

Topology Preserving Map to Physical Map - A Thin-Plate Spline Based Transform

Ali F. Buoud

Department of Electrical and Computer Engineering,
Colorado State University, Fort Collins, CO 80523, USA
aboutd@rams.colostate.edu

Anura P. Jayasumana

Department of Electrical and Computer Engineering,
Colorado State University, Fort Collins, CO 80523, USA
Anura.Jayasumana@Colostate.edu

Abstract— A Topology Preserving Map (TPM) is an easily obtainable localization free connectivity based map that preserves physical layout features of 2D/3D sensor networks. This paper considers how to obtain physical maps and physical coordinates from a TPM when the physical locations of a subset of nodes are known. First, we present a General Procrustes Alignment (GPA) based solution, which is the optimal linear transformation solution to the problem. Second approach is based on thin-plate spline (TPS), which transforms the set of topology coordinates to physical coordinates using radial basis functions. Five representative 2D network topologies are used to evaluate and compare the TPS approach with GPA approach, and also with the existing distance vector-hop (DV-Hop) technique. Results are presented for the cases where the reference nodes are selected randomly from the entire network, or randomly from the inner and outer boundaries of the network. The results show that with less than 10% of nodes as reference nodes, a map with an average error less than 0.6 of radio range can be achieved with the TPS approach, which significantly outperforms both the GPA and the DV-Hop. TPS approach generalizes directly to 3-D networks as well.

Index Terms—Localization, Thin-plate Spline, Procrustes Analysis, DV-Hop, Virtual Coordinates, Topology Preserving Maps.

I. INTRODUCTION

Many Wireless Sensor Network (WSN) algorithms rely on node localization and the resulting physical coordinates. Certain applications such as those related to environmental monitoring often require the actual position of the sensors. Different localization techniques can be compared based on different trade-offs such as accuracy, scalability, complexity, redundancy, recovery and cost efficiency. Many of the solutions for WSN localization are range-based approaches, where the sensor nodes use measurements such as RSSI or time delay to estimate the distance. In these cases, nodes have to be equipped with specific hardware to determine these values. Note that the modern wireless sensor nodes range from extremely simple and low cost devices such as smart RFIDs, in which RSSI or time delay measurements are not feasible, to more sophisticated ‘mote’ type devices which come equipped with circuitry to provide RSSI. Even with mote type devices, distance estimation becomes impossible when they are deployed in harsh environments. In general, the localization of

sensor position involves significant uncertainties and complexity [1]. Other range-based localization approaches rely on onboard hardware such as GPS or ultrasound transceivers. These add-ons increase the overhead and cost [2], yet not feasible in many environments.

An attractive alternative solution when range-based localization is not feasible is range-free localization [2], but the main trade-offs include the processing overhead and accuracy. Range-free approaches utilize the connectivity information to infer approximations to physical positions or to represent sensor positions in a virtual domain referred to as a virtual coordinate system. An example of latter is the anchor-based virtual coordinates, in which a small subset of M sensors is selected as anchors (landmarks), and each node in the network is characterized by the M -tuple, or the Virtual Coordinates (VCs) consisting of the minimum hop distance to each of the anchors. $M \ll N$, where N is the number of network nodes. VCs represent the network topology in an M -dimensional space. Although, the virtual coordinates may be used with some success for functions such as routing, they do not replace the physical coordinates. Extracting the exact physical coordinates cannot be done directly from virtual coordinates techniques, since the directionality information is lost in the virtual domain.

Several approaches attempt to obtain geographical coordinates from such a representation. Distance Vector-Hop (DV-Hop) approach [3] attempts to transform the virtual representation to the physical domain by using some reference nodes. The approach assumes that the relation between the hop number and the Euclidean distance is linear in the case of short distance. Accordingly, it estimates the hop size for each sensor in the selected set. Then, every sensor calculates its distance from each sensor in the set by using the hop size of the closet sensor in the set. Finally, a triangulation algorithm is applied to locate each sensor. The advantage of DV-Hop is that it can be implemented in distributed fashion with a low computational complexity. MDS-MAP is an approach based on multidimensional scaling algorithm [4]. It considers the shortest hop distance between sensor pairs as a distance matrix. Then, it implements the metric-MDS to find the closet estimation sensor’s position which maintains the same pairwise distance in the distance matrix. MDS-MAP

implementation requires significant computational complexity and causes high broadcast traffic in the network.

Topology Preserving Map (TPM) [5], [6], [7] is a connectivity based network map with many advantages. Generating TPMs is less complex compared to MDS-MAP. Unlike DV-Hop and MDS-MAP, TPM does not require any physical coordinates or physical distances. Yet, TPM preserves the geographical features such as voids and boundaries and also seamlessly extend to 3D surfaces and volumes [7]. The topology coordinates (TCs), on which TPMs are based, outperform other forms of coordinates including the physical coordinates in many network related functions such as routing [8]. TCs are easy to generate and they can be implemented in a distributed manner. TPMs are based purely on hop distances, and thus they are nonlinearly distorted versions of physical maps.

The main contribution of this paper is a Thin-Plate Spline based algorithm, TPM-TPS, that derives the physical map of a network, i.e., physical locations of nodes, from its TPM when the physical coordinates of some of the nodes are known. As the results for several representative 2D and 3D networks presented demonstrate, it is able to compensate for the non-linear distortions in TPMs and recover accurate physical maps. Prior to TPM-TPS, we present a General Procrustes Alignment (GPA) based approach that we call TPM-GPA, to solve the same problem. GPA is the best linear transformation to fit the topology coordinates of the set of reference nodes to their physical coordinates, and as such it provides a baseline comparison for the effectiveness of TPM-TPS. We also compare our results with DV-Hop approach [3]. We emphasize that the proposed localization approach TPM-TPS combining TPM and TPS has significant advantages over DV-Hop. DV-Hop approach requires the anchor nodes to know their physical locations, whereas with TPM-TPS, the anchors can be completely independent of the reference nodes whose locations are known. Identification of the optimum set of anchors is a difficult problem on its own [6], but ensuring such anchors are in known locations complicates the task even more and will result in large anchor sets thus increasing the complexity of algorithms. Furthermore, TPM-TPS, unlike DV-Hop and MDS-MAP can start with no reference nodes at all, and can gradually improve as location information of more and more nodes become available. There are many applications for TPM-TPS where DV-Hop is not applicable. A network may operate purely using topology maps, which is a distorted version of the physical map, and with each addition of new location aware node, the map can continue to improve. Results demonstrate that the location of the sensor with acceptable error margin can be obtained from the local connectivity information and few reference nodes in distributed manner. The thin-plate spline algorithm (TPS) converts TPMs into the physical map in three simple steps. In the first step, a small subset of nodes broadcast their geographical/topology coordinates in the network. Secondly, each sensor calculates the TPS weights. Finally, sensor locates its position by using the TPS weight and its own topology coordinates.

