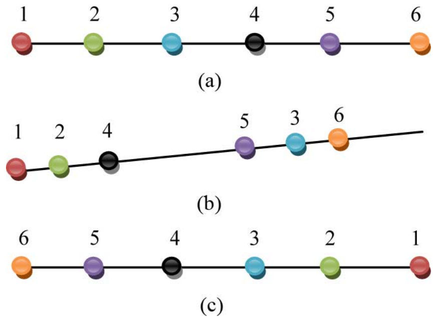


Fig. 7. (a) Network. (b) Topology map of (a) with a node flip. (c) Topology map of (a) with 180° rotation.

IV. METRIC FOR EVALUATING 2-D TOPOLOGY PRESERVATION

Evaluating the degree of topology preservation of the sensor node maps generated is essential for investigating the effectiveness of the proposed scheme. While visual inspection can provide preliminary evidence of its effectiveness, a formal metric is needed for quantifying the accuracy. A quantitative parameter to express the error provides a framework to compare and improve different mapping techniques. An effective metric should be able to capture and quantify the failures to preserve the topology of the real node map and the neighborhoods. Such a metric is not currently available. Here, we develop a metric that can be used for this purpose.

A method based on coloring of nodes is used in [28] to show whether a neighborhood has been altered in the topology map. In [28] and [32], error is quantified as the difference of the positions in the actual physical map and the topology map, and the residual variance, respectively. The focus of our paper is TPMs based on hop distances. The requirement is that the map from calculated $[X_T, Y_T]$ set is homeomorphic to the physical layout and preserves information about node connectivity, physical layout, and physical voids. Thus, the actual physical distance is not of significance, and the metrics in [28] and [32] are not appropriate.

Consider as an example a 1-D network with six nodes numbered 1–6 as in Fig. 7(a). Fig. 7(b) and (c) shows two derived maps that need to be evaluated. If all the nodes are in same order as in initial topology, then topology preservation error must be 0%. Node 3 in Fig. 7(b) has flipped two node positions. The error metric should identify the number of out-of-order nodes as well as the degree of the error/node flips [one node and two node positions, respectively, for Fig. 7(b)].

Consider a 1-D network with N nodes, and define an indicator function $I_{i,j}$, where

$$I_(i,j) = \begin{cases} 1, & i \text{ and } j \text{ are out of order} \\ & \quad \text{compared to original placement} \\ 0, & i \text{ and } j \text{ are in same order} \\ & \quad \text{as original placement or } i = j \end{cases}$$
$$i, j = 1 \text{ to } N.$$

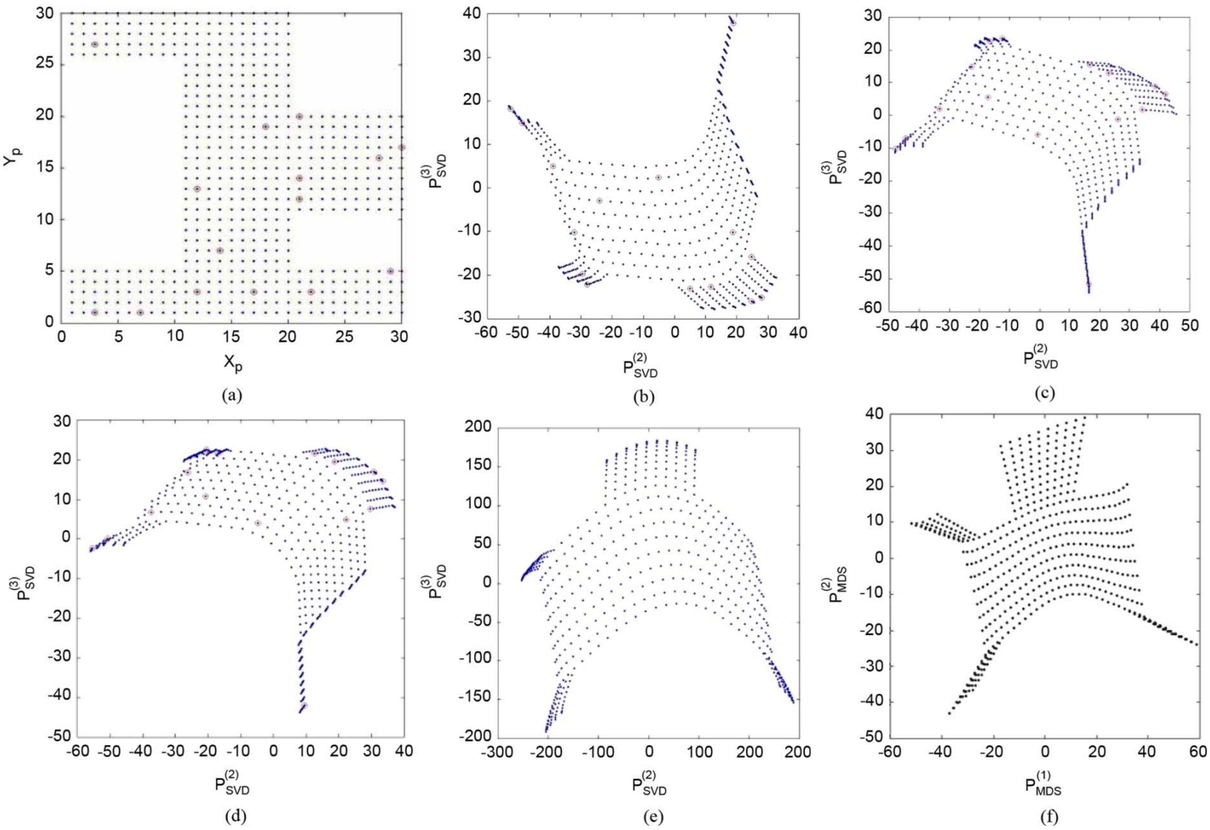


Fig. 8. (a) Odd-shaped network with 550 nodes and 10 random anchors. $[X_T, Y_T]$ is generated based on (b) Case 1: entire vc set; (c) Case 2: anchors' coordinate set; (d) Case 3: randomly selected nodes' coordinate set; (e) Case 4: coordinate set with all the nodes are anchors, and (f) MDS.

