This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

On Boundary Detection of 2-D and 3-D Wireless Sensor Networks*

Dulanjalie C. Dhanapala, Anura P. Jayasumana and Sahil Mehta
Department of Electrical and Computer Engineering,
Colorado State University, Fort Collins, CO 80523, USA
{dadcd, anura, smehta85}@engr.Colostate.edu

***Abstract*— A novel method of identifying boundaries of wireless sensor networks deployed on 2D and 3D surfaces is presented. It does not require costly, error prone localization algorithms or physical locations of nodes. Instead, a Virtual Coordinate System (VCS) is used in which each node is characterized by the hop-distances to a set of randomly selected nodes known as anchors. To use geometric relationships for boundary detection, it transforms the VCS to a Topology Preserving Map (TPM). A TPM generation scheme for networks deployed on 3D surfaces is derived as well. The boundary detection scheme proposed is simple, not computationally intensive, energy efficient, and can be used with physical coordinates as well. Five representative example networks show the proposed scheme to be effective, with 100% of boundary nodes identified correctly with no erroneous identification of non-boundary nodes as boundary nodes. Use of TPM based boundary detection scheme for detecting dynamic event boundaries, such as those of plumes, in a distributed manner is also illustrated.**

Keywords - Boundary detection, Singular Value Decomposition, Topology-Preserving Map, Wireless Sensor Networks

I. INTRODUCTION

Boundary detection plays a crucial role in information fusion and dissemination in 2D and 3D Wireless Sensor Network (WSN) applications such as target tracking, plume tracking, forest fires, animal migration, underwater WSNs and surveillance applications. It is also often important for self-organization of networks. A network has a specific embedding and can have three different types of boundaries which the scheme presented in this paper aims at detecting. First is network's outer boundary which consists of a unique subset of nodes. Second is an inner boundary. The last type of boundary is an event boundary. For example events such as mobile targets or forest fires have highly dynamic event boundaries while an underground chemical plume may have boundaries that change gradually over time.

Currently available boundary detection schemes that have been targeted exclusively at 2D networks can be broadly categorized as physical information-based and topological /connectivity information-based [1][4] schemes. The former uses physical position of nodes to identify the boundary while the latter uses topological/connectivity information of the network. Physical domain schemes rely on node location or physical position information obtained using localization algorithms or GPS. Equipping nodes with GPS is costly and infeasible for many applications. Localization based on parameters such as RSSI/time delay is error-prone even for 2D networks of modest size, is susceptible to interference, multipath and fading, which makes it impractical in many environments. Future sensor networks may have thousands or even millions of sensors, and hence distributed strategies that do not accumulate errors, and scalable in cost and complexity are of significant interest.

An alternative approach is connectivity based boundary detection [10][12]. A connectivity domain description of a network can have more than one valid embedding (configurations) [12] in physical domain, even though only one of them corresponds to the physical network. The actual embedding is one out of the many, but identifying the correct embedding solely based on the connectivity information is challenging. Hence, connectivity information based boundary identification captures a union of boundary nodes in every embedding. As a result the actual set of boundary nodes is a subset of it which leads to identifying a band of nodes as boundary nodes [12]. Due to such difficulties there is no connectivity based approach available to identify boundaries on 3D surfaces to the best of our knowledge.

Boundary detection in connectivity domain requires two steps: (1) identifying the correct embedding, and (2) detecting boundary. We propose a novel two step connectivity based approach for boundary detection. It produces highly accurate results by overcoming the ambiguity of network boundary due to multiple embeddings in connectivity domain. It uses a Virtual Coordinate System (VCS) to generate a Topology Preserving Map (TPM) that identifies the correct physical embedding. In VCS, a subset of nodes is selected as anchors [3]. Then all the nodes in the network including anchors estimate their shortest path hop distance to the anchors and use those values as virtual coordinates. Number of anchors is the cardinality of the coordinates. TPM is simply a map of the original network, in the original physical dimensionality, in which the neighborhood is preserved. 2D topology preserving map (TPM) generation based on virtual coordinate system (VCS) is discussed in [2]. The technique does not involve measuring signal strengths or time delays, which are costly and often impractical to implement in large scale networks.

