Cut Graph Based Information Storage and Retrieval in 3D Sensor Networks with General Topology

Yang Yang, Miao Jin, Yao Zhao, and Hongyi Wu

***Abstract*—We address the problem of in-network information processing, storage, and retrieval in three-dimensional (3D) sensor networks in this research. We propose a geographic location free double-ruling-based scheme for large-scale 3D sensor networks. The proposed approach does not require a 3D sensor network with a regular cube shape or uniform node distribution. Without the knowledge of the geographic location and the distance bound, a data query simply travels along a simple curve with the guaranteed success to retrieve aggregated data through time and space with one or different types across the network. Simulations and comparisons show the proposed approach with low cost and a balanced traffic load.**

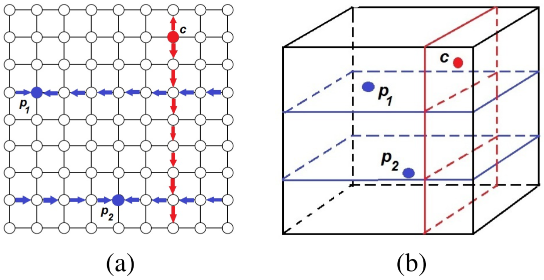
I. Introduction

This research focuses on in-network data-centric information storage and retrieval in large-scale three-dimensional (3D) sensor networks, aiming to support a variety of applications that require scalable and energy-efficient tracking, processing, and storage of many simultaneously detected events within the network as well as delivering the requested data to the interested queries. We first summarize existing in-network data storage and retrieval algorithms for two-dimensional (2D) networks, and then provide an overview of our proposed approach in 3D.

A. An Overview of Distributed Data Storage and Retrieval Algorithms

Geographical hash table (GHT) [1] is one of the earliest approaches for in-network data-centric storage in sensor networks. A basic GHT scheme hashes a datum by its type into geographic coordinates and stores at the sensor node geographically nearest to such coordinates. Queries apply the same hash table with the desired type to retrieve data from the storage node. To avoid creating a hotspot of communication and storage at the node where many data with the same type are hashed to, GHT based schemes apply a structured replication with multiple mirrors scattered in the network. Structured replication reduces the cost of storage but increases the cost of queries.

Different from GHT, a double-ruling scheme works as follows. A datum (or a pointer to the datum) is duplicated along a curve called replication curve, and a query travels along another curve called retrieval curve. Successful retrieval is guaranteed if the retrieval curve intersects the replication curve. A simple double-ruling scheme on a planar grid is illustrated in Figure 1 (a) where nodes are located at lattice points. The replication curves follow the horizontal lines and the retrieval curves follow the vertical lines. By traveling along a vertical line, a data query can always find the requested data.



(a) (b)

Fig. 1. A simple double-ruling scheme on (a) a 2D grid sensor network; (b) a 3D grid sensor network. p_1 and P_2 are two information producers with their replication routes marked in blue. c is an information consumer with its retrieval route marked in red.

Double-ruling based schemes support efficient data retrieval, since all data with different types generated in a network can be conveniently retrieved along one retrieval curve. This is in a sharp contrast to GHT based schemes where a query has to visit multiple nodes scattered in the network to collect data with different types hashed to various locations. Moreover, with modestly increased data replication, double-ruling based schemes have well balanced load across the network, while nodes near the hashed location suffer much higher traffic load than others in the GHT scheme. Double-ruling also has better fault tolerance against geographically concentrated node failure by replicating data on nodes that are uncorrelated with node proximity.

With all the desired properties, double-ruling based schemes have harder constraints on the shape of a sensor network than GHT based schemes. Previous double-ruling based schemes either assume networks with 2D grid shape [2]–[4] or with heavy data replication to achieve high probability that the retrieval curve would meet one of the replication curves within the sensor network field [5]. To extend double-ruling scheme to networks with uneven sensor distribution and irregular geometric shapes, landmark-based scheme [6] is proposed to partition the sensor field into tiles. GHT is adopted at the tile level, i.e., a data type is hashed to a tile instead of a single node. Inside each tile, a double-ruling scheme is applied to ensure the intersection of a retrieval path and a replication path. Later, a location-free double-ruling scheme is introduced in [7] based on boundary recognition and the computation of the respective gradient fields. To improve the flexibility of retrieval, a spherical projection-based double-ruling scheme

The authors are with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA.