Then, the number of out-of-order pairs is $\sum_{\text{all } i,j}(I_{i,j})$. The total number of possible pairs in an N-node network is P_2^N. We define the following metric:

$$\text{Topology Preservation Error} = E_{\text{TP}} = \frac{\sum_{\text{all } i,j}(I_{i,j})}{(P_2^N)}. \tag{13}$$

For the network in Fig. 7(b), $N = 6$, and

$$\begin{aligned} E_{\text{TP}} &= (I_{3,4} + I_{3,5} + I_{4,3} + I_{5,3})/P_2^6 \times 100\% \\ &= (I_{3,4} + I_{3,5})/C_2^6 \times 100\% = 13.3\%. \end{aligned}$$

Nodes 1 and 2 are in the right position compared to the rest, while node 3 is shifted by two positions. Moreover, nodes 4 and 5 flipped their positions by 1. Therefore, total node flips are 4, and E_{TP} is 13.3%. A TPM is invariant to rotations. Thus, for Fig. 7(c), where nodes are just reversed, E_{TP} has to be zero. To handle such cases, the two lines being compared need to be adjusted for any rotations.

To extend the E_{TP} equation to 2-D topologies, we evaluate the 2-D topology by considering all contiguous line segments in two orthogonal directions (say $\vec{H}$ and $\vec{V}$) of the physical map.

Let there be α lines in $\vec{H}$ direction and β lines in $\vec{V}$ direction in the network, then topology preservation error in $\vec{H}$-direction is equal to

$$E_{\text{TP}|\vec{H}} = \frac{\sum_\alpha \sum_{\text{all } i,j}(I_{i,j})}{\sum_\alpha P_2^{N_{\text{h}}}} \tag{14}$$

where i, j are nodes in each horizontal line, and each line has N_{h} nodes. Similarly, error in vertical direction is evaluated as vertical neighborhood preservation error

$$E_{\text{TP}|\vec{V}} = \frac{\sum_\beta \sum_{\text{all } i,j}(I_{i,j})}{\sum_\beta P_2^{N_{\text{v}}}} \tag{15}$$

where i, j are nodes in each vertical line, and each has N_{v} nodes. The overall topology preservation error E_{TP} can be defined as

$$E_{\text{TP}} = \frac{\sum_\beta \sum_{\text{all } i,j}(I_{i,j}) + \sum_\alpha \sum_{\text{all } i,j}(I_{i,j})}{\sum_\alpha P_2^{N_{\text{h}}} + \sum_\beta P_2^{N_{\text{v}}}}. \tag{16}$$

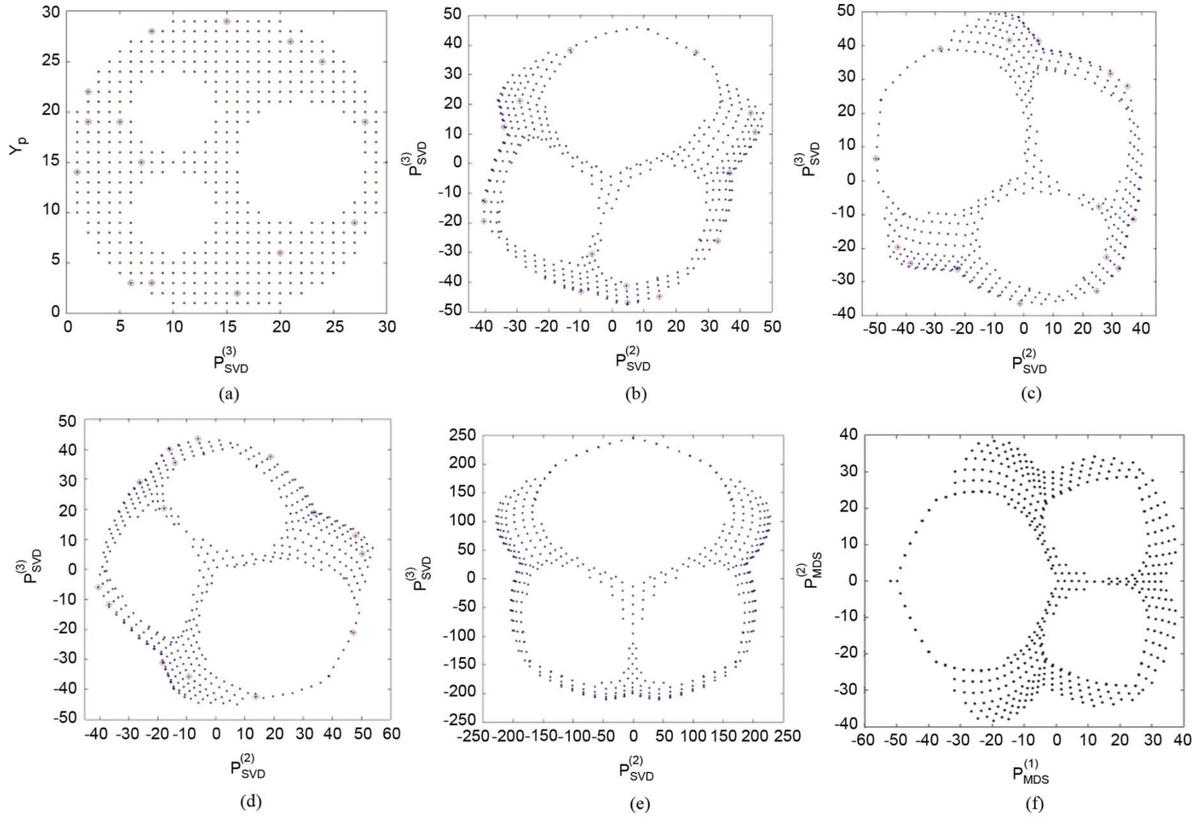
V. Results

The performance of the proposed TPM generation method is evaluated next using three 2-D examples and two 3-D examples representative of a variety of networks. MATLAB 2009b was used for the computations.