This paper is also the first to address the generation of topology preserving maps of 3D surface networks. Emerging technologies point to many applications for such networks. For example, an oil pipeline, a boiler or a bridge that needs to be monitored for corrosion, temperature distribution, or structural integrity. Tiny nano-sensors capable of wireless communication and minimal computation capability can be deployed in massive quantities on their surfaces.

*This research is supported in part by NSF Grant CNS-0720889

In Section II, the related work is discussed. Next in Section III methodology of 3D topology preserving surfaces is proposed. Then, inner, outer and event boundary recognition of 2D and 3D networks are discussed in Section IV. Section V evaluates the performance of the proposed boundary detection algorithm, while Section VI concludes our work.

II. RELATED WORK

A. Network boundary detection

A decentralized localized algorithm to identify the perimeter nodes using a barycentric technique on neighborhood information is described in [9]. In [5], sensor nodes remotely collect data about various points on the boundary and estimate the boundary along with the confidence intervals using a regression relationship among sensor locations and the distances to the boundary. A Voronoi and neighbor-embracing polygon based localized boundary detection approach is discussed in [6], while [11] proposes a localized perimeter detection algorithm for dense networks based on the angle between nodes. A localized algorithm for 3D boundary detection based on unit ball fitting followed by a reinforcement algorithm named isolated fragment filtering is presented in [13] for a network with known node locations.

In [10], nodes identify patterns called 'flowers'. If a flower exists, the node is an internal node. In an extension of this work in [12], nodes try to identify a family of patterns by defining a set of rules. If a node satisfies the defined set of rules it is an internal node. An isocontours based hole boundary detection scheme is proposed in [6]. This algorithm requires significant computational power.

B. Event boundary detection

The goal of event boundary detection is to detect the profile or the contour of a region or a surface over which the event has occurred. Examples of events include spread of a chemical plume, or contour of a segment of field which needs application of fertilizer. Reference [1] proposes an algorithm for detecting event boundaries based on a Gaussian mixture model. Main disadvantages are the uncertainty associated with the probabilistic prediction and the complexity. Three different schemes based on localized algorithms to identify event boundaries are proposed in [1], namely statistical approach, classifier-based approach and image processing approach. A median-based localized approach is presented in [4] for faulty sensor identification and fault-tolerant event boundary detection. A noise-tolerant algorithm for event and event boundary detection based on moving averages to eliminate noise effects in evenly distributed localized WSNs is presented in [8].

III. TOPOLOGY PRESERVING MAPS OF 3D SURFACES

Topology preserving map (TPM) [2], in the present context, refers to a map of a network that preserves the connectivity topology of a network while capturing the physical properties of the network such as its shape, internal and external boundaries, and their relationships. Recent research [2] provides an intriguing approach for extracting TPMs of 2-D networks from hop distances to a small set of nodes. Here we extend it to 3D surface networks.

Consider a network with N nodes. Denote the i^{th} node by N_i $(1 \leq i \leq N)$. A VC system is used in which each node is characterized by a vector of virtual coordinates denoting the distances to each of a set of M anchors ($N >> M$). A_m $(1 \leq m \leq M)$ denotes the m^{th} anchor. Note that an anchor is one of the N nodes, but we use A_m for clarity. Let $n\left(N_i, N_j\right)$ denote the hop distance between nodes N_i and N_j; $n\left(N_i, N_j\right)$ is zero when $N_i = N_j$, otherwise it is a positive integer. A node N_i is thus identified by the vector of M virtual coordinates given by $[n(A_1, N_i), n(A_2, N_i), \ldots\ n(A_M, N_i)]$, where, $n[A_m, N_i]$ is the number of hops from N_i to the m^{th} anchor A_m.

Let P be the $N \times M$ matrix containing the virtual coordinates of the nodes of the network. i^{th} row of P is corresponding to the VC of node N_i. Let the Singular Value Decomposition (SVD) of P be

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

$$P_{SVD} = P.V \tag{2}$$

where, U, S and V are $N \times N$, $N \times M$, and $M \times M$ matrices respectively. U and V are unitary matrices, i.e., $U^T U = I_{N\times N}$ and $V^T V = I_{M\times M}$. S has non-zero elements only at $S(i,i)$ which are called singular values. P_{SVD} is the projection of P on to basis V. Columns of P_{SVD} are the Principle Components (PCs).