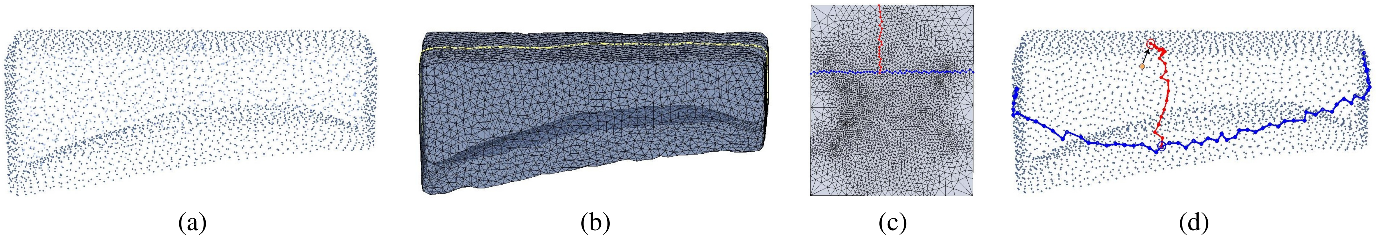


Fig. 2. (a) The model of a 3D sensor network deployed under water. (b) A computed cut graph marked with yellow. (c) The boundary surface is cut open to a topological disk along the cut graph and mapped to an aligned planar rectangle. The horizontal line marked with blue is one data replication curve. The vertical line marked with red is one data retrieval curve. (d) A data query.

is proposed in [8], where a planar network is mapped to a sphere based on stereographic projection. Both the replication and retrieval curves are great circles such that a retrieval curve always intersects all other replication circles.

B. Our Approach in 3D

Although double-ruling has shown highly effective for distributed data storage and retrieval in 2D sensor networks, it is extremely challenging to design in 3D sensor networks due to the constraint on the shape of a sensor network.

Figure 1 (b) shows a naive design of double-ruling based scheme in 3D sensor networks. In such a 3D grid sensor network, data replication and retrieval are along the horizontal and vertical planes respectively, such that a retrieval plane intersects all replication planes. Besides an extremely high cost of data replication, such 3D grid-based double-ruling scheme requires a network with regular cube shape and uniform node distribution.

On the contrary, the proposed approach does not require a 3D sensor network with a regular cube shape or uniform node distribution. Our approach is based on a topology concept that any closed surface can be cut open to a topological disk along an appropriate set of edges called a cut graph of the surface [9]. One example is a 3D sensor network deployed under water (see Figure 2 (a)). Its boundary nodes are identified and a triangular structure is extracted. A cut graph of the boundary surface is computed and marked with yellow color (see Figure 2 (b)). The boundary surface is cut open to a topological disk along the cut graph and then mapped to an aligned planar rectangle such that each boundary node is assigned a planar rectangle virtual coordinates (see Figure 2 (c)). Each non-boundary sensor stores the ID of its neighbor nearest to the boundary. A data generator follows the sequence of IDs to the boundary, and then travels along a horizontal line of the virtual planar rectangle and leaves data copies. The horizontal line marked with blue shown in Figure 2 (c) corresponds to the data replication curve shown in Figure 2 (d) marked with the same color. similarly, a query follows the sequence of IDs to the boundary and collects the aggregated data of different types along a vertical line. The vertical line marked with red shown in Figure 2 (c) corresponds to the data retrieval curve shown in Figure 2 (d) marked with the same color.

Without the knowledge of the geographic location and the distance bound, the success of data retrieval is always guaranteed because a pair of horizontal and vertical lines surely intersect. Retrieval of aggregated data through time and space with different types is also guaranteed. A query travels along one simple curve and then collects all desired information in the network because the retrieval curve intersects all replications curves of the network.