A. TPMs of 2-D Networks

Figures identified as (a) in Figs. 8–10 show the physical maps of the three 2-D networks considered: An odd-shaped network with 550 nodes [Fig. 8(a)], a 496-node circular-shaped network with three physical voids/holes [Fig. 9(a)], and a network of 343 nodes on walls of a building [Fig. 10(a)]. Communication range of a node in all three networks is unity. Detailed specifications of these networks are available in [4]. Topology maps are generated based on methods summarized in Table IV.

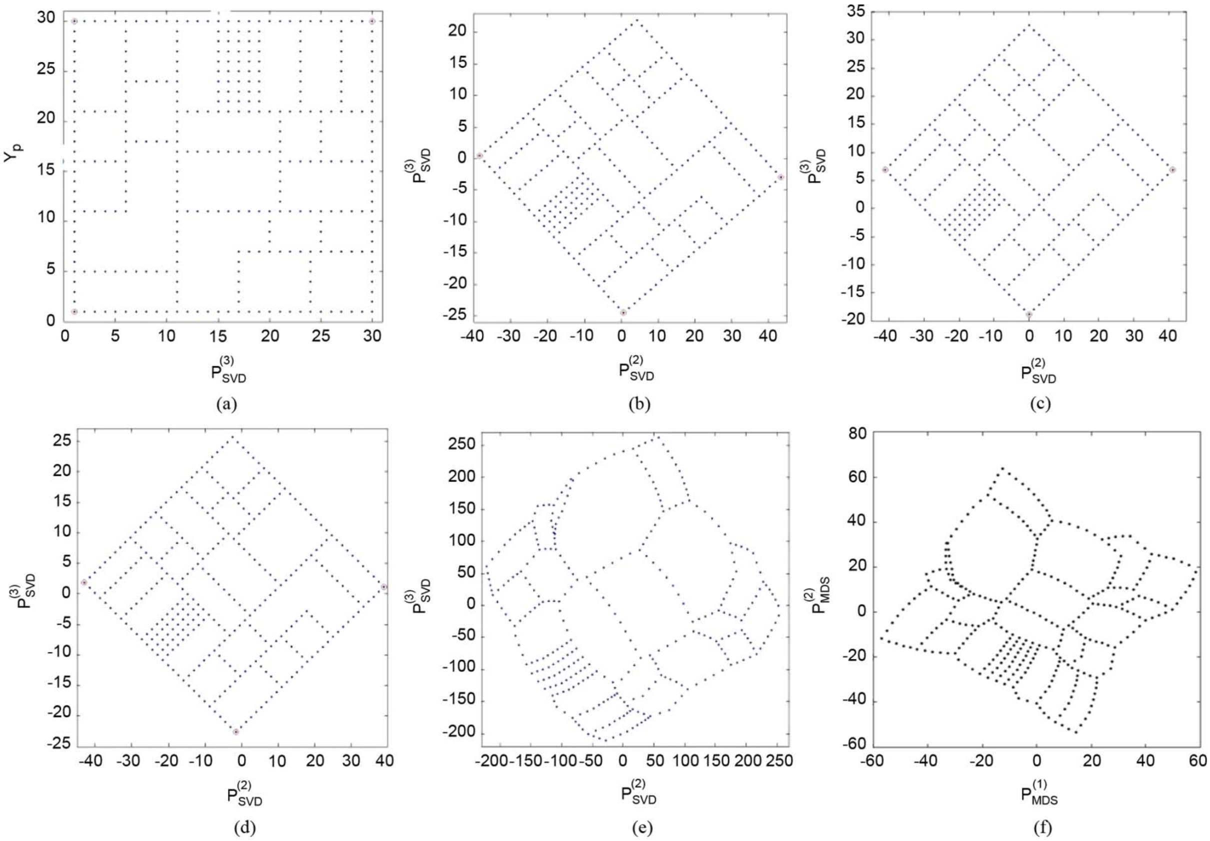
Unless otherwise indicated, the results shown correspond to 15 randomly placed anchors in each of the networks.



(a) (b) (c)

(d) (e) (f)

Fig. 9. (a) Circular network with three physical voids with 496 nodes and 10 random anchors. $[X_T, Y_T]$ is generated based on (b) Case 1: entire VC set; (c) Case 2: anchors' coordinate set; (d) Case 3: randomly selected nodes' coordinate set; (e) Case 4: coordinate set with all the nodes are anchors; and (f) MDS.



(a) (b) (c)

(d) (e) (f)

Fig. 10. (a) Network in a building with 343 nodes and three anchors. $[X_T, Y_T]$ is generated based on (b) Case 1: entire VC set; (c) Case 2: anchors' coordinate set; (d) Case 3: randomly selected nodes' coordinate set; (e) Case 4: coordinate set with all the nodes are anchors; and (f) MDS.

TABLE IV
DIFFERENT TOPOLOGY MAP GENERATION APPROACHES FOR WSNs OF N NODES AND M ANCHORS

Case	Description	Size of input data matrix
1	$[X_T, Y_T]$ from P	$N \times M$
2	$[X_T, Y_T]$ from Q_A	$M \times M$
3	$[X_T, Y_T]$ from Q_R	$R \times M$
4	$[X_T, Y_T]$ from $Q_A\|_{\|A\|=M=N}$	$N \times N$

Building network in Fig. 10(a) has just three anchors. Figs. 8(b), 9(b), and 10(b) show TPMs constructed based on (4) using the entire VC set of each network. Therefore, TPMs in Figs. 8(b), 9(b), and 10(b) use input data matrices of sizes 550×15, 496×15, and 343×3, respectively (Case 1, Table IV). Figs. 8(c), 9(c), and 10(c) are the topology maps created using only the anchors' coordinate set, that is, using (7) and (8) based on the input data matrices Q_A of size 15×15, 15×15, and 3×3, respectively (Case 2, Table IV). Topology maps in Figs. 8(d), 9(d), and 10(d) are created based on coordinates of 10 randomly selected nodes, i.e., the corresponding sizes of Q_R are 10×15, 10×15, and 10×3, respectively (Case 3, Table IV). For the purpose of comparison, Figs. 8(e), 9(e), and 10(e) consider all the nodes in the network to be anchors, corresponding to P of sizes 550×550, 496×496 , and 343×343, respectively, for the three networks (Case 4, Table IV). Case 3 is more efficient in terms of memory consumption and computational complexity. Finally, we compare our results to those of the MDS-MAP method proposed in [29] shown in Figs. 8(f), 9(f), and 10(f). For MDS, data from the complete distance matrix D, which is defined as the matrix of all the pairwise distances between points/nodes, are required. $D = [d_{ij}]_{N \times N}$, where N is the number of nodes in the network and d_{ij} is the distance from node i to node j. As proposed in [29], d_{ij} can be either geodesic distance or hop distance between i and j. For this comparison, we use hop distances to generate MDS-MAP, thus VCS requires all the nodes to be anchors. TPM for the circular network of Fig. 1(a) can be found in [9].

