3D Surface Localization with Terrain Model

Yang Yang, Miao Jin, and Hongyi Wu

***Abstract*—The majority of current research on sensor network localization focuses on wireless sensor networks deployed on two dimensional (2D) plane or in three dimensional (3D) space, very few on 3D surface. However, many real world applications require large-scale sensor networks deployed on the surface of a complex 3D terrain. Compared with planar and 3D network localizations, surface network localization generates unique and fundamental hardness.**

In this research, we explore 3D surface network localization with terrain model. A digital terrain model (DTM), available to public with a variable resolution up to one meter, is a 3D representation of a terrain's surface. It is commonly built using remote sensing technology or from land surveying and can be easily converted to a triangular mesh. Given a sensor network deployed on the surface of a 3D terrain with one-hop distance information available, we can extract a triangular mesh from the connectivity graph of the network. The constraint that the sensors must be on the known 3D terrain's surface ensures that the triangular meshes of the network and the DTM of the terrain's surface approximate the same geometric shape and overlap. We propose a fully distributed algorithm to construct a well-aligned mapping between the two triangular meshes. Based on this mapping, each sensor node of the network can easily locate reference grid points from the DTM to calculate its own geographic location. We carry out extensive simulations under various scenarios to evaluate the overall performance of the proposed localization algorithm. We also discuss the possibility of 3D surface network localization with mere connectivity and the results are promising.

I. Introduction

A variety of applications in wireless sensor networks require geographic locations of sensor nodes. Instead of equipping each sensor node with a high cost localization hardware such as GPS receiver, different localization algorithms and protocols have been proposed that allow the sensor nodes to derive their own locations.

Current localization research focuses on sensor networks deployed on two-dimensional (2D) plane or in three-dimensional (3D) space [1]–[28]. They take distance information as input, and then search the solution space to find coordinates of sensor nodes that preserve the distance matrix as much as possible. Distance between adjacent sensors can be measured by received signal strength (RSS) or time difference of arrival (TDOA), or simply approximated by one-hop radio range. For remote sensors, their distance can be approximated by hop counts of the shortest path.

In real-world applications, many large-scale sensor networks are deployed over complex terrains, such as the volcano monitoring project [29] and ZebraNet [30]. Localization of a network deployed over a 3D surface generates a unique hardness compared with the well-studied localization of a network in 2D or 3D space. Specifically, due to limited radio range, the distance between two remote sensors deployed over a 3D surface can only be approximated by their surface distance, the length of the shortest path between them on the surface. Such surface distance is different from the 3D Euclidean distance of two nodes. As proved in [31], a localization algorithm doesn't exist for a network deployed over a 3D surface with surface distance information only, even if we assume accurate range distance measurement available. One intuitive example is that a piece of paper can be rolled to different shapes, but distance between any pair of points on the paper doesn't change. With pure surface distance information, we can never figure out the current shape of the paper. We can also learn the hardness of localization of a network deployed over a 3D surface from differential geometry. Consider that the distance information of a sensor network deployed over a 3D surface approximates the distance information of the surface, there exists no unique embedding in 3D within rigid motions for a general surface with distance only [32].

In [31], authors assume each sensor node can measure not only distances between its neighboring nodes but also its own height information. They require a sensor network is deployed on a surface with single-value property - any two points on the surface have different projections on plane. Such property ensures that they can project the network deployed over a 3D surface to 2D plane by removing z coordinate without ambiguity. They apply existing 2D network localization method on the projected one to compute the x and y coordinates of each sensor node, and then add the height information back as the z coordinate.

Later, a cut-and-sew algorithm is proposed in [33] to generalize the localization algorithm introduced in [31] from single-value surfaces to general surfaces. The algorithm takes a divide-and-conquer approach by partitioning a general 3D surface network into a minimal set of single-value patches. Each single-value patch can be localized individually, and then all single-value patches are merged into a unified coordinates system.

However, integrating height measurement into every sensor of a network is not always practical and affordable, especially for a large-scale sensor network. The motivation of this work is to explore the possibility of localization of a network deployed over surfaces with one-hop distance information only or even just mere connectivity, if we have the information of the deployed terrain surface.

The authors are with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA. Yang Yang and Miao Jin are supported by NSF CAREER Award CCF-1054996.

978-14799-3360-0/14/$31.00 ©2014 IEEE

A. Our Approach

A 3D representation of a terrain's surface is called a digital terrain model (DTM). DTMs are commonly built using remote sensing technology or from land surveying, and are available to public with a variable resolution up to one meter. For example, the Shuttle Radar Topography Mission (SRTM) [34] is a high-resolution digital topographic database that provides DTM data for North and South America with high accuracy and dense coverage. It is expected that acquisitions from radar satellites TerraSAR-X and Tan DEM-X will be available in 2014 to provide a uniform global coverage of DTMs up to 5 m absolute height accuracy at 10 m grid spacing [35].

A DTM is represented by a grid of squares, where the longitude, latitude, and altitude (i.e., 3D coordinates) of all grid points are known. It is straightforward to convert the grid into a triangulation, e.g., by simply connecting a diagonal of each square. Therefore a triangular mesh of the DTM of a terrain surface can be available before we deploy a sensor network on it. On the other hand, given a wireless sensor network deployed on a terrain surface with one-hop distance information available, a simple distributed algorithm can extract a refined triangular mesh from the network connectivity graph. Vertices of the triangular mesh are the set of sensor nodes. An edge between two neighboring vertices indicates the communication link between the two sensors. The constraint that the sensors must be on the known 3D terrain surface ensures that the triangular mesh of the DTM of the terrain surface overlaps with the triangular mesh extracted from the network connectivity graph. The question is how the latter can be localized in reference to the former.

The proposed approach is based on surface conformal structure. Conformal structure is an intrinsic geometric structure of surfaces, determined by surface distance. Conformal structure can tolerate a small local deformation of a surface, so the conformal structure of a surface is consistent even if the surface is approximated by different triangulations with various densities. Surfaces sharing the same conformal structure exist conformal mapping between them. A conformal mapping is a one-to-one and continuous mapping which preserves angles and local shape.

The triangular mesh of the DTM of a terrain surface and the triangular mesh extracted from the connectivity graph of a network deployed over the terrain surface approximate the geometric shape of the same terrain surface. Theoretically, the two triangular meshes share the same conformal structure. We can construct a well-aligned conformal mapping between them. Based on this mapping, each sensor node of the network can easily locate reference grid points of the DTM to calculate its own location.