2D topology maps generated in [2] is based on second and third columns, $[{P_{SVD}}^{(2)}, {P_{SVD}}^{(3)}]$, of P_{SVD}. We extend it to 3D by taking second, third, and fourth columns of P_{SVD}, to provide a set of 3-dimensional Cartesian coordinates for node positions on a 3D topology preserving map. Therefore topological coordinates of node N_i can be written as,

$$[X_T, Y_T, Z_T]_{(i)} = [{P_{SVD}}^{(2)}, {P_{SVD}}^{(3)}, {P_{SVD}}^{(4)}]_{(i)} \tag{3}$$

where $[{P_{SVD}}^{(j)}]_{(i)}$ is j^{th} PCs of node N_i.

The extracted Topology Coordinate System for 3D surfaces, like its counterpart for 2-D [2], possesses many of the properties of the physical coordinate system. In network realization of the TPM algorithm is broadly discussed in [2][2]. Same implementation can be extended to 3D.

IV. SENSOR NETWORK BOUNDARIES

Next we address the identification of the nodes forming the network boundary. Three types of boundaries are of interest: (a) network outer boundaries, (b) network inner boundaries, and (c) event boundaries. Consider the example network in Fig. 1(a). Each node can communicate with its 1 hop neighbors, and up to four neighbors are possible. We define the boundary of a network as the set of nodes that has a contribution toward the outer bound of the network communication (can also be interpreted to mean sensing) coverage. Hence the identified boundary nodes need not to be connected in the communication topology. Consider the coverage by the entire network as in Fig 1(b). If the outer physical loop of the coverage is considered, then the nodes that have contributed to this loop are boundary nodes. Outer boundary and the inner boundary of the network are as

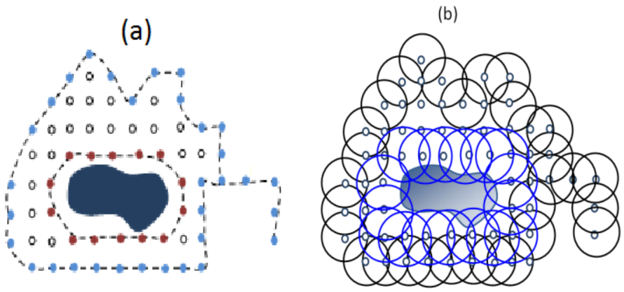

Figure 1. (a) Example network with a lake (b) Outer bound of the coverage of the network nodes.

indicated in Fig 1(a). Network boundary is a function of a nodes' communication range and node density of the network. Boundary nodes in the 3D case can be defined the same way.

First, the detection of boundaries in 2D networks is considered, and then the algorithm is extended to 3D network surfaces.

A. 2D network boundary detection

Consider a 2D network. In a connected network any node has one or more neighbors. Let the node, which is to be tested for a boundary node be N_i. Assume it has k neighbors denoted by $K_i = \{N_j\}, j = 1, \dots, k$. If k is 1 or 2, N_i is a boundary node. When k is 3 or greater an algorithm is required to check whether N_i is a boundary node or an internal node.

Consider the case where N_i has three or more neighbors. Let the topological coordinates of N_i be $[x_{T,i}, y_{T,i}]$. Select any three neighbors $\mathrm{N_{k1}}, \mathrm{N_{k2}}$ and $\mathrm{N_{k3}}$ (Fig. 2). If N_i is an internal node, area of triangle $\mathrm{N_{k1}N_{k2}N_{k3}}$, denoted by $\Delta\mathrm{N_{k1}N_{k2}N_{k3}}$ is the same as the total area enclosed by $\Delta\mathrm{N_{ki}N_{k2}N_{k3}}$, $\Delta\mathrm{N_{k1}N_{ki}N_{k3}}$, and $\Delta\mathrm{N_{k1}N_{k2}N_{ki}}$. If $\mathrm{N_i}$ is an external node, area of $\Delta\mathrm{N_{k1}N_{k2}N_{k3}}$ is lower than the total area enclosed by $\Delta\mathrm{N_{ki}N_{k2}N_{k3}}$, $\Delta\mathrm{N_{k1}N_{ki}N_{k3}}$, and $\Delta\mathrm{N_{k1}N_{k2}N_{ki}}$. Fig. 2 illustrates this relationship. If N_i is an external node for all of the triplets of its neighbors, then N_i is a boundary node. If there exist at least single triplet of neighbors where N_i is an internal node, N_i is not a boundary node. Instead of evaluating trangular areas, polygonal area evaluation can be used in the case of $k > 3$. Again the area of a polygon can be calculated in terms of triangles. In this paper we consider the relatinship in terms of triangles for the boundary detection.