Figs. 8–10 clearly demonstrate the effectiveness of the proposed method in generating TPMs. Starting just with VCs, without explicit knowledge of geographical information, the generated topology maps have captured significant features such as the physical voids and boundaries of the original network. A key observation we can draw from Figs. 8–10 is that the constructed topology maps are nonlinearly scaled and rotated compared to the actual network map. Yet, the original and constructed maps are topologically isomorphic. In contrast to previous cases, the topology maps of Fig. 10(b)–(d) are simply rotated and linear scaled versions of the original. In this network, we used only three anchors that were manually selected. The physical voids present in Fig. 10(a) are well preserved. Even though the map in Fig. 10(e) was obtained using all the nodes as anchors, its shape is deformed compared to Fig. 10(b)–(d), but in terms of neighborhood preservation, Fig. 10(e) is better. For example, one of the L-shaped rooms in the building network [Fig. 10(a)] is distorted in the topology maps of Fig. 10(b)–(d). In Fig. 10(e), the L-shape is deformed, but neighborhood of that L-shaped room is preserved. Case 4 is presented here only for the purpose of comparison. If all nodes are anchors, a very expensive proposition for WSNs, need for maps does not arise for many applications such as routing. Obtaining MDS-MAPs shown in Figs. 8(f), 9(f), and 10(f) requires the hop distances from each node to every other node. A major disadvantage is that it is not feasible to implement MDS in a distributed manner due to the extremely high communication cost associated with generating the distance matrix consisting of distance between every pair of nodes. In fact, if such information is available at each node, it can be used to achieve 100% routability without the need to generate topology preserving maps.

TABLE V
E_{TP} FOR TOPOLOGY MAPS IN FIGS. 8–10

	$\mathbf{E_{TP}}$ **(%)**			
Fig	**Case 1**	**Case 2**	**Case 3**	**Case 4**
Fig. 8 (a)	1.6777	1.5894	1.5011	1.4570
Fig. 9 (a)	0.3605	1.0698	0.4884	0
Fig. 10 (a)	0.1315	0.1315	0.1315	0.0376

Moreover, from topology maps in Fig. 10, we can draw the valuable conclusion that a good anchor placement can significantly reduce the number of anchors required for topology map generation. It is topology-preserving to a very high degree as intended. It can be clearly seen that maps in Fig. 10(b)–(d) are very close to the original map, indicating that an appropriately placed small number of anchors can produce very accurate topology maps. This points to the possibility of obtaining even physically representative layout maps with appropriate selection of anchor nodes for a certain class of networks. Furthermore, our later research in [22] demonstrates that TPMs can be obtained even under large communication ranges.

E_{TP} [in (16)] for the different topology maps is presented in Table V. Note that the error in all the cases is less than 2%. The best performance in terms of E_{TP} was achieved when all the nodes were selected as anchors for the networks in Figs. 8–10. Case 4 (Table IV) acts as a lower bound for the E_{TP} for each network.

Even though SVD-based TPM generation started with a VC set where there is no directionality information, resultant topology map has directional information that can be used for routing in many ways. For example to avoid logical voids in VC routing, organized random routing and GR on TPM may be used [12]. Moreover, as discussed in Section II, there are other VCSs [21], [33], which are derivatives of hop-distance-based VCS used here. Use of the proposed TPM generation method with two such systems is addressed in Appendix-B.

B. TPMS of 3-D Networks

In this section, we present the 3-D TPMs generated using the proposed scheme. Two example networks deployed on 3-D are considered as shown in Figs. 11(a) and 12(a).

1) T-joint (3-D surface network): A pipeline structure joining two perpendicular cylinders in a T-joint. There is a hole in one of the cylinders [see Fig. 11(a)]. Each cylinder has a unit radius and a height of 7 units. It is covered with 1642 nodes, each with a communication range of 0.4. Fifty randomly selected nodes (i.e., 3% of the nodes) served as anchors.

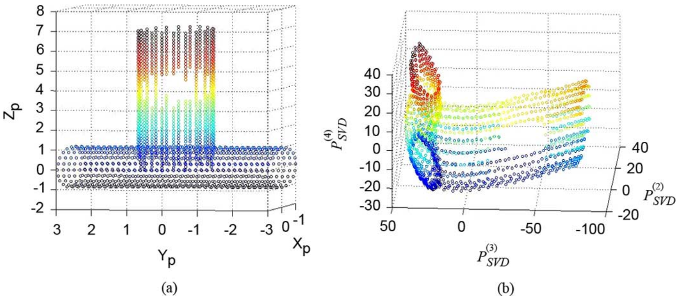


Fig. 11. 3-D surface network, consisting of two perpendicular cylinders (T-joint) (1642 nodes, 50 randomly selected anchors): (a) physical layout and (b) TPM.

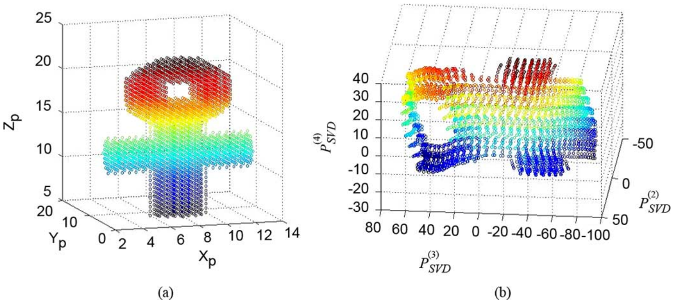


Fig. 12. 3-D volume network, consisting of a sphere standing on two crossed cylinders. Sphere has a hole in it (3827 nodes and 50 randomly selected anchors): (a) physical layout and (b) TPM.