Fig. 1 illustrates the basic idea. Fig. 1 (a) shows the triangular mesh of the DTM of a terrain surface. Fig. 1 (c) shows the triangular mesh extracted from the connectivity graph of a network deployed over the terrain surface. We first compute two conformal mappings, denoted as f_1 and f_2 respectively, to map the two triangular meshes to plane as shown in Figs. 1 (b) and (d) respectively. Such mapping exists based on Riemanns theorem that a topological disk surface can be mapped to plane through a conformal mapping [36]. However, the two mapped triangular meshes on plane are not aligned. Three anchor nodes marked with red as shown in Fig. 1 (c) are deployed with the network to provide the reference for alignment. Based on the positions of the three anchor nodes, We construct another conformal mapping, denoted as f_3, to align the mapped network triangular mesh with the mapped DTM triangular mesh on plane. Combining the three mappings, $f_1^{-1} \circ f_3 \circ f_2$, induces a well-aligned conformal mapping between the two triangular meshes shown in Fig. 1 (a) and (d) respectively. Based on the well-aligned mapping, each sensor node of the network, a vertex of the network triangular mesh, simply locates its nearest grid points, vertices of the DTM triangular mesh, to calculate its own geographic location. Note that the proposed localization algorithm, theoretically speaking, only requires three anchor nodes for a network with thousands or even tens of thousands of sensor nodes.

The rest of this paper is organized as follows: Sec. II-A introduces briefly the background knowledge necessary to the proposed surface network localization algorithm. Sec. III provides in detail the proposed distributed algorithm to localize a wireless sensor network deployed on a terrain surface. Sec. IV discusses some possible solutions if one-hop distance information or anchor nodes are not available. Sec. V presents simulation results. Sec. VI concludes the paper.

II. Theoretical Background

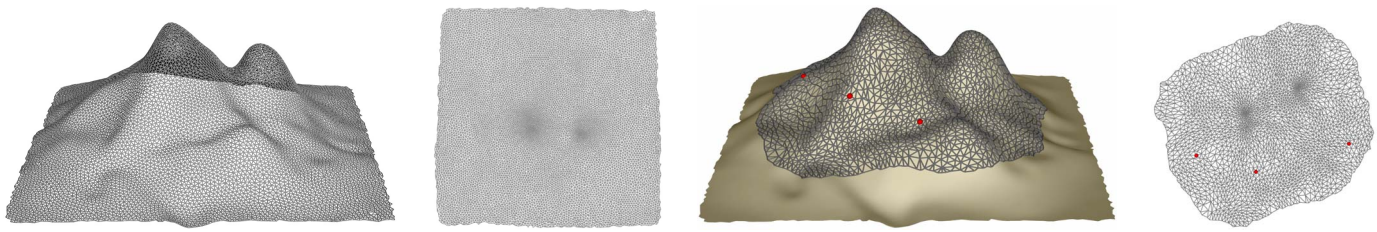
Before giving the details of the proposed surface network localization algorithm in Sec. III, we introduce briefly the background knowledge necessary to the algorithm. Specifically, we introduce the concept of discrete conformal mapping in Sec. II-A and discrete surface Ricci flow, a tool we apply to compute discrete conformal mapping of a triangular mesh from 3D to 2D plane in Sec. II-B. We then introduce M*ö*bius Transformation, a tool we apply to align two planar triangular meshes in Sec. II-C.

A. Discrete Conformal Mapping

Intuitively speaking, a conformal mapping is a one-to-one and continuous mapping that maps infinitesimal circles to infinitesimal circles and preserves the intersection angles among the infinitesimal circles.

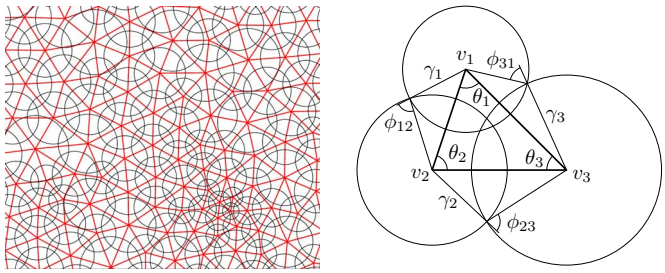
In discrete setting, we denote $M = (V, E, F)$ a connected triangular mesh embedded in $\mathbb{R}^3$, consisting of vertices (V), edges (E), and triangle faces (F). Specifically, we denote $v_i \in V$ a vertex with ID i; $e_{ij} \in E$ an edge with two ending vertices v_i and v_j; $f_{ijk} \in F$ a triangle face with vertices v_i, v_j, and v_k. A boundary edge is defined as an edge shared by one triangle face only. The two ending vertices of a boundary edge are defined as boundary vertices. A non-boundary edge is shared by two triangular faces.

If we use circles with finite radii to approximate infinitesimal circles, we can approximate conformal mapping in discrete setting. It is called circle packing metric, introduced by



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

Fig. 1. (a) The triangular mesh of the DTM of a terrain surface. (b) The triangular mesh of the DTM is conformally mapped to plane. (c) The triangular mesh extracted from the connectivity graph of a network deployed over the terrain surface. Three randomly deployed anchor nodes are marked with red. (d) The triangular mesh extracted from the network connectivity graph is conformally mapped to plane.



(a) (b)

Fig. 2. Circle Packing Metric. (a) Flat circle packing metric on a triangular mesh (b) Circle packing metric on a triangle.

Thurston in [37] as shown in Fig. 2. We assign each v_i a circle and denote γ_i its radius. The radius function is $\Gamma : V \to \mathbb{R}^+$. The two circles at v_i and v_j of edge e_{ij} intersect with an acute angle, denoted as ϕ_{ij} and called the *weight* on the edge. The edge weight function is then $\Phi : E \to [0, \frac{\pi}{2}]$.

Denote l_{ij} the edge length of e_{ij}. l_{ij} can be computed from the circle radii of the two ending vertices γ_i, γ_j and its weight ϕ_{ij} from the cosine law:

$$l_{ij}{}^2 = \gamma_i{}^2 + \gamma_j{}^2 + 2\gamma_i\gamma_j \cos\phi_{ij}. \tag{1}$$

Definition 1 (Circle Packing Metric): The pair of vertex radius function and edge weight function on a mesh M, (Γ, Φ), is called a circle packing metric of M.

From the definition of conformal mapping, a conformal mapping on a discrete surface with circle packing metric modifies the vertex radii, and preserves the edge weights.