II. Cut Graph and Planar Rectangle Virtual Coordinates

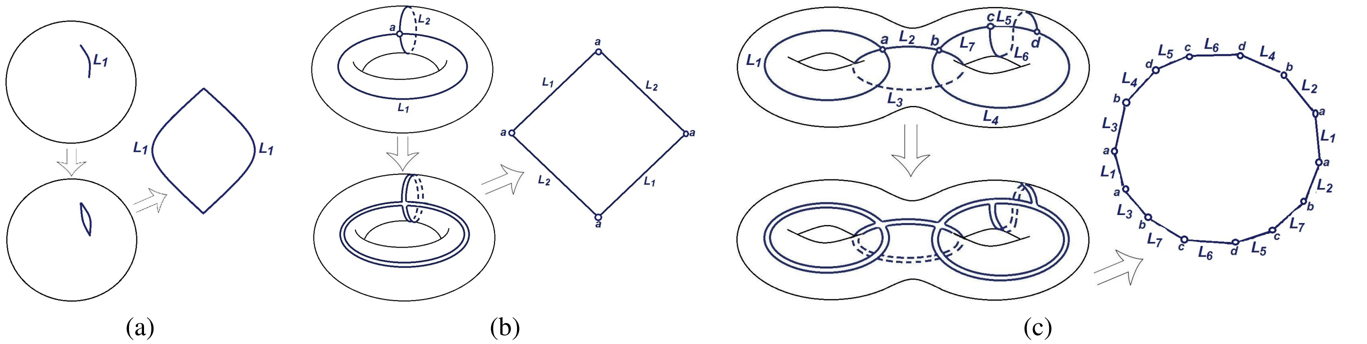
Given a sensor network deployed in 3D, we apply our previous algorithm [10] to detect its boundary vertices and then extract a triangulation of the boundary surface [11]. Note that we only need a connected triangulation to approximate the boundary surface, so we allow some mistakenly detected non-boundary vertices on the triangulation.

A. Computing Cut Graph

Any closed surface (e.g., a surface without a boundary) can be opened into a topological disk D (e.g., a surface with one boundary) by cutting along an appropriate set of edges called cut graph. Denote G a cut graph of the surface. Each edge of G appears twice on the boundary of D. We can obtain the original surface by gluing together these corresponding boundary edges of D. Figure 3 shows cut graphs of a genus 0, a genus 1, and a genus 2 surfaces respectively. The three closed surfaces are cut open to topological disks along the given cut graphs.

Denote $M = (V, E, F)$ a triangulation of the boundary surface of a 3D sensor network, consisting of vertices V, edges E, and triangle faces F. Denote $v_i \in V$ a vertex with id i; $e_{ij} \in E$ an edge with two ending vertices v_i and v_j; $f_{ijk} \in F$ a triangle face with vertices v_i, v_j, and v_k. M is connected, orientable, and closed. A fully distributed algorithm can be applied to compute the cut graph of M.

The algorithm starts from one randomly chosen triangle f_{ijk} of M, which can be the one with the smallest node id. f_{ijk} marks itself and its three edges e_{ij}, e_{jk}, and e_{ki}. Each of the marked edges checks whether it is shared by two marked triangles. For example, edge e_{ij} finds its neighboring triangle f_{jil} unmarked. e_{ij} then removes mark from itself but adds mark on triangle f_{jil} and edges e_{il} and e_{lj}. Note that it is possible that e_{il} or e_{lj} may have been marked already. The propagation algorithm stops when all the triangles of M have been marked. Let all the marked edges be G, which form a cut graph of M.



(a) (b) (c)

Fig. 3. Any closed surface M can be cut open to a topological disk D along one cut graph of the surface G: (a) a sphere surface (genus 0) (b) a torus surface (genus 1) (c) an eight surface (genus 2). Notice that the edges on G appear twice on the boundary of D. D has one-to-one mapping with M except the boundary.

Note that the proposed algorithm computes cut graph of M with general topology. While the boundary surface of a 3D sensor network with handles, as shown in Figure 3(b) and (c) is not common. Most 3D sensor networks are topologically equivalent to a solid ball, i.e., their boundary surfaces are a topological sphere as shown in Figure 3(a). Since a cut graph of a boundary surface topologically equivalent to a sphere is simply a path, we can have a much simpler algorithm to compute the cut graph. We conduct a simple flooding on M to find a pair of boundary nodes with the longest shortest path (hops) on the surface. We then virtually cut the triangulated surface M open to a topological disk D along the shortest path between the pair of nodes.

B. Computing Planar Rectangle Virtual Coordinates

We apply discrete surface Ricci flow to compute planar rectangle virtual coordinates of D. We refer readers to our previous work [12] for discrete surface Ricci flow. Here we only provide the implementation of the algorithm in detail.

Denote l_{ij} the length of e_{ij}. Denote θ_i^{jk} the corner angle attached to v_i belonging to f_{ijk}. Denote $\overline{K}_i$ and K_i the target and current Gaussian curvatures at v_i respectively. We uniformly pick four vertices along the boundary of D and then assign their target Gaussian curvatures: $\overline{k_i} = \frac{\pi}{2}$. For all other vertices of D, we assign: $\overline{k_i} = 0$.