2) 3-D volume network: It consists of a solid sphere of radius 4 with a cylindrical hole, mounted on two perpendicularly crossed cylinders with height 10 and radius 2 [see Fig. 12(a)]. The entire volume is filled with 3827 nodes, each with a communication range of 0.5. Fifty randomly selected nodes (i.e., less than 1.5% of the nodes) served as anchors.

TPMs of the corresponding physical topologies are shown in Figs. 11 and 12, respectively. The results clearly demonstrate the effectiveness of the TPM generation for sensor networks deployed on 3-D surfaces and in 3-D volumes. Moreover, it indicates that the maps can be obtained using a very small number of random nodes serving as anchors.

VI. Realizations, Applications, and Extensions

The major contribution of this paper is the technique described and evaluated above for the generation of TPM. Section VI-A, briefly addresses the realization details of the TPM algorithm in a static WSN. Routing is a crucial operation in WSNs. Section VI-B discusses how WSN routing can benefit from TPMs. Section VI-C discusses the impact of network dynamics on TPMs.

A. Off-Network and In-Network Realization of TPM

First, let us consider the case where the TPM computation is done at a central node. There are many scenarios where a centralized implementation is feasible or even preferable. In a sensor network where the nodes are randomly deployed (e.g., dropped from a plane), it may be necessary and useful for the command center to obtain a map of the sensor node deployment indicating geographic voids, boundaries, etc. In this case, each node may send information about its neighbors to a base or a central station. The adjacency matrix of the network is formed based on the nodes connectivity information, which can be gathered with the worst-case complexity of $O(N^2)$, where N is the number of nodes in the network. Then, the procedures explained in Section III can be used to generate an effective and accurate TPM since there is no computational or memory limitations at the base station. Moreover, if necessary, the map can be broadcast back to the individual nodes, together with the transformation matrix (V or V_Q), an operation of worst-case complexity of $O(N^2)$. Note that redistributing second and third columns of V or V_Q is sufficient for a node to calculate its topological coordinate. Generating coordinates at a central station avoids multiple flooding in the network [5], [6].

A distributed implementation of the above may be achieved as follows. The anchor-based VC generation is first carried out the traditional way, i.e., via flooding. Following that, the anchors broadcast their coordinates, which requires $O(MN)$ messages. Since the submatrix $(Q = Q_A)$ of all the anchors' coordinates is now available at each node i, every node can generate V_Q [using (7)] and compute its own $[x_{\gamma,i}, y_{T,i}]$ locally by simply multiplying its own coordinates by second and third columns of V_Q.

TABLE VI
Probability of Selecting the Correct Neighbor Based on TPM and Physical Map for the Networks in Fig. 10

Network	Topology Maps				Physical Maps
	M=20	M=15	M=10	M=5	
Circle with voids (Fig. 10 a)	0.64	0.61	0.62	0.52	0.54
Building network(Fig. 9 a)	0.84	0.80	0.78	0.72	0.83

TABLE VII
Performance Comparison of GLR, LCR, CSR and GPSR With 10 Anchors [12]

Routing scheme	Avg. routability%	
	Circle with voids (Fig. 10 a)	**Building network(Fig. 9 a)**
GLR	94.6	89.3
LCR	56.5	49.7
CSR	87.3	75.4
GPSR	93.8	97.4

B. TPM-based Routing

We already asserted that in many ways the TPM is a better candidate for GR than the original physical map, as the former is based on actual connectivity information rather than the node position. A set of coordinates is good for routing if it results in accurate forwarding decisions. This can be quantitatively evaluated using

$$P[\text{Selecting correct neighbor}] = \sum_{N_j \in N} \sum_{N_i \in N} \frac{\begin{array}{c}\text{Number of times a } N_j \text{ selected correct neighbor}\\ \text{to FWD the packet when destination is } N_i\end{array}}{\text{Total number of nodes } (N)}. \tag{17}$$

Table VI shows this probability using physical and topology-based Cartesian coordinates for two example networks. Topology maps generated with 10 randomly selected anchors have the capability of selecting the correct next neighbor as accurately as with physical coordinates for the networks in Figs. 9(a) and 10(a). Many other self-organization tasks can also be expected to perform well with TCs instead of geographical coordinates.

In static WSNs, the VC generation needs to be done less frequently or perhaps only once during initialization. Therefore, topological coordinates also need not be updated frequently. Thus, the cost incurred in calculating Cartesian coordinates may be more than compensated by efficiency gains in terms of performance during long-term operation. For example, as illustrated in [12], the GLR scheme that uses both VCS and TPM to overcome disadvantages in each other's domains outperforms the physical information-based routing scheme—GPSR [14]. Table VI summarizes the performance of GLR, the GC-based scheme GPSR, and two VCS-based routing schemes, namely Convex Subspace Routing (CSR) and Logical Coordinate Routing (LCR). Routability is evaluated over all possible source–destination address pairs. Additional details of GLR algorithm are available in [12].

C. *TPM for Dynamic Networks*

Network dynamics that cause changes in connectivity among nodes pose a challenge for VC-based approaches as VC values depend on the connectivity of the network. Examples of such conditions include node failures, the introduction of new nodes, and change in connectivity due to mobile nodes. TPMs presented here capture the physical layout information of the network, i.e., the topological coordinates correspond to the physical position of a node, albeit on a somewhat distorted layout. When a node (or even an anchor) fails, the already calculated TCs of a node still remain valid for the topology map. Thus, any algorithm relying on TCs can continue to function even though the underlying VCs may no longer be valid. This can be considered as an advantage of using the TCs instead of the VCs, as VCs have to be regenerated to accommodate the change in connectivity.

Introduction of new nodes or mobility of nodes that cause major changes in network topology can render the TPM inaccurate, thus requiring its recomputation. If the change in the connectivity pattern is completely localized, it may be possible to estimate the TCs of a new node based on some localized computations involving its immediate static neighbors.