This paper is organized as follows: Section II presents the background for proposed work. Section III covers the general Procrustes alignment (GPA) and Thin-Plate Spline (TPS) techniques for converting topology preserving maps to physical maps. Section IV evaluates the proposed schemes. Section V is the conclusion.

II. BACKGROUND

First we discuss the difference between physical and virtual coordinates. Next we review DV-Hop algorithm against which our results are compared. The Extreme Node Search (ENS) anchor selection is described and finally we review the TPM generation method.

A. Hop distance vs. physical distance

To illustrate the relation between the physical and VC (connectivity) domains, consider three sensors with radio ranges r_1 , r_2 and r_3 respectively as shown in Fig. 1. For sensor 1 and sensor 2 to be able to communicate with each other, their separation distance has to be less than or equal to $\min(r_1, r_2)$. Therefore any physical distance in range from 0 to $\min(r_1, r_2)$ will be mapped into one hop in the VCs domain. Thus different configurations of physical maps can be mapped into the same configuration in virtual coordinates, and thus same TPM. That is, sensors can be freely moved inside a specific region while keeping the same virtual coordinates. This region is referred as a freedom movement area (AR_i). Virtual to physical coordinate transformation is a challenging task due to this many-to-one relationship [5].

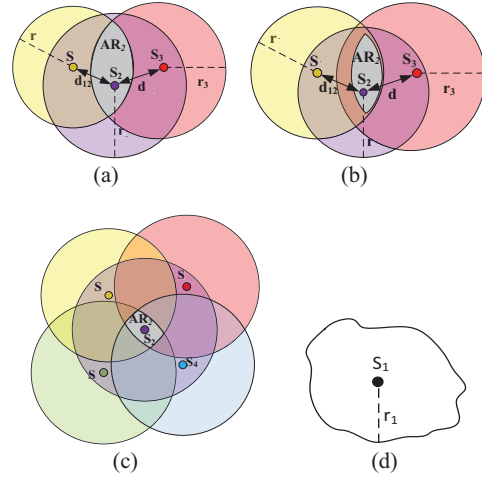


Fig. 1. (a) Three neighboring sensors with different radio ranges; (b) Effect of changing the radio range on freedom of movement area; (c) Effect of number of neighbors on freedom of movement area; and (d) An example of non-uniform communications range of a sensor.

Both radio range and number of adjacent neighbors can increase or decrease AR_i size. As illustrated in Fig. 1 (a), the physical position of the sensors for a given set of VCs is not unique. Sensor 2 can move forward, backward, up, and down inside AR_2 while staying within one hop of sensor 1. AR_2 is formed by the neighbor's coverage overlapping region (gray area) as long as radio range of sensor 2 is greater than or equal

to $\max(r_1, r_3)$. If this is not the case, then AR_2 will be sub-region of neighbor's coverage overlapping region where sensor 2 can move inside the intersection area under two constraints: its distance to sensor 3 is not more than $\min(r_1, r_3)$ and its distance to sensor 1 is not more than $\min(r_1, r_2)$, as shown in Fig. 1 (b). Sensor 2 can also move inside smaller area AR_2 than that depicted in Fig. 1 (a) without changing value of one-hop to sensor 1 and sensor 3 in the virtual domain. Furthermore, the freedom of movement region AR_i can be more restricted if the sensor has another neighbor, as shown in Fig. 1 (c).

In a real implementation, the wireless sensors nodes may be arbitrarily deployed, in which case the wireless network topology will emerge from the radio coverage overlap between adjacent sensors. The previous discussion assumed that the radio coverage is a uniform circle, but in reality it need not be, as in Fig. 1 (d), due to antenna radiation pattern, obstacles, medium absorption, etc.

B. DV-Hop algorithm

The DV-Hop algorithm [3] is based on finding a scale factor between the hop distance and Euclidean distance. In the first step, a virtual coordinate system is established by using a subset of sensors ("reference points") which are equipped with localization hardware as anchors. In the second step, each reference point estimates an average distance corresponding to the hop size. The average size hop of a reference point A is the sum of the pairwise Euclidean distance between the reference point A and other reference points divided by the sum of virtual coordinates of the reference point A . Next, each reference point floods its estimated average distance for a hop to the entire network. Each sensor can then determine its physical location in two steps: a) calculating its distances to all the reference points by multiplying its virtual coordinates by the average size hop of closest neighbor reference point, and b) performing a triangulation algorithm to calculate its own geographical coordinates.