1) For each v_i, we initiate a circle with radius $\gamma_i = 1$. For each edge e_{ij}, the two circles at v_i and v_j intersect with an angle $\phi_{ij} = \frac{\pi}{2}$. Let u_i be the logarithm of γ_i for each v_i. Then $u_i = 0$ in the initialization.
2) For each e_{ij}: $l_{ij} = e^{u_i} + e^{u_j}$.
3) Each $\theta_i^{jk} = cos^{-1}\frac{l_{ij}^2+l_{ki}^2-l_{jk}^2}{2l_{ij}l_{ki}}$.
4) For each v_i: $K_i = 2\pi - \sum_{f_{ijk}\in F}\theta_i^{jk}$.
5) Denote ε a threshold. If all $|\overline{k_i} - k_i| < \varepsilon$, the algorithm goes to the next step; otherwise, $u_i = u_i + \delta(\overline{k_i} - k_i)$, where δ is the step length, a small constant. The algorithm goes back to step 2.
6) Isometric embedding: Denote p_i the planar coordinates of each v_i. Start from a boundary edge e_{ij} with v_i one of the four chosen boundary vertices: we assign $p_i = (0,0)$, $p_j = (l_{ij}, 0)$. In a breadth first search way, if f_{ijk} has exactly two vertices (e.g., v_i and v_j) with planar coordinates (e.g., p_i and p_j), compute p_k as one intersection point of two circles centered at p_i and p_j with radii l_{ik} and l_{jk} respectively, and satisfying $(p_k - p_i)\times(p_j - p_k) > 0$ [1]. Repeat the above process until every vertex has its planar coordinates. The planar rectangle is automatically aligned with x-axis.

Note that the algorithm is fully distributed and gossip-style. Each vertex v_i only needs to exchange its u_i with its one range neighbors at one iteration. The convergence of the algorithm is proved in [13]. The number of iterations is determined by $-C\frac{\log\varepsilon}{\lambda}$ where C is a constant, ε is the threshold of curvature error, and λ is the step length of each iteration [13]. We set $\varepsilon = 1e-4$ and $\lambda = 0.1$ in our implementation.

III. Implementation

A. Data Replication

Since we assume location free of a given 3D sensor network, we let each non-boundary node store the ID of its neighbor nearest to the boundary of the network. A datum follows a sequence of nodes to the nearest boundary node denoted as p. Assume a data replication curve is along a horizontal line. Since the boundary surface of the network has been mapped to a virtual planar rectangle, the horizontal line through p is unique, solely determined by the y coordinate of the planar rectangle virtual coordinates of p. The datum leaves pointers or copies at nodes along the line with two directions - one with the increased and the other with the decreased x coordinate. At each step, the datum simply checks the planar rectangle virtual coordinates of its one range neighbors and chooses the one with the closest distance to the line and along the current direction. Once finishing data replication, the datum turns back and follows the reversed path back.

B. Data Retrieval

Without the aware of the knowledge of the requested data's location and distance, a query follows a sequence of nodes

[1] The direction of the cross product of the two planar vectors points outside instead of inside.

TABLE I
COMPARISON OF AVERAGE PRODUCER AND CONSUMER COSTS OF SINGLE TYPE OF DATA

	Seabed		
	Cut Graph	GHT	SR-GHT
Producer cost	69.3331	22.5738	19.4178
Consumer cost	19.5381	22.8416	94.4715

to the nearest boundary node denoted as p. Assume a data retrieval curve is along a vertical line. A vertical line passing through p is determined solely by the x coordinate of the planar rectangle virtual coordinates of p. The query simply travels along the line with two directions - one with the increased and the other with the decreased y coordinate. At each step, similarly, the query simply checks the planar rectangle virtual coordinates of its one range neighbors and chooses the one with the closest distance to the line and along the current direction. The query either stops as soon as it hits the replication curve of its desired data, or travels along a full vertical line to collect all the aggregated data in the network. Once data has been collected, the query turns back and follows the reversed path back.