B. Discrete Surface Ricci Flow

Richard Hamilton first introduced Ricci flow in his seminal work [38] in 1982. Chow and Luo proved a general existence and convergence theorem for discrete Ricci flow on surfaces in [39]. Jin et al. later provided a series of computational algorithms for discrete Ricci flow on surfaces in [40].

Before we introduce the definition of discrete surface Ricci flow, we need to give definitions of discrete metric and discrete Gaussian curvature first.

Definition 2 (Discrete Metric): A discrete metric on M is a function $l : E \to \mathbb{R}^+$ on the set of edges, assigning to each edge $e_{ij} \in E$ a positive number l_{ij} such that all triangles satisfy the triangle inequalities $f_{ijk} \in F$: $l_{ij} + l_{jk} > l_{ki}$.

Edge lengths of M satisfy the triangle inequalities, so they are sufficient to define a discrete metric on M.

Definition 3 (Discrete Gaussian Curvature): Denote θ_i^{jk} the corner angle attached to Vertex v_i in Face f_{ijk}, and ∂M the boundary of M, the discrete Gaussian curvature K_i on $v_i \in V$ is defined as the angle deficit at v_i:

$$K_i = \begin{cases} 2\pi - \sum_{f_{ijk} \in F} \theta_i^{ij}, & v_i \notin \partial M, \\ \pi - \sum_{f_{ijk} \in F} \theta_i^{jk}, & v_i \in \partial M. \end{cases} \tag{2}$$

Since we can compute corner angles directly from edge lengths, a discrete metric solely determines the discrete Gaussian curvature of M.

Definition 4 (Discrete Surface Ricci Flow): Let (Γ_0, Φ) be an initial circle packing metric of M. Denote $u_i = log\gamma_i$, and $\bar{K}_i$ and K_i the target and current Gaussian curvatures of v_i respectively, and t the time. The discrete surface Ricci flow is:

$$\frac{du_i(t)}{dt} = (\bar{K}_i - K_i(t)). \tag{3}$$

Discrete surface Ricci flow continuously deforms the circle packing metric of M according to the difference between the current and target Gaussian curvatures in a heat-like diffusion process, and converges when the difference is less than a threshold. The convergence of discrete surface Ricci flow is proved in [39]. The final circle packing metric induces a metric which satisfies the target Gaussian curvatures, and is conformal to the original surface metric.

C. Möbius Transformation

A complex number $z = a + bi$ defined on a complex plane can be simply considered as a point $p(a,b)$ on plane, where a and b are x and y coordinates of Point p respectively.

Definition 5 (Möbius Transformation): A Möbius transformation is a conformal map between complex plane to itself, represented as:

$$f(z) = \frac{az+b}{cz+d}, \tag{4}$$

where a, b, c, d are complex numbers, satisfying $ad - bc = 1$.

If a Möbius transformation maps four distinct complex numbers z_1, z_2, z_3, z_4 to four distinct complex numbers w_1, w_2, w_3, w_4 respectively, i.e., four distinct planar points are mapped to another four distinct planar points, the Möbius transformation keeps their cross-ratio invariant, represented as:

$$\frac{(z_1-z_3)(z_2-z_4)}{(z_2-z_3)(z_1-z_4)} = \frac{(w_1-w_3)(w_2-w_4)}{(w_2-w_3)(w_1-w_4)}. \quad (5)$$

Note that all operations in Eqn. 4 and 5 including addition, subtraction, multiplication, and division are all defined on complex numbers.

III. Surface Network Localization

Given a wireless sensor network deployed on a terrain surface, we apply the algorithm proposed in [41] to extract a refined triangular mesh from the connectivity graph of the network based on locally measured distances between nodes within one-hop communication range. Vertices of the triangular mesh are the set of sensor nodes. An edge between two neighboring vertices indicates the communication link between the two sensors. The algorithm is fully distributed and has no constraint on communication models.

Denote M_1 the triangular mesh of the DTM of the terrain surface and M_2 the triangular mesh extracted from the connectivity graph of the network. The proposed three-step localization algorithm is fully distributed. We explain each step in detail, specifically, conformal mapping of both M_1 and M_2 to plane in Sec. III-A, alignment of mapped M_1 and M_2 on plane in Sec. III-B, and localization of vertices of M_2 in Sec. III-C. Time complexity and communication cost are analyzed in Sec. III-D.

A. Conformal Mapping to Plane

Given a triangular mesh $M=(V,E,F)$ embedded in $\mathbb{R}^3$ (as defined in Sec. II-A), we apply discrete surface Ricci flow defined in Eqn. 3 to conformally map M to plane. Denote the mapping $f: M \to D \in \mathbb{R}^2$. The mapping result is stored at each v_i as a complex number (i.e., $z=x+yi$), and (x,y) serves as the planar coordinates of v_i.

One fact to consider when designing the mapping algorithm is that the boundary shape of a large-scale sensor network can be random and complicated, and the mapping result should be independent of the boundary shape. So we apply discrete surface Ricci flow with the following free-boundary condition: we assign the target Gaussian curvatures of all non-boundary vertices to zero, and discrete surface Ricci flow only deforms the circle radii of non-boundary vertices. Discrete surface Ricci flow converges when the target Gaussian curvatures of non-boundary vertices equal to zero. i.e., flat. Note that boundary vertices are ending vertices of boundary edges. Boundary edges of M can be easily detected according to the definition that they are shared by only one triangle face.

We first construct an initial circle packing metric (Γ_0, Φ) of M based on edge lengths. Denote γ_i^{jk} the corner radius associated with corner angle θ_i^{jk}. Each v_i computes its corner radii as:

$$\gamma_i^{jk} = \frac{l_{ki}+l_{ij}-l_{jk}}{2},$$

where l_{ij}, l_{jk}, l_{ki} represent the distance measurements of edges e_{ij}, e_{jk}, e_{ki}, respectively. Then v_i computes its initial circle radius γ_i by averaging its corner radii:

$$\gamma_i = \frac{1}{m}\sum_{f_{ijk}\in F} \gamma_i^{jk},$$

where m is the number of the adjacent faces to v_i (i.e., the vertex degree of v_i). For each edge e_{ij}, we compute its edge weight ϕ_{ij}, i.e., the intersection angle of the two circles centered at v_i and v_j with radii γ_i and γ_j respectively based on the Euclidean cosine law:

$$\cos\phi_{ij} = \frac{l_{ij}^2-\gamma_i^2-\gamma_j^2}{2\gamma_i\gamma_j}.$$

With the constructed initial circle packing metric, in each iteration of discrete surface Ricci flow, only non-boundary vertices update their circle radii. Specifically, each non-boundary v_i exchanges its current $u_i = log\gamma_i$ with its direct neighbors and updates its adjacent edge lengths $\{l_{ij} | e_{ij} \in E\}$ according to Eqn. 1. With the updated edge lengths, v_i computes its corner angles $\{\theta_i^{jk} | f_{ijk} \in F\}$ according to the inverse cos law:

$$\theta_i^{jk} = \cos^{-1}\frac{l_{ki}^2+l_{ij}^2-l_{jk}^2}{2l_{ki}l_{ij}}.$$

Then v_i computes its current discrete Gaussian curvature K_i as the excess of the total angle sum at v_i (Eqn. 2). If for every non-boundary v_i, the difference between its target Gaussian curvature $\bar{K}_i$ that is set to zero and current Gaussian curvature K_i is less than a threshold (we set to $1e-5$ in our experiments), discrete surface Ricci flow converges. Otherwise, each non-boundary v_i updates its u_i: $u_i = u_i + \delta(\bar{K}_i - K_i)$, where δ is the step length (we set to 0.05 in our experiments).

When discrete surface Ricci flow converges, we can stop the iterations. Each edge e_{ij} updates its length according to the final circle radii $\gamma_i = e^{u_i}$ and $\gamma_j = e^{u_j}$ and the stored edge weight ϕ_{ij}:

$$l_{ij} = \sqrt{\gamma_i{}^2 + \gamma_j{}^2 + 2\gamma_i\gamma_j\cos\phi_{ij}}.$$

With the computed edge lengths, we can embed M to plane. For simplicity, we let the vertex with the smallest ID (denoted as v_0) initiate the embedding process. Its planar coordinates are set to $(0,0)$. Then it arbitrarily selects one of its direct neighbors, e.g., v_j, and sets the planar coordinates of v_j to $(0, l_{ij})$. For vertex v_k, adjacent to both v_i and v_j, it calculates the intersection points of the two circles centered at v_i and v_j with radii l_{ik} and l_{jk}, respectively. Then, v_j chooses one of the intersection points that satisfies $(v_j - v_i) \times (v_k - v_i) > 0$ [1] as its planar coordinates. The procedure continues until all vertices of M have their planar coordinates.

Note that we can pre-compute the conformal mapping of M_1 to plane and then pre-load the mapping result to sensor nodes before the deployment of a network.

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

B. Alignment

Denote f_1 and f_2 the mappings that conformally map M_1 and M_2 to planar regions D_1 and D_2 respectively. We need to construct another conformal mapping that aligns D_2 with D_1 on plane.

Eqn. 5 provides a natural alignment of two planar regions based on three pairs of anchor points. Denote f a Möbius transformation that maps the planar region D_1 with three distinct points z_1, z_2, z_3 to the planar region D_2 with three distinct points w_1, w_2, w_3. Particularly, z_1, z_2, z_3 are mapped to w_1, w_2, w_3 respectively. We use complex numbers to represent points on plane. Assume we use z_{ij} to denote $z_i - z_j$, and w_{ij} to denote $w_i - w_j$, f can be represented in a closed form from Eqn. 5,

$$f(z) = \frac{w_2(z-z_1)z_{23}w_{12} - (z-z_2)z_{13}w_{23}w_1}{(z-z_1)z_{23}w_{12} - (z-z_2)z_{13}w_{23}}. \quad (6)$$

Again, all the operations in Eqn. 6 are defined on complex numbers.

Before we continue the alignment algorithm, we give a brief introduction of *Barycentric coordinates*. They provide a convenient way to interpolate a function on triangles as long as the function's value is known at vertices. Let's consider a function f defined on a triangle f_{abc} with $f(v_a)$, $f(v_b)$, and $f(v_c)$ known. Denote $Area|f_{abc}|$ the area of triangle f_{abc}. The function value of any point p located inside this triangle can be written as a weighted sum of the function value at the three vertices:

$$f(p) = t_1 f(v_a) + t_2 f(v_b) + t_3 f(v_c),$$

where $t_1 = \frac{Area|f_{pbc}|}{Area|f_{abc}|}$, $t_2 = \frac{Area|f_{pca}|}{Area|f_{abc}|}$, and $t_3 = \frac{Area|f_{pab}|}{Area|f_{abc}|}$. It is obvious that t_1, t_2, and t_3 are subject to the constraint $t_1 + t_2 + t_3 = 1$. t_1, t_2, and t_3 are called Barycentric Coordinates of Point p on f_{ijk}.

Assume three anchor nodes - sensor nodes equipped with GPS - are randomly deployed with other sensors. Each anchor node is assigned planar coordinates, e.g., mapped to plane by f_2. Denote the planar point of an anchor node mapped by f_2 with a complex numbers $z_i (1 \le i \le 3)$.

Each anchor node then checks its stored M_1 or simply sends a request with its known geographic position to a server to locate three nearest grid points of the DTM, denoted as v_i, v_j, and v_k. Since M_1 and M_2 are not perfectly overlap in general, the anchor node does not necessarily locate inside $f_{ijk} \in M_1$. We compute the projection point of the anchor node to f_{ijk}. The projection point is the closest point of M_1 to the anchor node. Since f_1 is a continuous and one-to-one mapping, we can compute the planar coordinates of the projection point mapped by f_1 based on the planar coordinates of v_i, v_j, and v_k. Specifically, denote (t_1, t_2, t_3) the Barycentric Coordinates of the projection point on f_{ijk}, $f_1(v_i)$, $f_1(v_j)$, and $f_1(v_k)$ the planar coordinates of v_i, v_j, and v_k mapped by f_1, the planar coordinates of the projection point mapped by f_1 is: $t_1 f_1(v_i) + t_2 f_1(v_j) + t_3 f_1(v_k)$. Denote the planar point of the projection point mapped by f_1 with a complex number $w_i (\ 1 \le i \le 3)$.

Each anchor node conducts a flooding to send out its z_i and w_i to the whole network. When receiving the three pairs of planar coordinates, a non-anchor node $v_i \in M_2$ simply plugs them and its planar coordinates by f_2 into Eqn. 6. The computed one is the aligned planar coordinates of the sensor node.

