Scalable and Fully Distributed Localization With Mere Connectivity

Miao Jin[1], Su Xia[1], Hongyi Wu[1], and Xianfeng Gu[2]
[1]The Center for Advanced Computer Studies, University of Louisiana at Lafayette.
{mjin,sxx1110,wu}@cacs.louisiana.edu
[2]Department of Computer Science, Stony Brook University. gu@cs.sunysb.edu

***Abstract*—This work proposes a novel connectivity-based localization algorithm, well suitable for large-scale sensor networks with complex shapes and non-uniform nodal distribution. In contrast to current state-of-art connectivity-based localization methods, the proposed algorithm is fully distributed, where each node only needs the information of its neighbors, without cumbersome partitioning and merging process. The algorithm is highly scalable, with limited error propagation and linear computation and communication cost with respect to the size of the network. Moreover, the algorithm is theoretically guaranteed and numerically stable. Extensive simulations and comparison with other methods under various representative network settings are carried out, showing superior performance of the proposed algorithm.**

I. Introduction

Geographic location information is imperative to a variety of applications in wireless sensor networks, ranging from position-aware sensing to distributed data storage and processing, geographic routing, and nodal deployment. While global navigation satellite systems (such as GPS) have been widely employed for localization, integrating a GPS receiver in every sensor of an entire large-scale sensor network is unrealistic. Moreover, some application scenarios prohibit the reception of satellite signals by part or all of the sensors, rendering it impossible to solely rely on global navigation systems.

Even for those ranging information based localization schemes, extra equipments installed to measure the distance or the angle between nodes, can also lead to a dramatically increase of network cost. To this end, many interesting approaches have been proposed for localization with mere connectivity information. Each node only knows what nodes are nearby under its local communication range, but not know how far away and what direction its neighbors are.

Previous localization methods with mere connectivity have mainly focused on dimension reduction of multidimensional data sets based on the input distance matrix, which is approximated by hop counts between each possible pair of nodes. The two major methods, multi-dimensional scaling (MDS) based [1]–[3] and neural network based [4], [5] achieve the highest localization accuracy and yield coordinates of sensor nodes that preserve the distance matrix between the data points of the input space and the output space (i.e., a 2D plane) as much as possible.

One of the major problems for MDS based methods is their low *scalability*. The time complexity for obtaining the distance matrix is $(O(n^3))$; Eigen decomposition is also with $(O(n^3))$ complexity. With the increase of network size n, the computational cost is prohibitive. Another issue is that they are inherently *centralized*. While for general wireless sensor networks, with the limited power and computation capability of each sensor node, distributed algorithm is highly preferred. Different algorithms have been proposed to overcome these disadvantages. One approach is to partition the network to many subnetworks, and compute the localization of each subnetwork, and then merge these subnetworks together. This method requires delicate strategies and great caution in the merging stage.

For neural network based methods, *stability* is their major problem due to the non-convex shape of their minimized energy. Although several approaches have been proposed to increase the possibility to escape from local minima of the minimized energy, the selection of initial values are still crucial for the final localization results [5].

A. Our Approach

We propose a novel localization algorithm which overcomes the major difficulties of conventional MDS and neural network based methods. We explain our intuitions in the smooth setting first, then formulate the problem and design the algorithm in the discrete setting. We treat dense network as a smooth surface. The distance among points are determined by the Riemannian metric of the surface. The ground truth is that the surface is flat, therefore the Gaussian curvature which measures how much the surface is non-flat, should equal to zero everywhere. The metric can be estimated by measuring the distances among the points. Due to the measurement error, the estimated metric is curved. Our goal is to distort the estimated metric to make it flat with minimal distortion involved.

The method to distort a curved Riemannian metric to a flat one is the Ricci flow method. Basically, the Riemannian metric at each point is scaled proportional to the curvature at that point, which makes the Gaussian curvature evolve according to a heat diffusion process. Eventually, the surface becomes flat. By controlling the boundary condition of the curvature flow, one can minimize the overall metric distortion introduced.