C. Delivery of Data and Query

As a preprocessing, each of the boundary nodes sends messages recording its minimum hop count to boundary (initialized to zero) to its neighbors. A non-boundary node receives a message and compares with its current record (initialized to infinity). If the received count has more than one hop count less, the node updates its current one and records the ID of its neighbor sending this message. The node also updates the count of the message and then sends to its neighbors. Otherwise, the node simply discards the message. When there is no message in the network, each of the non-boundary nodes of the network has recorded the ID of its neighbor nearest to boundary. It is then straightforward for a datum or a query to travel along the shortest path to the boundary according to the sequences of IDs.

D. Storage

We have very limited information stored at the nodes of the network. A non-boundary node stores the ID of its neighbor nearest to boundary, and a boundary node stores the computed planar rectangle virtual coordinates. For the data replication, we can leave copies of data on either all the nodes along the replication curve; or just a small portion of nodes sampled along the replication curve. It is a trade off between the storage cost and the retrieval cost.

IV. SIMULATIONS

We evaluate the performance of the proposed location-free cut graph based double-ruling scheme on 3D sensor network given in Figure 2. The network has 4369 number of nodes and the average number of neighbors of each node is 13.79. Data storage and retrieval costs are measured by the number of hop counts needed to store or retrieve data. Traffic load on each node is measured by the number of messages passing through it. In our simulation, each node has equal probability to be a datum or a query.

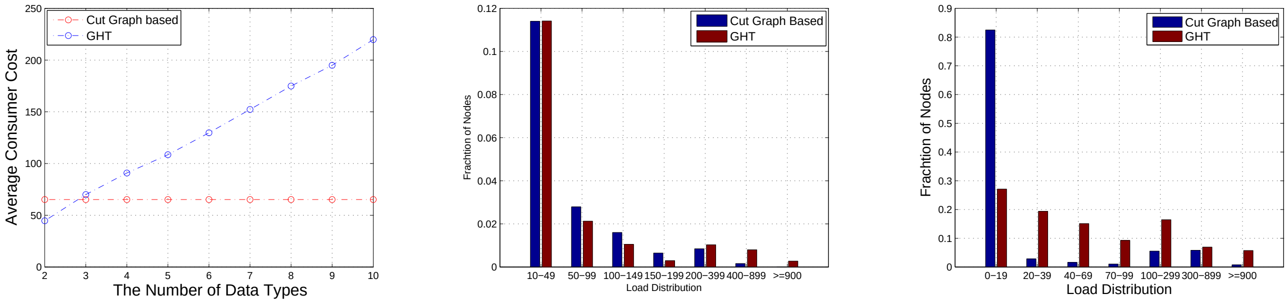
Note that there are very limited algorithms to compare with because all previous double-ruling based schemes work in 2D sensor networks and can't be applied in 3D. GHT based schemes can be more easily applied in 3D but require geographic information. Our implementation of the GHT based scheme in 3D for comparison has actually considered geographic information to design the hash function and stored heavy routing information on each node (shortest path tree rooted at each node) to guarantee the routing path a shortest one from a datum or a query to the hashed location, and hence all "improved GHT" approaches won't help to achieve better performance in our comparison.

A. Data Storage and Retrieval Costs

1) Single Type of Data: We compare cut graph based scheme with GHT scheme with and without structured replication. For GHT with structured replication (SR-GHT), we apply 1 level hierarchy with extra 3 mirror points scattered in network to store the nearby data. Table I lists the average data storage and retrieval costs with one type of data generated in network. For cut graph based scheme, the data storage cost is the highest and the retrieval cost is the lowest; a datum needs to travel and leave copies of data along the whole replication curve while a query can stop immediately when its retrieval curve intersects a data replication curve. For SR-GHT scheme, on the contrary, the data storage cost is the lowest and the retrieval cost is the highest; a datum can store data at the closest location, but a retrieval has to travel to both the hashed location and its three mirror points to collect data.

2) Aggregated Data: If there are more than one data type in network, as shown by Figure 4(a), the retrieval cost of cut graph based scheme is fixed; a query collects all different types of data by simply traveling along one retrieval line. While the retrieval cost of GHT scheme increases proportional to the number of data types; a query has to travel to different hashed locations for different types of data. Note that the cost of GHT scheme may decrease because we simply take a round trip to each hashed location in our implementation. But to find a minimum tour to visit all of the locations is the traveling salesman problem, which is NP-hard. The producer cost does not change for either cut graph based or GHT based schemes with the increase of data types.