C. Localization

With the aligned planar coordinates, each sensor node locates three nearest grid points on plane. Denote v_i, v_j, and v_k the three nearest grid points on plane. The mapped planar point of the sensor node locates inside the planar triangle $f_{ijk} \in M_1$. Denote (t_i, t_j, t_k) the Barycentric Coordinates of the mapped planar point of the sensor node on f_{ijk}. The 3D geographic coordinates of the sensor node can be computed as $t_i p(v_i) + t_j p(v_j) + t_k p(v_k)$, where $p(v_i)$, $p(v_j)$, and $p(v_k)$ are the 3D geographic coordinates of v_j, v_k, and v_l respectively.

D. Time Complexity and Communication Cost

Assume we measure the communication cost by the number of exchanged messages. Both the time complexity and communication cost of the proposed localization algorithm are dominated by the step to compute conformal mapping of M_1 and M_2 to plane. The time complexity of discrete surface Ricci flow is measured by the number of iterations, given by $-C\frac{log\varepsilon}{\lambda}$, where C is a constant, ε is a threshold of curvature error, and λ is the step length of each iteration (we set to 0.05 in our experiments) [39]. Since each vertex only needs to exchange u values with its direct neighbors, the communication cost is given by $O(-C\frac{log\varepsilon}{\lambda}ng)$, where g is the average vertex degree of M, and n is the size of M. Note that g is six for a triangular mesh. The time complexity and communication cost of planar embedding based on computed edge lengths by discrete surface Ricci flow are linear to n.

The time complexity and communication cost of the other steps of the localization algorithm are either linear to the size of the network or constant complexity.

A special note is that we don't need to compute the conformal mapping of M_1 each time. We only need to compute it once before we start to deploy a network, and then pre-load only the mapping data related with the FoI (Field of Interest) to sensor nodes if they have sufficient storage. Otherwise, a server may be designated to keep the DTM database.

IV. Discussion

A. The Size of Anchor Nodes

Theoretically speaking, the proposed localization algorithm requires only three anchor nodes to align two triangular meshes on plane, even one triangular mesh is extracted from the connectivity graph of a network with thousands or even tens of thousands of sensor nodes. If there are more than three anchor nodes deployed with the network, we can apply the least-square conformal mapping method introduced in [42] instead of Möbius transformation to incorporate all anchor nodes into the alignment to improve the localization accuracy.

Fig. 3 shows one example. For a network with size 2.6k deployed on a 3D surface as shown in Fig. 1(d), the localization error of the network decreases with the increased number of anchor nodes. Compared with Möbius transformation based alignment introduced in Sec. III-B, least-square conformal mapping based alignment is more flexible to take anchor nodes into alignment. But from the other side, least-square conformal mapping method introduced in [42] is centralized with high computational complexity.

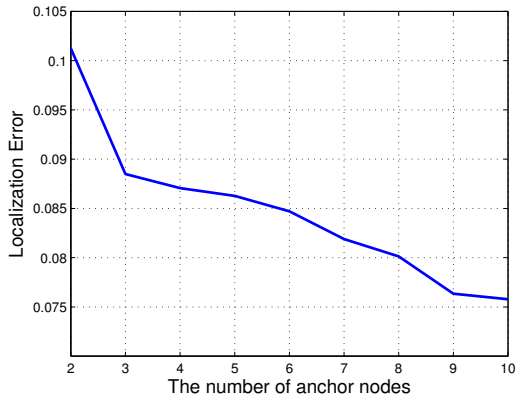


Fig. 3. Localization error decreases with the increased number of anchor nodes.

B. Anchor Node Free

As we introduced in Sec. II-A, conformal mapping maps infinitesimal circles to infinitesimal circles, so locally conformal mapping introduces no distortion, only scaling. Such scaling is called conformal factor. conformal factor at v_i can be approximated as the ratio of the triangle areas in $3D$ and mapped in $2D$ plane of all f_{ijk} incident to v_i,

$$cf(v_i) = \frac{\sum_{f_{ijk} \in F} Area_{3D}|f_{ijk}|}{\sum_{f_{ijk} \in F} Area_{2D}|f_{ijk}|}.$$

Conformal factor at the peak of a terrain surface is usually huge. We can demonstrate the fact by an extreme case. Suppose we have a long and open tube-shape can and we conformally map it to plane. The center of the bottom of the can is mapped to the origin. No matter what conformal mapping we construct, conformal factor increases exponentially fast as the mapped point on plane close to the origin.

Based on the fact, vertices of M_2, i.e., sensor nodes, with the highest conformal factors are around the peaks of a terrain surface. We can apply them as anchor nodes for alignment. Assume the network shown in Fig. 1(d) is anchor node free. We compute conformal factors of the triangular meshes of the DTM and the network and use colors to encode them at the mapped planar regions shown in Fig. 4. It is obvious that areas marked with red represent the regions with high conformal factors. We pick one vertex with the highest conformal factor for each red marked region. Suppose we pick v_1 and v_2 for the triangular mesh of the network, v_3 and v_4 for the triangular mesh of the DTM. Suppose v_1 shares a similar conformal factor with V_3. v_1 simply determines its 3D coordinates the same as v_3. Similarly, v_2 determines its 3D coordinates the same as v_4.

Note that if the shape of a mountain region is extremely complicated, conformal factors may identify wrong pairs of nodes between M_1 and M_2. The anchor free localization method is not stable in that case.

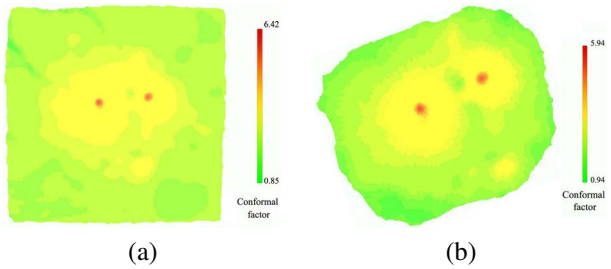


Fig. 4. We use colors to encode conformal factors of mapped triangular meshes on plane: (a) the mapped triangular mesh of the terrain surface shown in Fig. 1(b). (b) the mapped triangular mesh of the network shown in Fig. 1(d).