C. Anchor selection

Anchor selection in VC systems is a critical task since improper anchors can result in identical coordinates and low routability. As the complexity of coordinate generation and subsequent computations increase with the number of anchors, it is necessary to keep the number of anchors as small as possible. Extreme node search (ENS) is very effective anchor placement schemes to produce general virtual coordinates with a small number of anchors [6], thus we use ENS for anchor selection in this paper. ENS uses a random pair of nodes (A_1, A_2) as initial anchors. Each of these anchors initiates a message containing a counter, and broadcasts it to its neighbors. The neighbors in turn increase the counter and repeat the process thus flooding the network. At the end of this process, each node contains $h_{n_i A_j}, h_{n_i A_k}$ corresponding to its distance to the two anchors. We denote the hop distance between node i and node j as $h_{n_i n_j}$, thus $h_{n_i A_j}$ is the distance between node i and anchor j . Next, ENS calculates a

Directional Virtual Coordinate (DVC) for each node n_i , [6] as follows:

$$\vec{f}(h_{n_i A_j}, h_{n_i A_k}) = \frac{(h_{n_i A_j}^2 - h_{n_i A_k}^2)}{2h_{A_j A_k}} \vec{u}_{A_j A_k} \quad (1)$$

$\vec{u}_{A_1 A_2}$ is the unit vector in direction (A_1, A_2) . This pair of anchors thus produces a single scalar value $|\vec{f}(n_i)|$ for each node, corresponding to the amplitude of the DVC. The nodes with maximum and minimum values of DVC in a local h-hop neighborhood are extreme nodes and they are selected as anchors. h-neighbor depends on the size of the wireless sensor network [6]. These extreme nodes together with the two initial anchors are the ENS anchors. Additional random anchor may be added to ENS if necessary to overcome the identical virtual coordinates.

D. Topology Preserving Map (TPM) generation

Given the set of M ENS anchors $(A_1, \dots, A_M)$, each node n_i ($i = 1, \dots, N$) is characterized by a virtual coordinate vector $(h_{n_i A_1}, \dots, h_{n_i A_M})$. These traditional VC's are built using the shortest hops count to the anchors, and as such they do not provide any physical direction or shape information. Topology Preserving Maps (TPM) [6],[7] is a technique to transform these VCs to network maps that preserve physical direction and shape information.

Let $\mathbf{P}$ denote the $N \times M$ matrix whose ij^{th} element correspond to the distance from n_i to A_j , $h_{n_i A_j}$. The i^{th} row thus corresponds to the VC vector of node n_i . Note that the total number anchors M is significantly less than the number of nodes N , i.e., $M \ll N$. The singular value decomposition (SVD) of $\mathbf{P}$ is as shown in Eq. (2). Finally, projection matrix of $\mathbf{P}$ on $\mathbf{V}$ is used to produce $\mathbf{P}_{svd}$ as in Eq. (3).

$$\mathbf{P} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (2)$$

$$\mathbf{P}_{svd} = \mathbf{P}\mathbf{V} = \mathbf{U}\mathbf{\Sigma} \quad (3)$$

The TPM of a 2-D network can be obtained by using 2nd and 3rd columns of $\mathbf{P}_{svd}$ as (X, Y) Cartesian coordinates. For 3-D networks, the TPM is provided by the 2nd, 3rd, and 4th columns of $\mathbf{P}_{svd}$ Cartesian coordinates [5],[7].

III. MAPPING OF TOPOLOGY PRESERVING MAP

According to [5], a TPM is a rotated, translated, and non-linearly scaled, i.e., distorted version of the physical map of the network. The question of interest is whether a map closer to the physical map of the network can be obtained from the TPM when physical locations of most of the sensor nodes are unknown or unmeasurable. An example would be a large-scale network in which a few nodes are equipped with GPS, but others are much simpler devices that have no means of measuring physical distances. Another example would be a mobile node going through a sensor field calibrating the physical location of some of the nodes. Next, we present two

approaches for transforming TPMs to physical maps, i.e., topology coordinates to physical coordinates.

A. General Procrustes Alignment (GPA) approach

General Procrustes Alignment is used to match or align two sets of data points by minimizing the differences between locations of corresponding points of the two data sets. Source data set is the one that needs to be transformed, and the target set is the desired set of locations. Procrustes alignment calculates the scaling factor, rotation angle, and shift (Euclidean transformations) between the two sets [9]. In case the transformation between the two sets is not linear, Procrustes alignment, given by Eq. (4) will find the linear transformation that has the minimum error. Procrustes alignment however requires the dimension and the number of input points and targets points to be equal. Let $\mathbf{X}$ and $\mathbf{Y}$ denote the source points set and the target points set, respectively. Then,

$$\mathbf{Y} \approx \mathbf{s}\mathbf{X}\mathbf{T} + \mathbf{1}\mathbf{t}^T \quad (4)$$

where, $\mathbf{1}$ is $[1,1,1,\dots,1]^T$. The parameters $\mathbf{s}$, $\mathbf{T}$ and $\mathbf{t}$ are the scaling factor, rotation matrix, and the shift point, respectively. The function in Eq. (4) is linear and it is the best linear transformation approximation from the source points to the target points.

While Procrustes alignment requires the source set and target set to be equal, in our case, the number of reference points is much smaller than the total set of nodes. Thus mapping can be done by following two phase approach:

Phase 1: Let L be the set of reference points of which physical coordinates are known. Let $\mathbf{Y}$ represents the $L \times 2$ matrix of these physical coordinates, and $\mathbf{X}$ be the $L \times 2$ matrix of the corresponding topology coordinates (obtained using Eq. (3)). Given those $\mathbf{X}$ and $\mathbf{Y}$ for the selected reference points L , all parameters $\mathbf{s}$, $\mathbf{T}$ and $\mathbf{t}$ are estimated.

It is important to note that more than two non-co-linear input points are needed. To find the parameters the following steps are applied:

- Find the centers of gravity for $\mathbf{X}$ and $\mathbf{Y}$ by subtracting the mean from each set so we will have $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$.
- Calculate matrix $\mathbf{A} = \bar{\mathbf{Y}}^T \bar{\mathbf{X}}$.
- Obtain SVD for $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$.
- Compute the rotation matrix $\mathbf{T} = \mathbf{V}\mathbf{U}^T$.
- Compute the scaling factor $\mathbf{s} = \frac{\text{trace}(\bar{\mathbf{Y}}^T \bar{\mathbf{X}} \mathbf{T})}{\text{trace}(\bar{\mathbf{X}} \bar{\mathbf{X}}^T)}$.
- Compute the shift vector $\mathbf{t} = \frac{1}{n}(\mathbf{Y} - \mathbf{s}\mathbf{X}\mathbf{T})^T \mathbf{1}$.