In practice, given a large-scale sensor field, a subset of nodes can be uniformly selected in a distributed way and denoted as landmarks, such that any two neighboring landmarks

are approximately a fixed K hops away. Then a triangular mesh structure can be constructed based on mere connectivity information with a simple distributed scheme (as discussed in [6], [7]), where vertices of the triangular mesh is the set of landmarks, an edge between two neighboring vertices is a shortest path between the two landmarks in the given sensor network. The extracted mesh structure approximates well the sensor network. Then the Riemannian metric of the discrete sensor network is represented by the edge lengths. The curvature at each vertex is approximated by the discrete angle deficit. Each vertex has its own scaling factor, which is proportional to its curvature. The edge length is scaled by the product of the scaling factors of its two ending vertices. This process could induce an infinite number of flat metrics which isometrically embed the whole network onto the plane with theoretical guarantee. But the key is which one of those flat metrics induces the isometrically planar embedding to achieve the minimal localization error. We prove that by controlling the scaling factors on the boundary vertices, one can obtain the exact flat metric with the least distortion from the estimated metric. So the key step of our algorithm is to compute the optimal flat metric. Based on the computed flat metric, the next step of isometric embedding is trivial.

In our algorithm, all the involved computations for each node only require information from its direct neighbors, therefore it is *fully distributed* without cumbersome cutting and merging process. The proposed method is *numerically stable*, free of the choice of initial values and local minima with theoretical guarantee. The computational cost and communication cost are both linear to the size of the network, so the method is *scalable*, suitable for large ad-hoc networks with thousands of highly resource-constrained sensor nodes (processor, memory, and power) which have limited communication range. Furthermore, if there is some measurement error occurred at one node, the impact to another node decreases dramatically in terms of the distance between them. This *limited error propagation* also contributes to the high scalability. The method is *general* to planar networks with complex shapes and topologies. These merits are demonstrated in our experimental results.

B. Related Works

With merely connection information available, three major techniques are employed in current state-of-the-art localization schemes: multi-dimensional scaling (MDS), neural networks, and graph rigidity test.

MDS is a non-linear dimension reduction and data projection technique that transforms distance matrix into a geometric embedding (e.g., a planar embedding for 2D sensor network localization). MDS-based localization is originally proposed in [1]. It constructs a proximity matrix based on the shortest path distance (approximated by hop counts) between all pairs of nodes in the network. The singular value decomposition (SVD) is employed to produce the coordinates matrix that minimizes the least square distance error. Finally, it retains the first 2 (or 3) largest eigenvalues and eigenvectors as 2D (or 3D) coordinates. Subsequent improvements on MDS are made by dividing the graph into patches to enable distributed calculation [2], [3]. In addition [8] proposes to apply SVD to the matrix based on a set of beacon nodes only and thus reduces complexity. A similar idea is adopted in [9], with the simplex method (instead of SVD) for error minimization.

The second method is based on neural networks [4], [5], where non-linear mapping techniques and neural network models such as self-organizing map (SOM) are employed for dimension reduction of multidimensional data sets, yielding coordinates of sensor nodes that preserve the distances (also approximated by hop counts) between the data points of the input space and the output space (i.e., a 2D plane) as much as possible.

The localization algorithms based on graph rigidity theory [10]–[12] aim to create a well-spread and fold-free graph that resembles the given network. They focus on finding a globally rigid graph which can be embedded without ambiguity in plane. While with mere globally rigid structure, like a topological disk triangulation in [10], there exist infinite number of flat metrics which induce different planar embedding as long as the total Gaussian curvatures satisfy the discrete Gauss-Bonnet Theorem, which will be introduced in Sec. II. A brut force way is applied to find one planar embedding of the extracted global structure, which in general can not be easily guaranteed. So compared with MDS and neural network-based approaches, the graph rigidity-based methods exhibit lower localization accuracy in general.