C. Connectivity Only

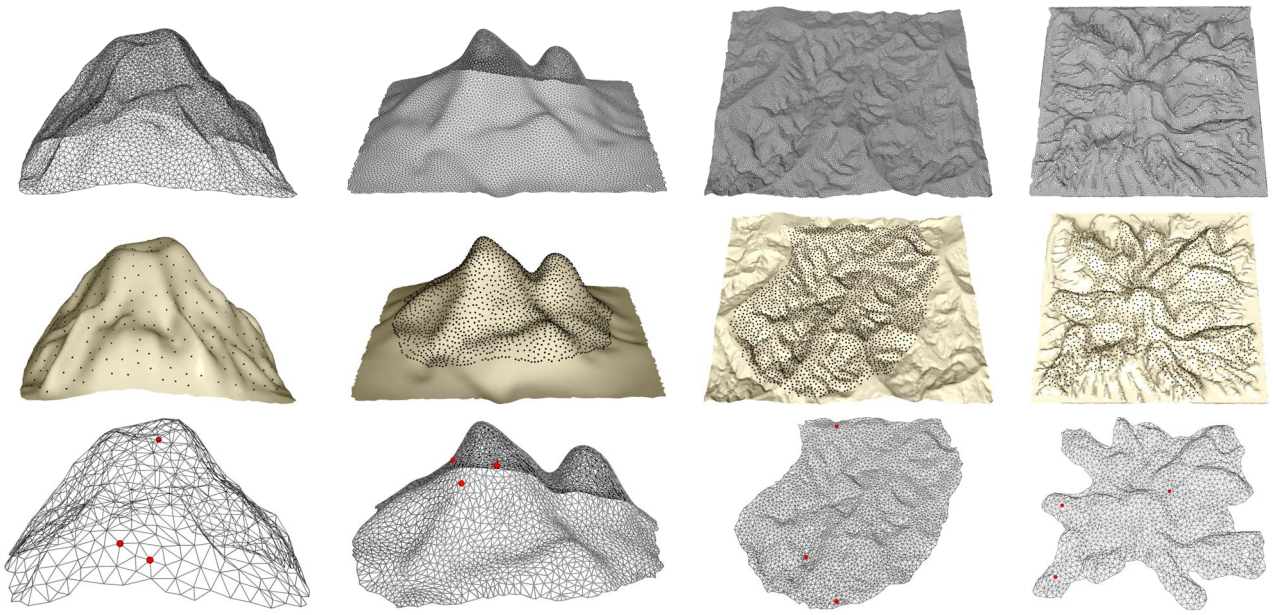
When range distance measurement is not available, we can still extract a sparse triangular mesh from a network connectivity graph. A simple landmark-based algorithm discussed in [43], [44] uniformly selects a subset of nodes in a distributed way and denotes them as landmarks, such that any two neighboring landmarks are approximately a fixed K hops away ($K \geq 6$). The dual of a discrete Voronoi diagram with generators the set of landmarks forms a triangulation. Vertices of the triangulation is the set of landmarks. Edge between two neighboring vertices is a shortest path between the two landmarks. We simply assume the edge length of the triangulation a unit one, and then apply exactly the same localization algorithm for landmark nodes as discussed in Sec. III.

A non-landmark node, denoted as n_i, finds its three nearest landmarks, denoted as v_1, v_2, v_3 with computed 3D coordinates $p(v_1)$,$p(v_2)$, and $p(v_3)$ respectively. Denote d_1, d_2, and d_3 the shortest distances (hop counts) of node n_i to the three landmarks v_1, v_2, v_3 respectively. Then node n_i computes its 3D coordinates $p(n_i)$ simply by minimizing the mean square error among the distances:

$$\sum_{j=1}^{3} (|p(n_i) - p(v_j)| - d_j)^2. \quad (7)$$

D. Network Density

The algorithm in [41] to extract a triangular mesh from the connectivity graph of a network assumes that a triangular graph is a sub-graph of the connectivity graph of the network. Such assumption is true only when the node density of the network is not too low. In our simulations, the average node degree of the connectivity graph of a network is around or above 8.



(a) DTM I (b) DTM II (c) DTM III (d) DTM IV

Fig. 5. The first row shows a set of DTMs of representative terrain surfaces. The second row shows wireless sensor networks marked with black points randomly deployed on these terrain surfaces. The third row shows the localized sensor networks with anchor nodes marked with red. The given set of anchor nodes provides each network an median localization error of the repeated tests.

V. Simulations

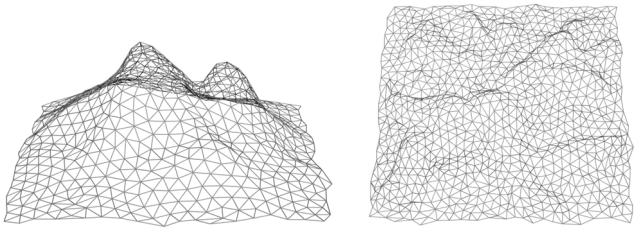
We pick a set of representative terrain surfaces and their corresponding DTMs as shown in Fig. 5, on which wireless sensor networks are randomly deployed (see the black points on these terrain surfaces). Both the terrain surfaces and the networks are not necessarily convex shape. The sizes of the sensor networks deployed on DTM I, II, III, and IV, are 0.5k, 2.6k, 3k, and 2k respectively. We carry out extensive simulations under various scenarios to evaluate the overall performance of the proposed algorithm with different factors such as the positions of the three anchor nodes, the resolution of a DTM, and the one-hop distance measurement error. We compute the localization error as the ratio of the average node distance error (all sensors in the network) and the average transmission range. Note that there is no simple alternative to localize a surface network as we discussed in Sec. I, so there is no comparison made.

A. Deployment of Anchor Nodes

We assume sensor nodes with accurate one-hop distance measurement and DTMs with high resolutions. For each network, we randomly deploy three anchor nodes and calculate the localization errors of the network based on the proposed algorithm in Sec. III. We repeat eight times for each network. Denote x_i the i^{th} localization error. We compute the arithmetic mean $\mu = \frac{1}{8}\sum_{i=1}^{8} x_i$ and the standard deviation $\sigma = \sqrt{\frac{1}{7}\sum_{i=1}^{8}(x_i-\mu)^2}$. Table I shows the mean, the median ($\tilde{x}$), and the standard deviation of localization errors under different sets of anchor nodes. The positions of the three anchor nodes affect the performance of the proposed localization algorithm slightly. In general, the more scattered we deploy the three anchor nodes in a network, the lower the localization error is.