Phase 2: Using $\mathbf{s}$, $\mathbf{T}$ and $\mathbf{t}$ estimated in stage 1, now we use Eq. (4) to transform the TPM coordinates of the remaining nodes into new coordinates which we refer to as mapped physical coordinates (MPC). In this step, $\mathbf{X}$ corresponds to the TC matrix and the resulting $\mathbf{Y}$ is the corresponding MPC set. The method for solving Eq. (4) can be found in [9], where it can be turned into an optimization problem over $\mathbf{s}$, $\mathbf{T}$ and $\mathbf{t}$.

B. Thin-Plate Spline (TPS) interpolation approach

Thin-plate spline (TPS) has been widely used in image alignment in graphics and image measuring match in the medical applications. Many of its features make TPS an attractive algorithm. For example, it can be used in mapping or interpolation of an M -dimensional data set into N -dimensions. TPS, formulated in Eq. (5), consists of two components which are affine and non-affine. These two capture the localized nonlinear and global linear components in the mapped function domain [10].

$$\mathbf{y} = \mathbf{A}^T \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} + \sum_{i=1}^n w_i \phi(\|\mathbf{x} - \mathbf{c}_i\|) \quad (5)$$

constrained to

$$\sum_{i=1}^L w_i = 0, \quad \sum_{i=1}^L w_i c_{j,i} = 0 \text{ for } j = 1, \dots, d$$

Where, $\mathbf{x} = [x_1, x_2, \dots, x_d]^T$ is a real-valued vector of dimension d , $\mathbf{c}_i = [c_{i,1}, c_{i,2}, \dots, c_{i,d}]^T$ is the coordinate vector of the i^{th} reference point, $\mathbf{w} = [w_1, w_2, w_3, w_4, \dots, w_L]^T$ are the L weights of the basis functions, L is number of reference points and $\mathbf{A} = [a_1, a_2, a_3, a_4, \dots, a_{d+1}]^T$ are the $d+1$ weights of the polynomial. In the proposed implementations for 2D and 3D sensor network, for a 2D network $\mathbf{A}$ is a 2×3 matrix, whereas for a 3D network it is a 3×4 matrix.

Our approach consists of two phases.

Phase 1: First, the weights $\mathbf{w}$ and affine $\mathbf{A}$ coefficients are calculated. Let L be the set of n reference points whose physical coordinates are known. Eq. (5) can be written as the following linear system of equations:

$$\begin{bmatrix} \mathbf{K}_L & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{A} \end{bmatrix} = \begin{bmatrix} \mathbf{Y} \\ \mathbf{0} \end{bmatrix} \quad (6)$$

where, $\mathbf{Y}$ represents the $n \times 2$ matrix of physical coordinates of reference points. $\mathbf{P}$ is $n \times 3$ matrix whose i^{th} row is equal to $[1, \mathbf{c}_i]$, where $\mathbf{c}_i$ is the topology coordinates of i^{th} reference point, $\mathbf{K}_L$ is $n \times n$ and consists of $K_{ij} = \phi(\|\mathbf{c}_i - \mathbf{c}_j\|)$, and $\mathbf{0}$ is a zero matrix. $\phi(\|\mathbf{c}_i - \mathbf{c}_j\|)$ is the thin-plate spline function $\phi(r) = r^2 \log(r)$. The solution for weights and affine coefficients can be found by solving Eq. (6), however, it is recommended to use the QR decomposition method [11], in which $\mathbf{P}$ is decomposed as:

$$\mathbf{P} = [\mathbf{Q}_1 \ \mathbf{Q}_2] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \quad (7)$$

Here $\mathbf{Q}_1$ is $n \times 3$, $\mathbf{Q}_2$ is $n \times (n-3)$ and $\mathbf{R}$ is 3×3 . Since $\mathbf{P}^T \mathbf{w} = 0$ and $\mathbf{P}^T \mathbf{Q}_2 = 0$ or $\mathbf{Q}_2^T \mathbf{P} = 0$, so ' $\mathbf{w}$ ' has to be in the column space of $\mathbf{Q}_2$. Both sides of $\mathbf{K}_L \mathbf{w} + \mathbf{P} \mathbf{A} = \mathbf{Y}$ are multiplied by $\mathbf{Q}_2^T$ and uses $\mathbf{w} = \mathbf{Q}_2 \mathbf{B}$ [11], resulting in:

$$\mathbf{B} = (\mathbf{Q}_2^T \mathbf{K}_L \mathbf{Q}_2)^{-1} \mathbf{Q}_2^T \mathbf{Y} \quad (8)$$

$$\mathbf{w} = \mathbf{Q}_2 \mathbf{B} = \mathbf{Q}_2 (\mathbf{Q}_2^T \mathbf{K}_L \mathbf{Q}_2)^{-1} \mathbf{Q}_2^T \mathbf{Y} \quad (9)$$

$$\mathbf{A} = \mathbf{R}^{-1} \mathbf{Q}_1^T (\mathbf{K}_L - \mathbf{Y} \mathbf{w}) \quad (10)$$

Phase 2: Using $\mathbf{A}$ and $\mathbf{w}$ estimated in phase 1, now we use Eq. (5) to transform the TPM coordinates of the remaining nodes into MPC. In this phase, $\mathbf{x}$ corresponds to the TC of a sensor location, $\mathbf{c}_i$ corresponds to the TC of the i^{th} reference point and n is number of reference points.

IV. RESULTS AND DISCUSSION

The performance of the proposed TPM-GPA and TPM-TPS methods are evaluated. We also compare the performance with that of DV-Hop [3], even though DV-Hop has several other limitations. First we define the error metric used to evaluate the methods.