II. Flat Metric

To find the 'optimal' flat metric which induces isometrically planar embedding of the triangular mesh (the sensor network) with the minimal localization error, we need to introduce first the concepts of metric and Gaussian curvature (Sec. III-A), surface Ricci flow in discrete setting (Sec. II-B), and then the condition to find an optimal flat metric (Sec. II-C).

A. Discrete Metric and Gaussian Curvature

In discrete setting, we let $M = (V,E,F)$ to represent a triangular mesh (or mesh in short), consisting of vertices (V), edges (E), and triangle faces (F).

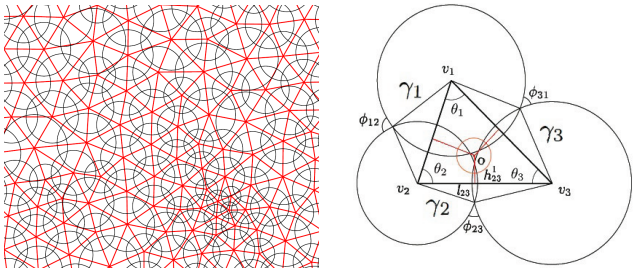
Definition 1 (Discrete Metric): A discrete metric on M is a function $l : E \to \mathbb{R}^+$ on the set of edges, assigning to each edge $e_{ij} \in E$ a positive number l_{ij} such that the triangle inequalities are satisfied for all triangles $t_{ijk} \in F$: $l_{ij} + l_{jk} > l_{ki}$.

If M is embedded in the Euclidean space $\mathbb{R}^3$, a discrete metric of M can be defined by its edge lengths.

Definition 2 (Discrete Gaussian Curvature): The discrete Gaussian curvature K_i on a vertex $v_i \in V$ is defined as the angle deficit:

$$K_i = \begin{cases} 2\pi - \sum_{f_{ijk} \in F} \theta_i^{ij}, & v_i \notin \partial M, \\ \pi - \sum_{f_{ijk} \in F} \theta_i^{jk}, & v_i \in \partial M, \end{cases} \tag{1}$$

where θ_i^{jk} represents the corner angle attached to Vertex v_i in Face f_{ijk} and ∂M is the boundary of the mesh.



(a) (b)

Fig. 1. Circle Packing Metric. (a) Flat circle packing metric on a triangular mesh (b) Circle packing metric on a triangle.

Since corner angles can be directly computed from edge lengths. It is obvious that the discrete Gaussian curvatures are determined by the discrete metric.

The following theorem says the total Gaussian curvature of M is solely determined by its topology:

Theorem 1 (Discrete Gauss-Bonnet Theorem): For a topological disk mesh $M = (V,E,F)$ with or without inner holes, the total Gaussian curvature of M is a topological invariant. It holds as follows:

$$\sum_{v_i \in V} K_i = 2\pi\chi(M), \tag{2}$$

where $\chi(M) = 2-b$, b is the number of boundaries.

B. Discrete Surface Ricci Flow

Ricci flow was first introduced by Richard Hamilton for Riemannian manifolds of any dimension in his seminal work [13] in 1982. Chow and Luo [14] proved a general existence and convergence theorem for the discrete Ricci flow on surfaces. Jin et al. provided a computational algorithm in [15]. In [7], Ricci flow was applied in wireless sensor network for greedy routing.

To briefly introduce the concept of discrete surface Ricci flow, we start from the concept of the circle packing metric which was introduced by Thurston in [16] as shown in Fig. 1. Each vertex v_i is assigned a circle with radius γ_i. The radius function is denoted as $\Gamma : V \to \mathbb{R}^+$. The two circles at the ending vertices v_i and v_j of edge e_{ij} intersect with an acute angle ϕ_{ij}, which is called the *weight* on the edge. The edge weight function is denoted as $\Phi : E \to [0, \frac{\pi}{2}]$.