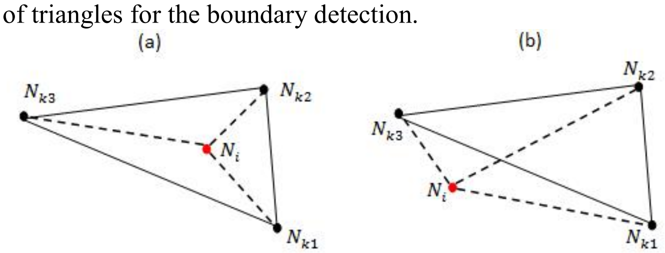

Figure 2: Relationships among a) an internal node with respect to its 3 neighbors and b) a external node with respect to its 3 neighbors.

The L[2] distance between nodes N_i and N_j in the network based on their topological coordinates, $D_{N_iN_j}$ is given by,

$$D_{N_iN_j} = \sqrt{\left(X_{T,i} - X_{T,j}\right)^2 + (Y_{T,i} - Y_{T,j})^2} \quad (10)$$

The area $\Delta\mathrm{N_{k1}N_{k2}N_{k3}}$ can thus be found in terms of $D_{N_{k1}N_{k2}}$, $D_{N_{k3}N_{k2}}$ and $D_{N_{k1}N_{k3}}$ using the Heron's formula:

$$\Delta N_{k1}N_{k2}N_{k3} = \sqrt{S\left(S - D_{N_{k1}N_{k2}}\right)\left(S - D_{N_{k1}N_{k3}}\right)\left(S - D_{N_{k3}N_{k2}}\right)} \quad (5)$$

where S is the semi perimeter of the triangle

$$S = \frac{1}{2}\left(D_{N_{k1}N_{k2}} + D_{N_{k1}N_{k3}} + D_{N_{k3}N_{k2}}\right) \quad (6)$$

The boundary detection algorithm is summarized in Fig. 3. Once the topology coordinates are known, the additional worst case computational complexity is $O(max(|K|)^2)$ while the memory complexity is $O(max(|K|))$. $max(|K|)$ is the largest number of neighbors a node can have.

Input: $[X_T, Y_T, Z_T]_{(j)}, j \in K_i$
Output: Is N_i a boundary node?
If $|K_i| \le 2$
  N_i is a boundary node
Elseif $|K_i| > 2$
  For all possible neighbor triplets $N_{k1}, N_{k2}, N_{k3} \epsilon K_i$
    If *area of any of the three triangles* $\Delta N_{k1}N_{k2}N_{k3}$ OR $\Delta N_iN_{k2}N_{k3}$ OR $\Delta N_{k1}N_{k2}N_i$ OR $\Delta N_{k1}N_iN_{k3} == 0$
      N_i is a boundary node candidate
    Elseif $\Delta N_{k1}N_{k2}N_{k3} > \Delta N_iN_{k2}N_{k3} + \Delta N_{k1}N_{k2}N_i + \Delta N_{k1}N_iN_{k3}$
      N_i is an internal node
      Stop checking
    Elseif $\Delta N_{k1}N_{k2}N_{k3} == \Delta N_iN_{k2}N_{k3} + \Delta N_{k1}N_{k2}N_i + \Delta N_{k1}N_iN_{k3}$
      If $\Delta N_iN_{k2}N_{k3}$ AND $\Delta N_{k1}N_{k2}N_i$ AND $\Delta N_{k1}N_iN_{k3} \sim= 0$
        N_i is an internal node
        Stop checking
      Elseif $\Delta N_iN_{k2}N_{k3}$ OR $\Delta N_{k1}N_{k2}N_i$ OR $\Delta N_{k1}N_iN_{k3} == 0$
        N_i is a boundary node candidate
      End
    Elseif $\Delta N_{k1}N_{k2}N_{k3} < \Delta N_iN_{k2}N_{k3} + \Delta N_{k1}N_{k2}N_i + \Delta N_{k1}N_iN_{k3}$
      N_i is a boundary node candidate
    End
  End
  If N_i is a boundary node candidate for all triplets of neighbors in K_i
    N_i is a boundary node
  End
End

Figure 3: Pseudocode of boundary detection algorithm.