The average distance error, which is used to compare the different schemes for 2-D networks, is calculated as follows:

- Calculating the sensor distance error of node i 2D network:

$$E_i = \begin{cases} 0 & \text{if } S'_i \in AR_i \\ \frac{\min_{p \in AR_i} (\|S'_i - p\|_2)}{r_i} & \text{if } S'_i \notin AR_i \end{cases}$$

where S'_i is the estimated location of the sensor i , p are points of the freedom of movement region (AR_i) for sensor i , and r_i is the radio range. E_i is distance from the estimated location of sensor i to the nearest point of AR_i divided by r_i .

As an example, Fig. 2 shows S_i the physical location of sensor i and $N_i = \{N_i^k \text{ where } k=1, 2, 3, \dots \text{ etc.}\}$ the one-hop neighbor set. AR_i is the sub-region or full-region of intersection of radio ranges for N_i . If S'_i is the estimated location then the distance error for the sensor i is zero. On the other hand, if S''_i is estimated location then the error in the sensor localization is the shortest Euclidean distance (d_{ip}). Errors are normalized with respect to the radio range r_i .

- Average network distance error is computed by taking the average of all sensor distance errors in the network.
- Reported error value is obtained by repeating the computation for 100 instantiations, with different random reference point selections of a given set size.

Four different 2D sensor network topologies are used for the evaluation: an odd-shaped network with 550 nodes (with 5 ENS anchors), a spiral-shaped network with 421 nodes with 6 anchors, a building network of 343 nodes (with 6 anchors), and a circular network with three physical voids consisting of 496 nodes with 8 nodes as anchors with five of them selected by ENS and other three randomly selected. All the nodes in the four network topologies have the same communication range which is 1.4 distance unit. In addition, we also evaluate the performance for a 3D network in the shape of a T-joint pipe consisting of 1245 nodes with 30 nodes randomly selected as anchors. DV-Hop is not able to handle 3-D networks thus it was evaluated only using TPM-TPS. The physical layouts of the different networks are shown below. MATLAB® 2013a is used for the computations. Topology

maps are constructed using ENS anchors and plotted in 2D. The general Procrustes alignment based TPM-GPA, DV-Hop and thin-plate spline based TPM-TPS are applied on different size sets of reference points. The number of reference point is varied from 10 to 60 with increment of five. The reference points for each case are selected randomly and the error distance is calculated for each set using TPM-GPA, TPM-TPS and DV-Hop. For each number of reference points, the simulation is repeated for 100 realizations, and the average error distance is calculated over these realizations. Two methods are used to select the reference points: 1) reference nodes are randomly selected from the whole sensors in the network, 2) reference nodes are randomly selected from a subset of the peripheral sensors.

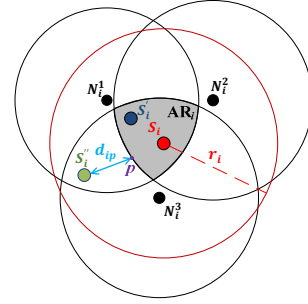


Fig. 2. An example of sensor localization error calculation in network.

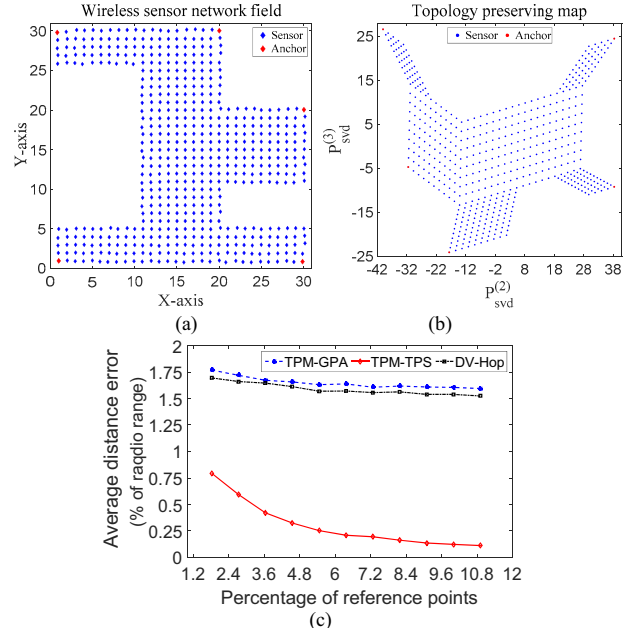


Fig. 3. (a) The odd-shaped network with 550 nodes and 5 anchors; (b) 2D topology preserving map; (c) Average distance error with TPM-GPA, TPM-TPS and DV-Hop.

Figures 3 and 4 illustrates the results for the odd-shaped network. Fig. 3 (a) corresponds to the physical layout of the network, with the anchor nodes highlighted. Fig. 3(b) shows the SVD based TPM. Fig. 3(c) compares the average error for different methods as the fraction of nodes whose coordinates

are known, i.e. reference points, is changed. Fig. 4(a), (b), and (c) show the maps recovered using 50 random reference points by TPM-TPS, TPM-GPA, and DV-Hop respectively. Note that TPM-TPS method significantly outperforms both the TPM-GPA and the DV-Hop in the first topology as demonstrated in Fig. 3 (c). The performance of the TPM-GPA and the DV-Hop are close because the correlation between the hop distance and Euclidean distance is good in this topology. However comparison of Fig. 4 (a), (b) and (c) for TPM-TPS, TPM-GPA and DV-Hop methods, corresponding to a random set of reference nodes, indicate that obtaining physical maps by starting with TPMs (i.e., with GPA or TPS) is significantly better than DV-Hop. The physical maps obtained using TPM-TPS, when reference points are selected randomly from only the peripheral sensors is shown in Fig. 4(d). It has similar performance compared to random reference points throughout the network as depicted in Fig. 4(a). Selecting reference points along the periphery may be more attractive in cases where the network is deployed in a harsh environment. Other two schemes are not shown for this case due to space limitations. However we note that with DV-Hop, the average distance error increases when using only a subset of peripheral sensors as reference points instead of selecting the reference points randomly from all the sensors. For this particular topology, it is sufficient to have 35 (~7%) reference points to obtain a good quality map by using TPM-TPS.