The length of an edge e_{ij} can be computed from the vertex circle radii γ_i, γ_j and the weight ϕ_{ij} by the cosine law:

$${l_{e_{ij}}}^2 = {\gamma_i}^2 + {\gamma_j}^2 + 2\gamma_i\gamma_j\cos\phi_{ij}. \tag{3}$$

Definition 3 (Circle Packing Metric): A circle packing metric of a mesh M includes the circle radius function and the edge weight function.

Suppose mesh M has an initial circle packing metric (Γ_0, Φ). Let u_i be the logarithm of γ_i associated with vertex v_i. The discrete Ricci flow is defined as follows:

$$\frac{du_i(t)}{dt} = (\bar{K}_i - K_i), \tag{4}$$

where $\bar{K}_i$ and K_i are the target and current Gaussian curvatures at v_i and t is the time. Discrete Ricci flow continuously deforms the circle packing metric according to the difference between the current and target Gaussian curvatures, such that the curvature evolves like a heat diffusion process. Convergence of discrete surface Ricci flow is proved in [14]. The final circle packing metric induces the metric which satisfies the target Gaussian curvature.

Discrete Ricci flow is a negative gradient flow of a special energy form, the so called *discrete Ricci energy*:

$$f(\mathbf{u}) = \int_{(\Gamma_0,\Phi)}^{(\Gamma,\Phi)} \sum_{i=1}^{n} (\bar{K}_i - K_i) du_i, \tag{5}$$

where (Γ_0, Φ) is the initial circle packing metric, which induces the surface original metric. It has been shown in [14] that the Ricci energy is convex, therefore it has a unique global minimum. The minimum corresponds to the desired metric (Γ, Φ), which induces the target Gaussian curvature. Therefore the discrete Ricci flow converges to this unique global minimum. Furthermore, the speed of convergence can be estimated by the following formula [14]:

$$|K_i(t) - \bar{K}_i| < c_1 e^{-c_2 t}, c_1, c_2 > 0,$$

namely the convergence is exponentially fast.

C. Optimal Flat Metric

Analytically, the distortion of the metric at each vertex is given by u_i. This motivates us to define the *total distortion energy* as

$$E(\mathbf{K}) = \int_{\mathbf{K_0}}^{\bar{\mathbf{K}}} \sum_{i=1}^{n} u_i dK_i,$$

where $\bar{\mathbf{K}}$ and $\mathbf{K_0}$ represent the set of vertex target and initial Gaussian curvatures, and n is the number of vertices of mesh M. The integration is along an arbitrary path from $\mathbf{K_0}$ to the target curvature $\bar{\mathbf{K}}$. This energy is the Legendre dual to the Ricci energy given in Eqn. 5. Therefore it is also convex and there exists a unique global minimum for a given $\bar{\mathbf{K}}$. Define

$$\Omega = \bigcap \{\sum K_i = 2\pi\chi(M)\} \bigcap \{k_j = 0, v_j \notin \partial M\},$$

where $\chi(M)$ is the Euler characteristic number of M. Our problem is now formulated as:

$$\min_{\mathbf{K}\in\Omega} E(\mathbf{K}), \tag{6}$$

which is among all possible flat metrics of M satisfying the Gauss-Bonnet Theorem, which one introduces the least distortion from the estimated curved metric of M?

Theorem 2: The solution to the optimization problem 6 is unique, and satisfies

$$u_j = const, \forall v_j \in \partial M. \tag{7}$$

Proof: The distortion energy $E(\mathbf{K})$ is convex. The domain Ω is a linear subspace of the original domain $\{\mathbf{K} | \sum_i K_i = 2\pi\chi(M)\}$. Therefore the restriction of $E(\mathbf{K})$ on Ω is still convex, it has a unique global optimum at an interior point. The gradient of

the energy is $\nabla E(\mathbf{K}) = (u_1, u_2, \cdots, u_n)$. At the optimal point, the gradient is orthogonal to Ω. Assume $v_i \in \partial M, 1 \leq i \leq m$, then the normal vector to Ω is given by $(1, 1, \cdots 1, 0, \cdots, 0)$. Therefore the gradient is along the normal vector. So equation 7 holds. If we set the constant as 1, the optimal flat metric is the one which satisfies the two conditions: during the process to distort the estimated metric to a flat one, we only distort the metric of interior vertices; at the end of the process, Gaussian curvatures of all interior vertices equal to zero.