During our experiments we observe that if nodes N_{k2}, N_{k3} and N_i lie on the boundary of the network, algorithm will fail to identify N_i as a boundary node.

B. Boundary detection in 3D surfaces

When we consider WSNs on 3D surfaces, the distance between two nodes may be the curvilinear distance between them and not the Line-of-Sight distance. But for our boundary detection algorithm operates in a one hop neighborhood, and hence the LOS (Line-of-Sight) distance is a reasonable approximation sufficient for decision making. L^2 is used to approximate for the curvilinear distance. However, in area

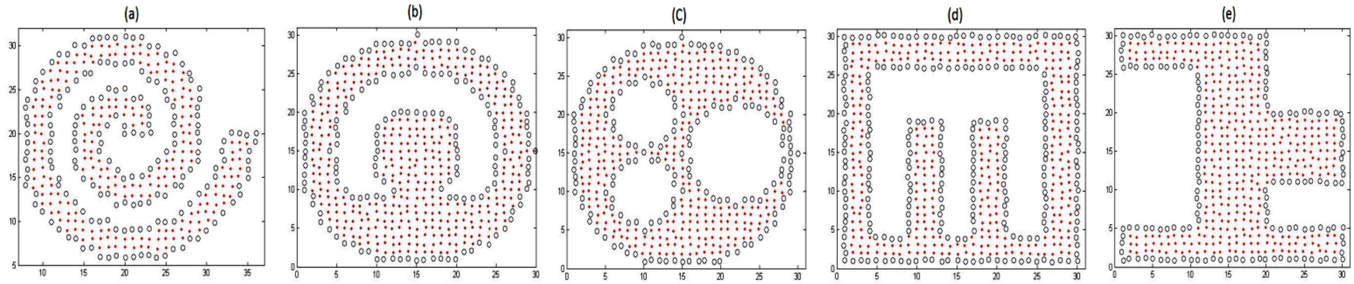

Figure 4: Boundary detection results for different network shapes: (a) a spiral network, (b) a circular network with a C shaped concave void, (c) a circular network with three voids, (d) a square network with an E shaped void, and (e) an odd shaped network.

evaluation, all three coordinates are used for distance evaluation in 3D domain using L^2 distance calculation.

D. Event boundary detection

To detect the boundary corresponding to the set of nodes detecting a certain event, the nodes detecting the event set an internal flag, 'event detected' or E. The event boundary is detected by executing the algorithm only on the subset of nodes with E set. The algorithm can easily be extended to detect contours of the sensed phenomenon, e.g., by having multiple flags corresponding to different contours or using the actual sensed value for E instead of it being a Boolean flag.

V. SIMULATION RESULTS: NETWORK AND EVENT BOUNDARY DETECTION

The performance of the boundary detection algorithm is evaluated next. We use the five example networks with convex and concave boundaries that are representative of a variety of networks. The number of nodes range from 300 to 800. The networks are (a) a spiral network, (b) a circular network with a C shaped concave void, (c) a circular network with three voids, (d) a square network with an E shaped void , and (e) an odd shaped network, and. MATLAB® 2009b based simulator was used for the computations. Ref. [2] indicates that 10 randomly selected anchors result in good topology maps and therefore our simulations use 10 anchors. Recently proposed anchor placement scheme in [3] achieves similar performance with less number of anchors.

Existing connectivity information based boundary detection schemes [10][12] evaluate their effectiveness through visually inspection. Instead we define and use two formal metrics that capture the accuracy of boundary identification (A%) and the error in boundary identification (E%) as follows:

$$\mathrm{A\%} = \frac{\#\ \text{nodes that are correctly identified}}{\text{Total number of Boundary nodes}}\ \%$$

$$\mathrm{E\%} = \frac{\#\ \text{nodes that incorrectly identified}}{\#\ \text{nodes that are correctly identified}}\ \%$$

A. Network inner and outer boundary identification

In the simulation a single hop neighborhood was considered. Results for boundary detection for the five networks are shown in Fig. 4. As it can be clearly seen for networks in Fig. 4 (a)-(e) value is zero for $E\%$ and 100% for $A\%$. This demonstrates the effectiveness of the proposed scheme in detecting the correct network boundary without need for physical coordinates. Due to the implementation complexity we were unable to evaluate $A\%$ or $E\%$ of [10][12] for proper comparison. But in terms of communication and computational complexity as well as visual inspection our scheme clearly outperforms the connectivity based schemes in [10][12].