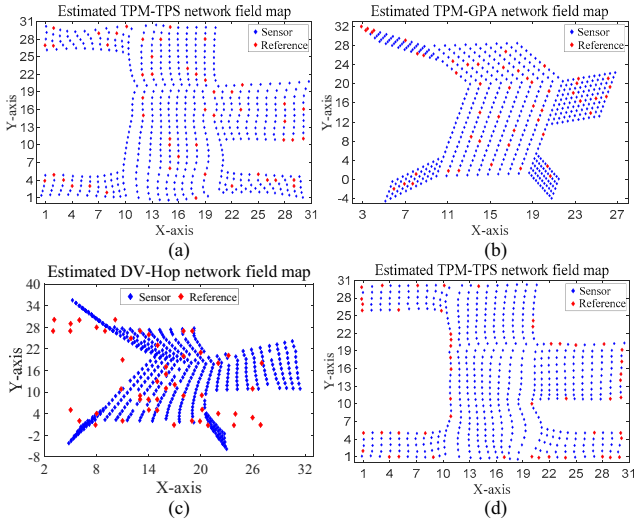


Fig. 4. The odd-shaped network: (a) recovered by 50 random reference points using TPM-TPS; (b) recovered by 50 random reference points using TPM-GPA; (c) DV-Hop recovered map using 50 random reference points; (d) recovered by 50 random reference points from subset of peripheral sensors using TPM-TPS.

Figures 5 and 6 shows the corresponding results for the second network topology which has a spiral shaped layout. Average distance error in Fig. 5 (c) shows that TPM-TPS is again significantly better than TPM-GPA and DV-Hop methods. The performance of even TPM-GPA is better than the DV-Hop because the correlation between the hop distance and Euclidean distance is weak in this topology. The results

produced by TPM-TPS, shown in Fig. 6 (a) and (d) clearly capture the physical shape of the network unlike those in Fig. 6 (b) for TPM-GPA and 6 (c) for DV-Hop. The TPM-TPS based maps in Fig. 6(a) and (d) obtained with TPM-TPS are quite close to the original map for the reference set size between 20 to 25. However, increasing the number of reference points of the TPM-GPA and the DV-Hop will not bring any significant improvement as shown in Fig. 5 (c).

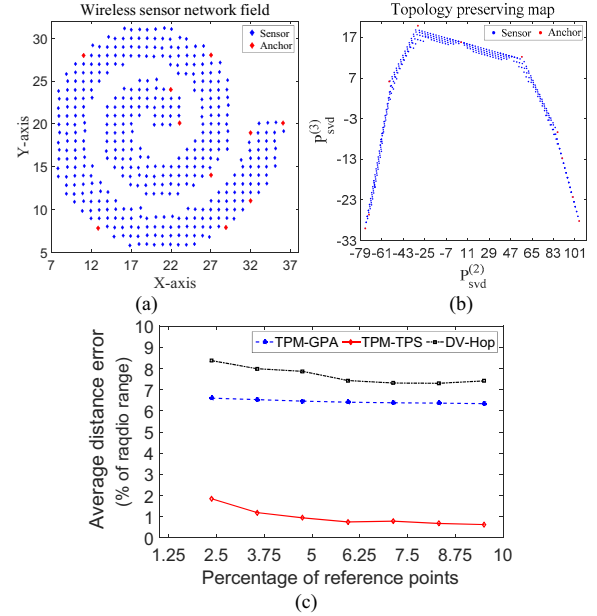


Fig. 5. The spiral-shaped network with 421 nodes and 8 anchors; (a) physical layout; (b) 2D topology preserving map; (c) Average distance error with TPM-GPA, TPM-TPS and DV-Hop.

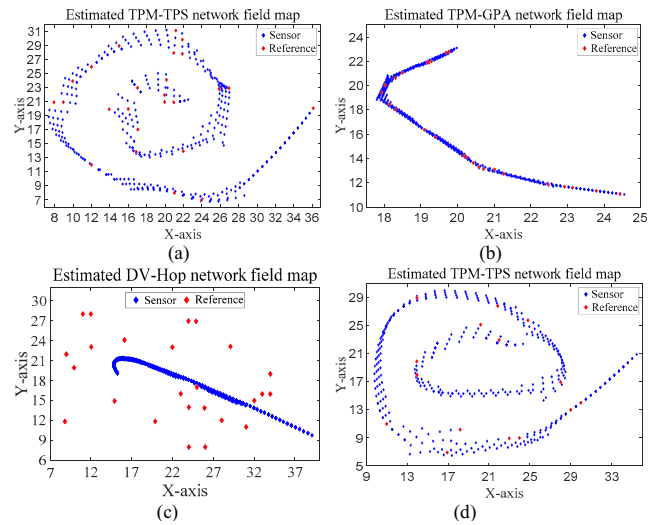


Fig. 6. The spiral-shaped network (a) is recovered by 25 random reference points using TPM-TPS; (b) recovered by 25 random reference points using TPM-GPA; (c) DV-Hop recovered map with using 25 random reference points; (d) recovered by 15 random reference points from subset of peripheral sensors using TPM-TPS.

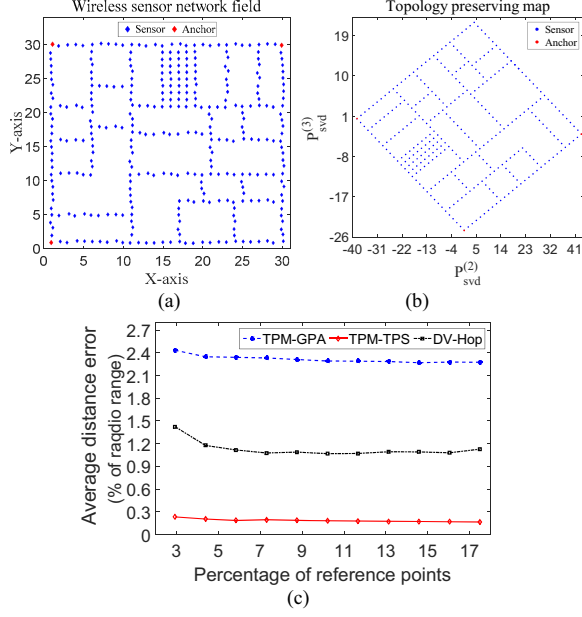


Fig. 7. A building network with 343 nodes and 6 anchors: (a) Physical map; (b) TPM; (c) Average distance error using TPM-GPA, TPM-PS and DV-Hop.