VII. Conclusion and Future Work

We presented a novel and a fundamental technique for generating TPMs from VCs for 2-D and 3-D (both surface and volume) WSNs. The transformation matrix for converting the virtual (logical) coordinates to a set of topological Cartesian coordinates can be obtained using the VCs of a very small set of nodes. Results show that a remarkable 2-D topology preservation error $(E_{\text{TP}}) \leq 2\%$ is achievable with a small number of anchors.

The topology coordinate space provides an alternative space for sensor networking algorithms beyond the traditional physical and VC spaces. It preserves the main advantage of VC scheme in not requiring distance measurements and that of GC scheme in having cardinal direction and boundary/void information. TPMs may be used in lieu of physical maps for many applications and WSN protocols [12]. The TPM generation scheme presented above has been used for functions such as boundary node identification, event region, and void detection [10] with performance on par with GC-based schemes. In fact, the TCs in conjunction with VCs from which they are derived have been demonstrated to be better than geographical coordinates for routing with significantly enhanced routing performance [12]. While there are certain applications for which the exact sensor location is necessary, for others that do not need such information, TPM presents a robust, accurate, and scalable alternative to physical map generation or localization. Sensor network applications of TPMs are diverse and vast; examples include routing, localization, boundary node identification, and effective anchor placement.

We envision many applications of the proposed topology preserving map extraction methodology in other types of networks as well as in multidimensional graphs, e.g., for dimension reduction, visualization, and information extraction. Methods to compensate for the distortion of the maps compared to physical maps and techniques that use derived Cartesian coordinates and the topology map to improve self-organization and routing protocols are also under investigation.

Appendix

This appendix addresses the convexity of first principle component of an anchor-based VCS and the applicability of proposed TPM generation scheme for other existing VCSs

A. *Convexity of the First Principle Component*

Being the distance to the corresponding anchor from a node, by definition each VC radially increases around the corresponding anchor. Due to the fact that the first PC captures the salient dominant features of the data set, its magnitude variation over the network is always convex; due to the possibility of having positive or negative sign, the actual shape of first PC variation is either convex or concave.

We demonstrate the convexity of magnitude first on a simple 1-D network, and then extend it to a 2-D full grid. Let the VCS with respect to M anchors of a 1-D network, as illustrated in Fig. 13(a), be $P = [P^{(1)}P^{(2)}\dots P^{(M)}]$. By definition, each VC with respect to anchor $A_j : P^{(j)}$ is a convex function with respect to the node position n_i.

The first PC can be written as

$$P^{(1)}_{\text{SVD}} = [P^{(1)}P^{(1)}\dots P^{(M)}]\cdot V^{(1)}. \tag{A.1}$$

$PV^{(1)}$ is a linear combination of the set of convex functions $P^{(j)}$'s. Reference [24] proves that the direction of the first PC, i.e., $V^{(1)}$, goes through the centroid of the data points. Since $P^{(j)}$ lie in the first orthant of the multidimensional space all the time, its centroid is also in the first orthant. Hence, $V^{(1)}$ is a unit vector with either all positive coefficients or all negative coefficients. Without loss of generality, one can say (A.1) is the addition of M convex functions, and thus the first PC is also a convex function. For example, variation of first and second PCs is shown in Fig. 13(b). The 1-D maps of the network obtained using the first PC [Fig. 13(c)] shows a network that is folded in two as expected, that using the second PC [Fig. 13(d)] shows a map where topology and local neighborhoods are preserved.

A similar argument can be made for the 2-D full grid since VCs with respect to anchor A_i are a 2-D convex surface. Moreover, all the ordinates lie in the first orthant. Hence, for a 2-D

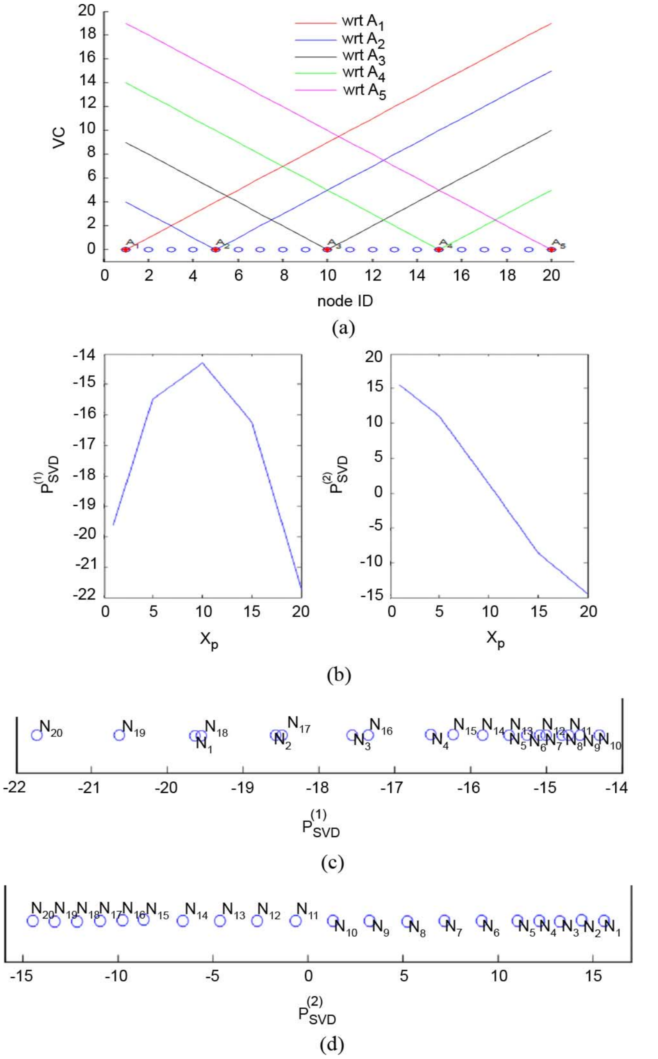


Fig. 13. (a) VCS corresponding to two anchors in a 1-D network. (b) First and second PCs for the VCS in (a). (c) TPM of the network in (a) based on first PC. (d) TPM of the network in (a) using second PC.