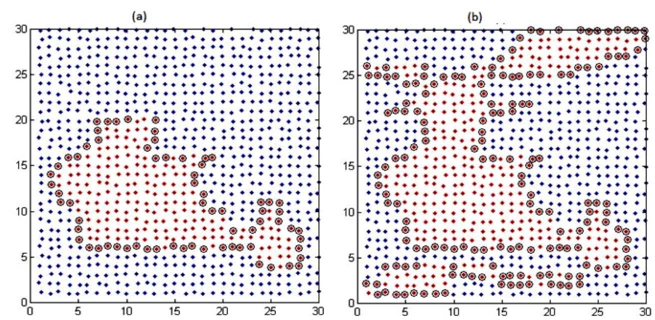

Figure 5: An example of event boundary detection. Red nodes are the event detected nodes.

B. Event boundary detection

In this section the effectiveness of the algorithm for event boundary detection is analyzed. We simulated the case where there is an event where the boundary changes with time, e.g., a forest fire or a chemical plume. Fig 5 indicates two time instances of a synthetic event that expands with time. Proposed algorithm identifies and tracks the event boundary as illustrated in Fig. 5. The value of $E\%$ of event boundary detection is zero, but the $A\%$ is 90%, when averaged over the two cases in Fig. 5(a) and (b) respectively. This is the first scheme that identifies a dynamic event boundaries without using physical information of the network.

C. Topology Preserving Maps and Boundary detection of 3D surfaces

Two example networks deployed on 3D surfaces are considered as shown in Fig. 6 to illustrate the effectiveness of proposed 3D topology map generation.

a. T-joint: A pipeline structure joining two perpendicular cylinders a T joint. There are 512 nodes, each with a communication range of 1. 20 randomly placed anchors were used.

b. Cylinder with a hole: A cylindrical structure of radius 2.54 and height 24 with a hole through it (two aligned voids on the surface on opposite sides) is covered with 490 nodes, each with a communication range of 0.5. 15 randomly placed anchors were used.

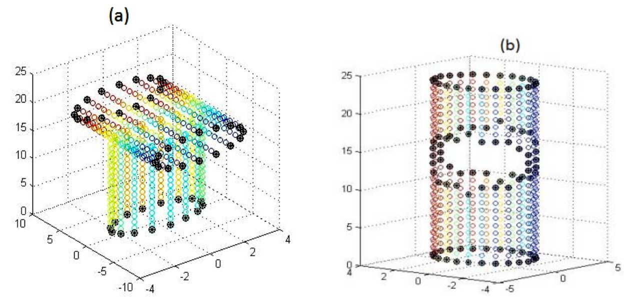

Figure 6. (a) Two perpendicular cylinders (T joint); (b) A cylinder with a hole (two voids on opposite ends). Black colored nodes are identified boundary nodes.

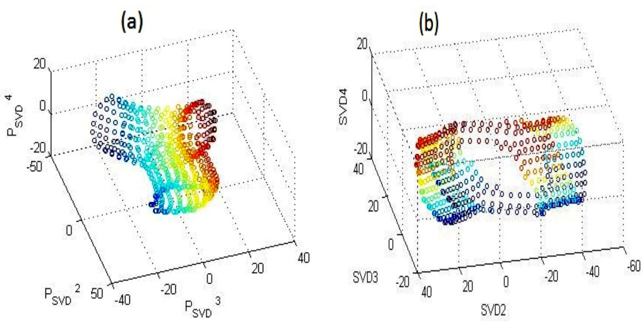

Figure 7: Topology preserving map (a) T joint (b) cylinder with a hole.

Topology preserving maps of the corresponding physical topologies are shown in Fig. 7. The results clearly demonstrate the effectiveness of the topology preserving map generation for network deployed on 3-D surfaces without using any physical information.