TABLE I
The distribution of localization errors under different sets of anchor nodes

		DTM I	DTM II	DTM III	DTM IV
Error	μ	0.2579	0.1356	0.0951	0.2098
	$\tilde{x}$	0.2306	0.1343	0.0956	0.1512
	σ	0.1089	0.1717	0.0158	0.0352



(a) DTM II (b) DTM IV

Fig. 6. The same set of DTMs as shown in Fig. 5 with very low resolutions.

B. Terrain Models with Different Resolutions

To evaluate the impact of the resolution of a DTM, we compute the localization errors of a network deployed on a terrain surface with four different resolutions of the DTM. The resolution of the highest one is almost twenty times of the resolution of the lowest one. Fig. 5(b) and (d) show the two

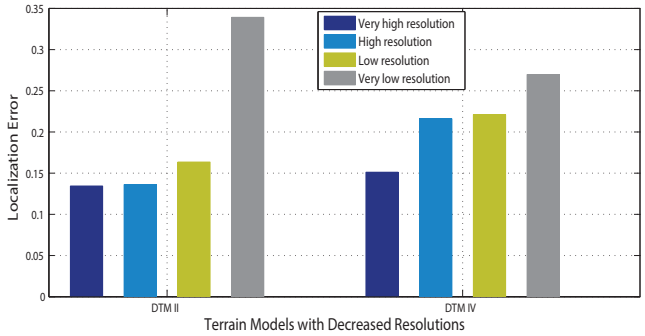


Fig. 7. The resolution of a DTM has a small impact on the performance of the localization algorithm.

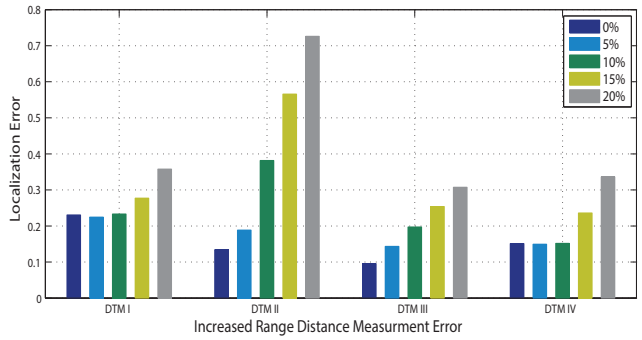
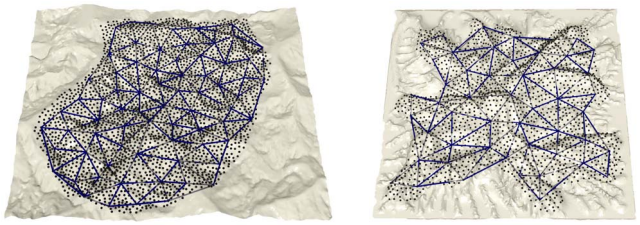


Fig. 8. The proposed localization algorithm is sensitive to distance measurement error.

DTMs of our testing with very high resolutions, and Fig. 6(a) and (b) show the same two DTMs with very low resolutions. The results given in Fig. 7 show that the resolution of a DTM has a small impact on the performance of the localization algorithm unless it is too low. Note that for each network, we choose the set of anchor nodes that gives an median localization error based on the repeated tests in Sec. V-A

C. Distance Measurement Error

We have also evaluated our algorithm when the one-hop distance information exists measurement error. For each network, we choose the set of anchor nodes that gives the median localization error based on the repeated tests in Sec. V-A. Fig. 8 shows that the proposed localization algorithm is sensitive to measurement error. A possible solution for a network with potentially large measurement errors is that we select uniformly a set of landmark nodes such that each landmark node has one hop distance to its landmark neighbors, i.e., a Voronoi diagram with a small and constant cell size, and then build a triangular mesh from the chosen landmark nodes with edge length approximately the averaged transmission range. Similar as connectivity based surface localization discussed in Sec. IV-C, we localize the landmark nodes first and then other non-landmark nodes.



(c) DTM III (d) DTM IV

Fig. 9. Networks with Connectivity Information Only.

D. Networks with Connectivity Information Only

As we discussed in Sec. IV-C, we uniformly select a subset of nodes marked as landmark nodes and build a sparse triangulation for a network with mere connectivity. Each vertex is a landmark node and each edge has an approximately constant length. Fig. 9 shows the sparse triangular meshes generated from the network with size $3k$ deployed on DTM III and the network with size $2k$ deployed on DTM IV. The localization errors for landmark nodes of the two networks are 0.2037 and 0.2610 respectively.

E. The Convergence Time

We carry out experiments to test the number of iterations of discrete surface Ricci flow required for convergence. Fig. 10 shows the convergence rates of discrete surface Ricci flow on network with size $3k$ deployed on DTM III and network with size $2k$ deployed on DTM IV. As we can pre-compute the planar conformal mapping of the DTM triangulation of a terrain surface, we can apply Newtons numerical method to compute the solution of discrete surface Ricci flow. The computation of this centralized method is very efficient with less than ten iterations in a few seconds for a triangulation with $10k$ size.

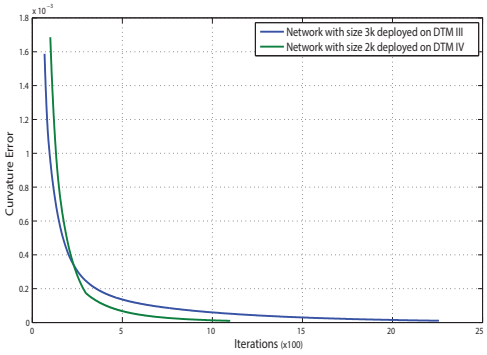


Fig. 10. The convergence rate of the discrete surface Ricci flow algorithm.

VI. Conclusion and Future Works

In this paper we have proposed a fully distributed algorithm to localize a wireless sensor network deployed on the surface of a complex 3D terrain. The algorithm constructs a well-aligned mapping between the triangular mesh of the DTM

of the terrain surface and the triangular mesh extracted from the connectivity graph of the network deployed on the terrain surface. Based on the mapping, each sensor node of the network can easily locate reference grid points from the DTM to calculate its own geographic location. We have carried out extensive simulations under various scenarios to evaluate the overall performance of the proposed algorithm with different factors. We have also discussed the possibility of 3D surface network localization with mere connectivity only.

As a future work, we will explore the best strategy to deploy the three anchor nodes. We will incorporate those useful contour features of surfaces like the peaks of valleys into the alignment algorithm. We will extend the proposed localization algorithm to be efficient in mobile environments.

REFERENCES

[1] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization For Very Small Devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, 2000.