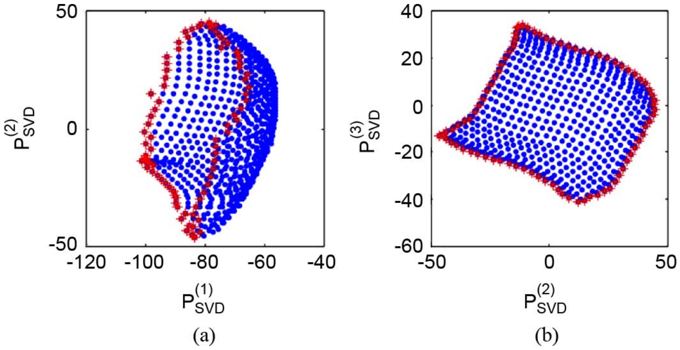


Fig. 14. Topology preserving map of a circular network [Fig. 1(a)] using (a) first and second PC and (b) second and third PC. The edge nodes of the network are highlighted.

grid, $V^{(1)}$ is a unit vector with either all positive or all negative coefficients resulting in a sum of convex functions, which is convex. Therefore, the first PC is convex for a 2-D full grid as well. Fig. 14(a) shows the map of the network of Fig. 1(a) when the first and second PCs are used as axis. The network gets folded due to the dominant convex shape of the first PC.

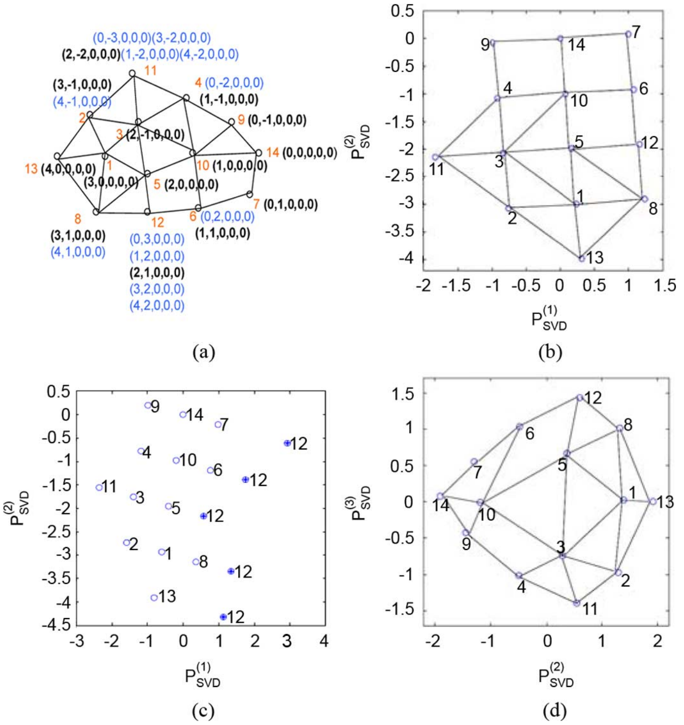


Fig. 15. (a) Example network with its ABVCap VCs [33]. (b) TPM only with ABVCap coordinates in bold black. (c) TPM with all possible ABVCap coordinates of node 12. (d) TPM from Aligned VCs.

In contrast, the map in Fig. 14(b) using second and third PCs as axis is a TPM that preserves local neighborhoods.

B. TPM From Other Virtual Coordinate Systems

The use of proposed TPM generation technique with other VCSs derived from hop distances is demonstrated next using two such schemes, ABVCap [33] and Aligned VCS [21].

ABVCap [33] characterizes each node by a 5-tuple consisting of (longitude, latitude, ripple, up, down). These entries are specified relative to virtual lines identified in the network as follows. Initially, three anchors (X, Y, Z) are selected based on VCap anchor selection. A fourth anchor, Z', is selected such that it is farthest away from Z and equidistant from X and Y. Generation of 5-tuple (longitude, latitude, ripple, up, down) involves several additional network floodings. Fig. 15(a) shows an example network used in [33] with ABVCap-VCS.

One notable property of ABVCap-VCS is that some nodes have more than one VC tuple assigned to them. Either one of the tuples has to be selected for each node, which introduces unnecessary complexity to identify the proper tuple for topology map generation, or multiple positions in TPM will be assigned to the same node based on different coordinate tuples. The TPM shown in Fig. 15(b) is generated using our scheme based on VC tuples identified in bold in Fig. 15(a). Fig. 15(c) indicates multiple positions created for node 12 due to its multiple coordinates in ABVCap. As ABVCap-based VCS does not have concentrically increasing property, the first PC and second PC provide the TPM. While this demonstrates the applicability of TPM for ABVCap, we note that essentially the same information can be obtained simply by applying the method to a simple VCS without having to undergo overhead required to generate ABVCap.

Aligned VCS [21] proposes a modification for VCS to alleviate the local minima problem simply by replacing the VCs of each node with the average of node's and its neighbors' VCs. Thus, we have used the VCS w.r.t. (X, Y, Z, Z') of Fig. 15(a) to evaluate aligned VCS as in [21]. The TPM from the corresponding aligned VCS is shown in Fig. 15(d). Since aligned VCs are also radial in nature, the radial component can be removed using the first PC, and the second and third PCs provide the Cartesian coordinates. These results indicate the applicability of the proposed TPM generation technique to other VCS as well.

REFERENCES

[1] I. F. Akylidi and J. M. Jornet, "The Internet of nano-things," *IEEE Wireless Commun..*, vol. 17, no. 6, pp. 58–63, Dec. 2010.

[2] J. Bachrach and C. Taylor, "Localization in sensor networks," in *Handbook of Sensor Networks*. Hoboken, NJ, USA: Wiley, 2005, ch. 9.

[3] Y. Bengio, J.-F. Paiement, and P. Vincent, "Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 177–184.

[4] Colorado State University, Fort Collins, CO, USA, "CSU sensor-net benchmarks," 2012 [Online]. Available: http://www.cnrl.colostate.edu/Projects/VCS/

[5] A. Caruso, S. Chessa, S. De, and A. Urpi, "GPS free coordinate assignment and routing in wireless sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2005, vol. 1, pp. 150–160.