Next we apply the 3D surface boundary detection algorithm for the two networks. Results are illustrated in Fig. 6 for the T-joint and the cylinder with holes. The algorithm is able to detect the boundaries with A%= 100% accuracy, and E%= 0%.. To our knowledge no other algorithm exists for 3D surface boundary detection based on connectivity information. We attribute to the complexity of the problem of mapping connectivity to 3D shapes by generating TPMs.

VI. CONCLUSION

A boundary detection scheme is proposed, which does not require physical locations of nodes by relying on Topology Preserving Map (TPM) generated using virtual coordinates of the network. TPMs overcome the challenge of identifying the actual physical embedding of the network from virtual coordinates out of all the possible connectivity based embeddings. Moreover, paper discusses TPM generation of 3D surfaces for the first time. Five representative 2D networks, two simple example 3D networks and a synthetic dynamic event example show the effectiveness of the algorithm's boundary detection capability. Even though we illustrate the capability of the boundary detection using connectivity based approach, scheme can be used with physical information as well.

Boundary detection is based on Eucledean geometric relationships. The mapping from physical coordinates to VCS is not linear, and therefore geometric relationships in physical space are not preserved in VCS, and not recoverable using Euclidean geometric relationships. However, the topology coordinate system, both for 2D and 3D surfaces, can be used to recover such geometric relationships, This is an example of the significance of the TPM extraction; starting with anchor based VCs, it easily extracts coordinates in the topology space which empowers the use of algorithms, geometric relationships, etc, that have been designed for Euclidean spaces such physical space in which the nodes are deployed.

REFERENCES

[1] M. Ding, X. Cheng, "Robust event boundary detection in sensor networks - A mixture model based approach," IEEE INFOCOM 2009, 19-25 April 2009,pp: 2991 – 2995.

[2] D. C. Dhanapala and A. P. Jayasumana, "Topology preserving maps from virtual coordinates for wireless sensor networks," Proc. 35rd IEEE Conf. on Local Computer Networks, Oct. 2010.

[3] D.C. Dhanapala and A.P. Jayasumana, "Anchor Selection and Topology Preserving Maps in WSNs - a Directional Virtual Coordinate Based Approach," 36th Annual IEEE Conference on Local Computer Networks (LCN) 2011. (Accepted)

[4] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-folerant event boundary detection in sensor networks," IEEE INFOCOM, March 2005, pp: 902 – 913.

[5] S. Duttagupta, K. Ramamritham, and P. Ramanathan, "Distributed boundary estimation using sensor networks," Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference, Vancouver, October 2006, pp. 316-325.

[6] S. Funke, "Topological hole detection in wireless sensor networks and its applications," Proc. 3rd ACM/SIGMOBILE International Workshop on foundations of Mobile Computing, DIAL-M-POMC, 2005.

[7] R. Hartley and A. Zisserman,"Multiple view geometry in compute vision," Appendix 4, 2nd Edition, Cambridge University PressNew York,2003.

[8] G. Jin and S. Nittel, "NED: An efficient noise-tolerant event and event boundary detection algorithm in wireless sensor Networks," Proc. of 7th Int. Conf. on Mobile Data Management, 2006.

[9] A. M. Khedr, W. Osamy and D. P. Agrawal, "Perimeter discovery sensor network," Journal of Parallel and Distributed Computing, November 2009,pp: 922-929.

[10] A. Kroller, S.P. Fekete, D. Pfisterer,and S. Fischer, "Deterministic boundary recognition and topology extraction for large sensor networks," Proc. of 17th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 06.

[11] F. Martincic, and L. Schwiebert, "Distributed perimeter detection in wireless sensor networks," Wayne State University, WSU-CSC-NEWS/03-TR03, July 2004.

[12] S. Olga, S. Robert, G. Matthias, and M. P. José, "On boundary recognition without location information in wireless sensor networks," Proc. of 7th Int. Conf. on Information Processing in Sensor Networks, April 22-24, 2008., pp:207-218.

[13] H. Zhou, S. Xia, M. Jin, and H. Wu ,"Localized Algorithm for Precise Boundary Detection in 3D Wireless Networks." 2010 IEEE 30th International Conference on Distributed Computing Systems (ICDCS), 21-25 June 2010, pp: 744 – 753.