Fig. 7 and 8 show the results corresponding to the third topology, that of a building network. TPM-TPS again outperforms TPM-GPA and DV-Hop. The original map can be recovered with a small set of the reference points. However, increasing the size set of reference points does not cause rapid improvement in the average error, at least within the small range that was selected. The TPM-TPS with random reference points from the subset of the peripheral sensors has the same performance of the TPM-TPS with random reference points as depicted in Fig. 8 (a) and (d). The border nodes capture the distortion as well as the inner nodes do. 93% of sensors in this case are on inner/outer borders, and therefore there is not much of a difference between Fig. 8 (a) and (d). In this topology, a good map can be obtained by using only 10 reference points.

Fig. 9 shows a circular network with three physical voids from [7] and its TPM. The error curves in Fig. 9 (c) show that the TPM-TPS again outperforms both TPM-GPA and DV-Hop. Although the topology is symmetric, the distortion is nonlinear in the TPM because the anchors selected were not in symmetric locations. Fig. 10 show the maps for the network when using 35 reference points.

The results of Fig. 10 (a) and (d) are significantly better than the results of Fig. 10 (b) and (c) even though they use the same number of reference points. Fig. 10 (a) corresponding to 35 reference points randomly selected from the entire network provides a better mapping than Fig. 10 (d), where the 35 reference points are selected from network border. The high nonlinear deformation of the outer border in the TPM causes the degradation of the performance in Fig. 10 (c).

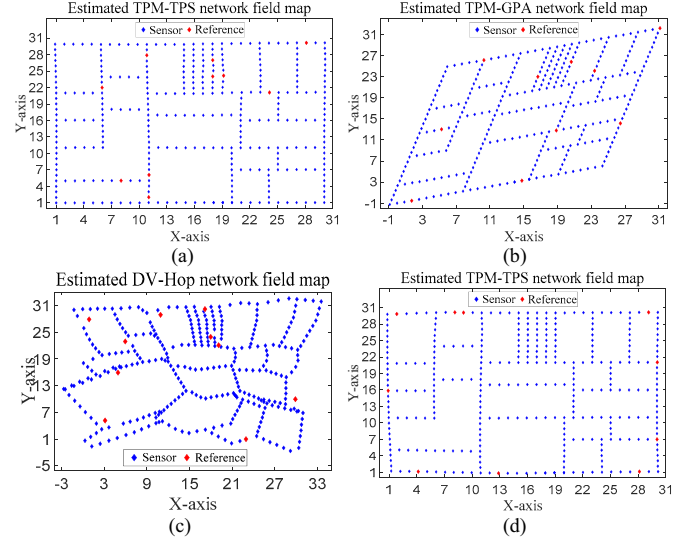


Fig. 8. The building network (a) recovered by 10 random reference points using TPM-TPS; (b) recovered by 10 random reference points using TPM-GPA; (c) DV-Hop recovered map using 10 random reference points; (d) recovered by 10 random reference points from subset of peripheral sensors using TPM-TPS.

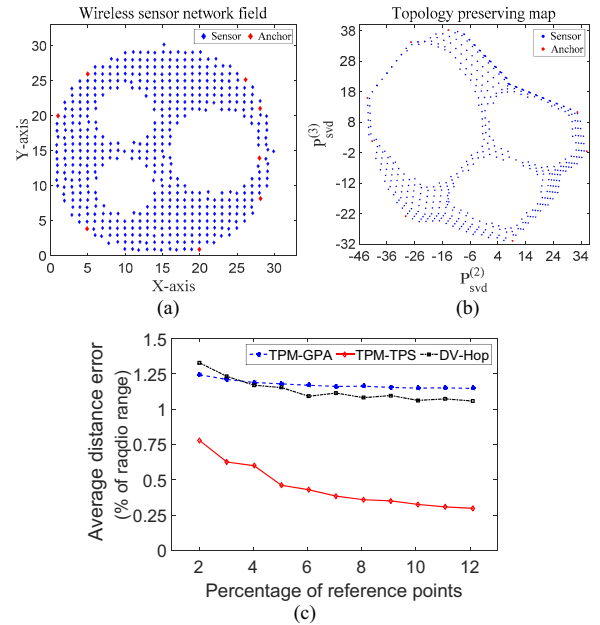


Fig. 9. Circular network with three physical voids with 343 nodes and 6 anchors: (a) Physical map; (b) TPM; (c) Average distance error using TPM-GPA, TPM-TPS and DV-Hop.

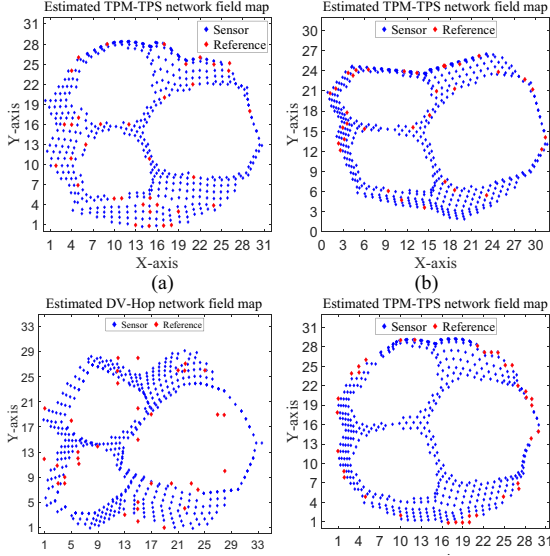


Fig. 10. Circular network with three physical voids: (a) recovered by 35 random reference points using TPM-TPS; (b) recovered by 35 random reference points using TPM-GPA; (c) DV-Hop recovered map using 35 random reference points; (d) recovered by 35 random reference points from subset of peripheral sensors using TPM-TPS from border.