Figure 4(a) clearly shows that cut graph based scheme performs the best for retrieval of multiple types of data generated in network. When there is only one type of data in network, cut graph based and GHT based schemes have a tradeoff between the data storage and retrieval costs. While with the increase of data replication, cut graph based scheme has a better fault tolerance and a more balanced load distribution across the network as discussed in Sec IV-B.



(a) (b) (c)

Fig. 4. (a) Comparison of average consumer costs with the increase of data types in the network. (b) Comparison of load distribution with one information producer and one data type in the network. (c) Comparison of load distribution with one hundred information producers and ten data types in the network.

B. Load Distribution

We simulate different scenarios to evaluate the load distribution of cut graph based approach and compare with GHT based one. The first scenario is a network stored with one data type. Each node in the network has equal probability to request for data. For both GHT and cut graph based approaches, the load on the majority of nodes (83% of nodes) are within a small number. Figure 4(b) shows the distribution of high traffic load on the remaining nodes. For GHT based approach, nodes near the hashed location suffer much higher traffic load; while for cut graph based approach, boundary nodes take a little bit more traffic load since the load has been evenly distributed among the boundary nodes. The node suffering the highest traffic has a load of 4368 with GHT based approach and 813 with cut graph based one.

The second scenario is a network stored with ten data types. We randomly choose 100 sensor nodes as datum generators from the network. Each node in the network has equal probability to request for aggregated data. Figure 4(c) shows the distribution of the total traffic load of data storage and retrieval. For GHT based approach, a query has to travel a long path to collect different types of data scattered in the network, which generates high traffic load; while for cut graph based approach, a query has fixed cost for aggregated data retrieval such that the majority of the traffic load of the network is still low. The node suffering the highest traffic has a load of 10211 with GHT based approach and 1542 with cut graph based one.

V. Conclusion and Future Works

We have presented a location-free cut graph based double-ruling scheme for large-scale 3D sensor networks. A data query simply travels along a simple curve with the guaranteed success to retrieve aggregated data through time and space with different types across the network. We have conducted simulations and comparisons that further show the proposed approach achieves low cost and a balanced traffic load. In current work, we have considered a network with a topological sphere shape and static sensors. As a future work, we will consider a dynamic network with more complicated topological shapes and possible mobile nodes. We will also design a local recovery scheme with the presence of nodes' failures and replacement.

References

[1] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "GHT: A geographic hash table for data-centric storage in sensornets," in *The 1st ACM Workshop on Wireless Sensor Networks ands Applications*, pp. 78–87, 2002.

[2] I. Stojmenovic and B. Vukojevic, "A routing strategy and quorum based location update scheme for ad hoc wireless networks," tech. rep., Technical Report TR-99-09, University of Ottawa, 1999.

[3] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *ACM MobiCom*, pp. 148–159, 2002.

[4] X. Liu, Q. Huang, and Y. Zhang, "Balancing push and pull for efficient information discovery in large-scale sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 241–251, 2007.

[5] D. Braginsky and D. Estrin, "Rumor routing algorthim for sensor networks," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 22–31, 2002.

[6] Q. Fang, J. Gao, and L. J. Guibas, "Landmark-based information storage and retrieval in sensor networks," in *IEEE INFOCOM*, pp. 1–12, 2006.

[7] S. Funke and I. Rauf, "Information brokerage via location-free double rulings," in *Proceedings of the 6th international conference on Ad-hoc, mobile and wireless networks*, pp. 87–100, 2007.

[8] R. Sarkar, X. Zhu, and J. Gao, "Double rulings for information brokerage in sensor networks," in *ACM MobiCom*, pp. 286–297, 2006.

[9] J. Munkres, *Topology (2nd Edition)*. Prentice Hall, 2000.

[10] H. Zhou, S. Xia, M. Jin, and H. Wu, "Localized algorithm for precise boundary detection in 3d wireless networks," in *IEEE ICDCS*, pp. 744–753, 2010.

[11] 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 *IEEE INFOCOM*, pp. 1053–1061, 2011.

[12] M. Jin, G. Rong, H. Wu, L. Shuai, and X. Guo, "Optimal surface deployment problem in wireless sensor networks," in *Proc. of the 31st Annual IEEE Conference on Computer Communications (INFOCOM'12)*, pp. 2345–2353, 2012.

[13] B. Chow and F. Luo, "Combinatorial Ricci Flows on Surfaces," *Journal Differential Geometry*, vol. 63, no. 1, pp. 97–129, 2003.