[2] A. Savvides, C. Han, and M. B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in *Proc. of MobiCom*, pp. 166–179, 2001.

[3] J. Albowicz, A. Chen, and L. Zhang, "Recursive Position Estimation in Sensor Networks," in *Proc. of ICNP*, pp. 35–41, 2001.

[4] L. Doherty, L. Ghaoui, and K. Pister, "Convex Position Estimation in Wireless Sensor Networks," in *Proc. of INFOCOM*, pp. 1655–1663, 2001.

[5] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free Localization Schemes in Large Scale Sensor Networks," in *Proc. of MobiCom*, pp. 81–95, 2003.

[6] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," in *Proc. of GLOBECOM*, pp. 2926–2931, 2001.

[7] C. Savarese and J. Rabaey, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," in *Proc. of USENIX Annual Technical Conference*, pp. 317–327, 2002.

[8] A. Nasipuri and K. Li, "A Directionality Based Location Discovery Scheme for Wireless Sensor Networks," in *Proc. of ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 105–111, 2002.

[9] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AOA," in *Proc. of INFOCOM*, pp. 1734–1743, 2003.

[10] H. S. AbdelSalam and S. Olariu, "Passive Localization Using Rotating Anchor Pairs in Wireless Sensor Networks," in *Proc. of The 2nd ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking*, pp. 67–76, 2009.

[11] Z. Zhong and T. He, "MSP: Multi-Sequence Positioning of Wireless Sensor Nodes," in *Proc. of SenSys*, pp. 15–28, 2007.

[12] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur, "A Theory of Network Localization," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1663–1678, 2006.

[13] K. Yedavalli and B. Krishnamachari, "Sequence-Based Localization in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 81–94, 2008.

[14] S. Capkun, M. Hamdi, and J. Hubaux, "GPS-Free Positioning in Mobile Ad-Hoc Networks," in *Proc. of The 34th Annual Hawaii International Conference on System Sciences*, pp. 3481–3490, 2001.

[15] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from Mere Connectivity," in *Proc. of MobiHoc*, pp. 201–212, 2003.

[16] Y. Shang and W. Ruml, "Improved MDS-based Localization," in *Proc. of INFOCOM*, pp. 2640–2651, 2004.

[17] V. Vivekanandan and V. W. S. Wong, "Ordinal MDS-based Localization for Wireless Sensor Networks," *International Journal of Sensor Networks*, vol. 1, no. 3/4, pp. 169–178, 2006.

[18] G. Giorgetti, S. Gupta, and G. Manes, "Wireless Localization Using Self-Organizing Maps," in *Proc. of IPSN*, pp. 293 – 302, 2007.

[19] L. Li and T. Kunz, "Localization Applying An Efficient Neural Network Mapping," in *Proc. of The 1st International Conference on Autonomic Computing and Communication Systems*, pp. 1–9, 2007.

[20] H. Wu, C. Wang, and N.-F. Tzeng, "Novel Self-Configurable Positioning Technique for Multi-hop Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 609–621, 2005.

[21] M. Jin, S. Xia, H. Wu, and X. Gu, "Scalable and Fully Distributed Localization With Mere Connectivity," in *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, pp. 3164–3172, 2011.

[22] J. Wang, M. Tian, T. Zhao, and W. Yan, "A GPS-Free Wireless Mesh Network Localization Approach," in *Proc. of International Conference on Communications and Mobile Computing*, pp. 444–453, 2009.

[23] R. Magnani and K. K. Leung, "Self-Organized, Scalable GPS-Free Localization of Wireless Sensors," in *Proc. of WCNC*, pp. 3798–3803, 2008.

[24] H. Akcan, V. Kriakov, H. Bronnimann, and A. Delis, "GPS-Free Node Localization in Mobile Wireless Sensor Networks," in *Proc. of The 5th ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pp. 35–42, 2006.

[25] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proc. of IPSN*, pp. 46–54, 2004.

[26] A. M.-C. So and Y. Ye, "Theory of semidefinite programming for sensor network localization," in *Proc. of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 405–414, 2005.

[27] Y. Wang, S. Lederer, and J. Gao, "Connectivity-based sensor network localization with incremental delaunay refinement method," in *Proc. of INFOCOM*, pp. 2401–2409, 2009.

[28] C.-H. Ou and K.-F. Ssu, "Sensor position determination with flying anchors in three-dimensional wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 9, pp. 1084–1097, 2008.

[29] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," in *Proceedings of the 7th symposium on Operating systems design and implementation*, OSDI '06, pp. 381–396, 2006.

[30] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," *SIGARCH Comput. Archit. News*, vol. 30, no. 5, pp. 96–107, 2002.

[31] Y. Zhao, H. Wu, M. Jin, and S. Xia, "Localization in 3d surface sensor networks: Challenges and solutions," in *Proc. of the 31st Annual IEEE Conference on Computer Communications (INFOCOM'12)*, pp. 55–63, 2012.

[32] A. Pressley, *Elementary Differential Geometry*. Springer, 2010.

[33] Y. Zhao, H. Wu, M. Jin, Y. Yang, H. Zhou, and S. Xia, "Cut-and-sew: A distributed autonomous localization algorithm for 3d surface wireless sensor networks," in *Proc. of the 14th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'13)*, 2013.

[34] Shuttle Radar Topography Mission (SRTM). http://www2.jpl.nasa.gov/srtm/.

[35] TerraSAR-X. http://www.astrium-geo.com/terrasar-x/.

[36] R. Remmert, *Classical topics in complex function theory*. Springer-Verlag, 1998.

[37] W. P. Thurston, *Geometry and Topology of Three-Manifolds*. Princeton lecture notes, 1976.

[38] R. S. Hamilton, "Three manifolds with positive Ricci curvature," *Journal of Differential Geometry.*, vol. 17, pp. 255–306, 1982.

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

[40] M. Jin, J. Kim, F. Luo, and X. Gu, "Discrete surface ricci flow," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 14, no. 5, pp. 1030–1043, 2008.

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

[42] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 362–371, 2002.

[43] S. Funke and N. Milosavljevic, "Guaranteed-delivery geographic routing under uncertain node locations," in *Proc. of INFOCOM*, pp. 1244–1252, 2007.

[44] R. Sarkar, X. Yin, J. Gao, F. Luo, and X. D. Gu, "Greedy routing with guaranteed delivery using ricci flows," in *Proc. of IPSN*, pp. 121–132, 2009.