[6] Q. Cao and T. Abdelzaher, "Scalable logical coordinates framework for routing in wireless sensor networks," *Trans. Sensor Netw.*, vol. 2, pp. 557–593, Nov. 2006.

[7] D. C. Dhanapala, "On performance of random routing and virtual coordinate based routing in WSNS," M.S. thesis, Colorado State Univ., Fort Collins, CO, USA, 2009.

[8] D. C. Dhanapala and A. P. Jayasumana, "CSR: Convex subspace routing protocol for wireless sensor networks," in *Proc. 33rd IEEE LCN*, Oct. 2009, pp. 101–108.

[9] D. C. Dhanapala and A. P. Jayasumana, "Topology preserving maps from virtual coordinates for wireless sensor networks," in *Proc. 35th IEEE LCN*, Oct. 2010, pp. 136–143.

[10] D. C. Dhanapala, A. P. Jayasumana, and S. Mehta, "On boundary detection of 2-D and 3-D wireless sensor networks," in *Proc. IEEE GLOBECOM*, 2011, pp. 1–5.

[11] D. C. Dhanapala and A. P. Jayasumana, "Clueless nodes to network-cognizant smart nodes: Achieving network awareness in wireless sensor networks," in *Proc. IEEE CCNC*, 2012, pp. 174–179.

[12] D. C. Dhanapala and A. P. Jayasumana, "Geo-logical routing in wireless sensor networks," in *Proc. 8th IEEE SECON*, Jun. 2011, pp. 305–313.

[13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 2003, Appendix 4.

[14] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. 6th ACM MobiCom*, 2000, pp. 243–254.

[15] M. Kirby, *Geometric Data Analysis—An Empirical Approach to Dimensionality Reduction and the Study of Patterns*. New York, NY, USA: Wiley, 2001.

[16] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: Of theory and practice," in *Proc. 22nd ACM PODC*, Jul. 2003, pp. 63–72.

[17] S. Lederer, Y. Wang, and J. Gao, "Connectivity-based localization of large scale sensor networks with complex shape," *Proc. 27th IEEE INFOCOM*, pp. 789–797, 2008.

[18] B. Leong, S. Mitra, and B. Liskov, "Path vector face routing: geographic routing with local face information," in *Proc. 13th IEEE ICNP*, Nov. 2005.

[19] K. Lferd, J. R. Martínez-de Dios, A. de San Bernabé, and A. Ollero, "Experimental comparison of RSSI and TOF measurements for localization," presented at the EWSN, 2011.

[20] O. Liang, Y. A. Sekercioglu, and N. Mani, "A low-cost flooding algorithm for wireless sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2007, pp. 3495–3500.

[21] K. Liu and N. Abu-Ghazaleh, "Aligned virtual coordinates for greedy routing in WSNs," in *Proc. IEEE Int. Conf. Mobile Ad hoc Sensor Syst.*, Oct. 2006, pp. 377–386.

[22] S. Mehta, "Evaluation of topology generation in sensor networks," Colorado State Univ., Fort Collins, CO, USA, MS Project Rep., 2011.

[23] N. Mitton, T. Razafindralambo, D. Simplot-Ryl, and I. Stojmenovic, "Hector is an energy effiCient tree-based optimized routing protocol for wireless networks," in *Proc. MSN*, Dec. 2008, pp. 31–38.

[24] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philos. Mag.*, vol. 2, pp. 559–572, 1901.

[25] H. A. B. F. Oliveira, E. F. Nakamura, A. A. F. Loureiro, and A. Boukerche, "Error analysis of localization systems for sensor networks," in *Proc. 13th ACM Int. Workshop Geo. Inf. Syst.*, 2005, pp. 71–78.

[26] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proc. 4th IPSN*, Apr. 15, 2005, pp. 364–369.

[27] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proc. 9th Int. Conf. Mobile Comput. Netw.*, 2003, pp. 96–108.

[28] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 23239–2326, Dec. 2000.

[29] Y. Shang, W. R. 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.

[30] J. A. Sanchez and P. M. Ruiz, "Exploiting local knowledge to enhance energy-efficient geographic routing," *Lect. Notes Comput. Sci.*, vol. 4325, pp. 567–578, Dec. 2006.

[31] M. Steyvers, "Multidimensional scaling," in *Encyclopedia of Cognitive Science*. New York, NY, USA: Wiley, 2002.

[32] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, Dec. 2000.

[33] M. J. Tsai, H. Y. Yang, and W. Q. Huang, "Axis based virtual coordinate assignment protocol and delivery guaranteed routing protocol in wireless sensor networks," in *Proc. IEEE INFOCOM*, May 2007, pp. 2234–2242.

[34] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: A recursive data dissemination protocol for WSNs" UCLA CS Dept., Los Angeles, CA, USA, Tech. Rep. UCLA/CSD-TR-01–0023, May 2001.

[35] H. Zhou, H. Wu, S. Xia, M. Jin, and N. Ding, "A distributed triangulation algorithm for wireless sensor networks on 2D and 3D surface," in *Proc. IEEE INFOCOM*, 2011, pp. 1053–1061.

Dulanjalie C. Dhanapala received the Bachelor's degree, with First Class Honors, in electrical and electronics engineering from the University of Peradeniya, Peradeniya, Sri Lanka, in 2006, and the M.S. degree in electrical and computer engineering from Colorado State University, Fort Collins, CO, USA, in 2009, and is currently pursuing the Ph.D. degree in electrical and computer engineering at Colorado State University.

Her research interests include wireless sensor networks, communication networks, communication theory, and digital communication.

Anura P. Jayasumana (S'83–M'85–SM'90) received the B.Sc. degree in electronic and telecommunications engineering, with First Class Honors, from the University of Moratuwa, Moratuwa, Sri Lanka, in 1978, and the M.S. and Ph.D. degrees in electrical engineering from Michigan State University, East Lansing, MI, USA, in 1982 and 1984, respectively.

He is a Professor of electrical and computer engineering and computer science with Colorado State University, Fort Collins. He also serves as a consultant to industry; his past clients range from startups to Fortune 100 companies. His current research interests include sensor networking, network protocols, and performance modeling.

Prof. Jayasumana is a member of Phi Kappa Phi and the Association for Computing Machinery (ACM).