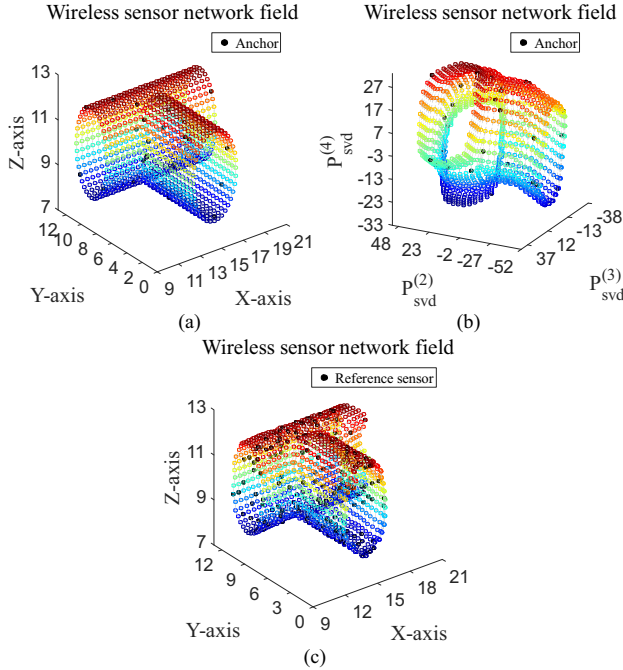


Fig. 11. 3D grid network, T-joint pipe consist of 1245 nodes with 30 random anchors; (b) 3D topology preserving map using random anchors; (c) Network map is recovered by 200 random reference points using the TPM-TPS.

In Fig. 11, we show the results from TPM-TPS method for a 3-D network. Unlike DV-Hop, TPM-TPS can also be applied to recover 3D networks. Fig. 11 (b) shows the TPM generated using 30 random reference points for the physical network map of Fig. 11(a). As shown in Fig. 11 (c), an accurate 3D

maps can be recovered by TPM-TPS using only 200 reference nodes compared to the total of 1245 nodes.

V. CONCLUSION AND FUTURE WORK

A Thin-Plate Spline (TPS) based method to transform a localization free TPM to an approximate physical map is proposed when the physical coordinates of a small subset of nodes are known. Comparison of the propose TPM-TPS with existing DV-Hop method for localization as well as a Generalized Procrustes Analysis (GPA) based TPM-GPA method shows that TPM-TPS significantly outperforms those schemes. The GPA defines a global linear transformation, and the DV-Hop attempts to estimate the hop size. The relation between TPM and the physical map suffers from local nonlinearities. Therefore, the GPA based mapping enhancement is not able to adapt when the number of the reference points increases. TPM-TPS method adapts well to the number of reference points, and the recovered map from the TPM-TPS converges to the physical map as the number of reference nodes whose locations are known increases. TPM-TPS also works well even when reference nodes are selected only from nodes on the boundary. Furthermore, the proposed TPM-TPS approach seamlessly adapts to 3-D networks as well, whereas DV-Hop is restricted to 2-D networks. Note that DV-Hop is an existing method that aims to localize nodes, whereas the proposed schemes (TPM-TPS and TPM-GPA) start with TPMs. Thus even if no reference points are available, they are capable of producing maps that preserve some physical layout information. As reference points are added, the derived physical maps continue to improve. In contrast, with DV-Hop, a critical number of reference points are needed. TPM-TPS method does not require that the anchors also be the reference points. The coordinate generation complexity in DV-Hop increases significantly with the number of reference points as they also are anchors. Therefore with TPM-TPS, maps can be incrementally improved as new reference points become available without regenerating the virtual coordinates. As the spiral network example shows, the DV-Hop method fails to recover physical coordinates in complex shapes. In case of DV-Hop, increasing the number of anchors significantly increases the coordinate generation cost, whereas with TPM-TPS, the number of anchors are independent of the number of reference points.

ACKNOWLEDGMENT

The authors thank their colleagues in the Computer Networking Research Laboratory for their useful comments.

REFERENCES

- [1] J. Bachrach and C. Taylor, "Localization in sensor networks," *Handbook of Sensor Networks*. Stojmenovic (Editor), John Wiley 2005.
- [2] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th annual*

international conference on Mobile computing and networking, 2003, pp. 81-95.

- [3] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Global Telecommunications Conference*, 2001. GLOBECOM '01. IEEE, 2001, pp. 2926-2931 vol.5.
- [4] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from Connectivity in Sensor Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961-974, Nov. 2004.
- [5] D. C. Dhanapala and A. P. Jayasumana, "Topology preserving maps from virtual coordinates for wireless sensor networks," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, Oct. 2010, pp. 136-143.
- [6] D. C. Dhanapala and A. P. Jayasumana, "Anchor selection and topology preserving maps in WSNs—a directional virtual coordinate based approach.," in *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, 2011, pp. 571-579.
- [7] D. C. Dhanapala and A. P. Jayasumana, "Topology preserving maps—extracting layout maps of wireless sensor networks from virtual coordinates," *Networking, IEEE/ACM Transactions on*, vol. PP, pp. 1-1, 2013.
- [8] R. Rahmatizadeh, S. A. Khan, A. P. Jayasumana, D. Turgut and L. Bölöni, "Routing towards a mobile sink using virtual Coordinates in a wireless sensor network," *Proc. IEEE International Conference on Communications (ICC 2014)*, June 2014.
- [9] I. Borg and P. J. Groenen, *Modern multidimensional scaling: Theory and applications*: Springer, 2005.
- [10] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 11, pp. 567-585, 1989.
- [11] Y. Wang, *Smoothing splines: Methods and applications*: Chapman & Hall/CRC Press, 2011.