III. Algorithm

In this section we give the implementation details of our proposed localization algorithm.

A. Computing Optimal Flat Metric

The estimated metric (edge length) of the triangular mesh M can be considered as a set of unit edge length since each edge is approximately a fixed K=4 hops in out experiments. If the distances between neighboring nodes can be more accurately estimated, the approximated unit edge length can be replaced. The initial circle packing metric (Γ_0, Φ) of M can be easily constructed by assigning each vertex v_i a circle with radius equal to the unit edge length which forms Γ_0, and computing the intersection angle of circles assigned to v_i and v_j for each edge e_{ij} which forms the Φ.

Then boundary vertices located on the boundary edges of M which adjacent with only one face are detected and marked. For those non-marked vertices (interior vertices), their target Gaussian curvatures $\bar{K}$ are set to zero. For all vertices of M, the logarithm of the circle radius γ_i assigned to vertex v_i is initialized to zero. In each iteration of discrete Ricci flow, only non-marked vertices are involved. Specifically, an interior vertex v_i collects the u values from its direct neighbors and update its adjacent edge length with $l_{ij} = e^{(u_i + u_j)}$. For each triangle f_{ijk} adjacent with vertex v_i, v_i can easily compute the corner angle $\angle_i^{jk}$ based on inverse cos law:

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

Therefore, current discrete Gaussian curvature K_i at v_i can be computed as the excess of the total angle sum at v_i (Eqn. 1). If for every interior vertex 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), the discrete Ricci flow converges. Otherwise, each interior vertex 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 0.1 in our experiments).

When the algorithm stops, all the curvature flux has been absorbed by boundary vertices, such that the interior vertices have zero Gaussian curvature, which induces a flat metric. Since in each step of the algorithm, there is always no deformation of circle radii for boundary vertices (e.g., $u_i - u_i^0 = 0, v_i \in \partial M$). According to Theorem 1, the computed flat metric introduces the least distortion from the estimated metric to be planar.

B. Isometrically Planar Embedding

Isometric embedding is a propagation process, starting from one vertex, embedding the whole triangular network into plane with computed flat metric (edge length) preserved. For simplicity, we let the vertex with the smallest ID (denoted as v_0) initiate the embedding process. Its coordinates are set to $(0,0)$. Then it arbitrarily selects one of its direct neighbors, e.g., v_j, and sets its coordinates to $(0, l_{ij})$. For vertex v_k, which is adjacent to both v_i and v_j, it calculates the intersection points of the two circles with centers at v_i and v_j, and radii of l_{ik} and l_{jk}, respectively. Then, one of the intersection points that satisfies $(uv(v_j) - uv(v_i)) \times (uv(v_k) - uv(v_i)) > 0$ is chosen as the coordinates of v_k. The procedure continues until all vertices of M have computed their planar coordinates.

Note that if the triangular network has more than one boundary (e.g., inner holes), we need to slice holes open to change the triangular network to a topological disk before embedding. First an initiator is elected on each boundary. The initiator will advertise the size of its boundary in terms of the number of boundary edges via simple flooding on the triangular network. As a result, each initiator learns a set of boundaries and their sizes. Let B_0 denote the longest boundary, and v_i the initiator of other Boundary B_i $(i > 0)$. For each B_i, through a local flooding, its initiator v_i finds a shortest path L_i to B_0. Then holes are 'sliced' open along the set of shortest paths $(L_1, L_2, ...)$, where each vertex on L_i is split to two virtual vertices with one on each side. Such spliced vertex will have two sets of coordinates after embedding and it will use the average as its planar coordinates.

For planar embedding of non-landmark nodes, each node n_i first finds its three nearest landmarks, denoted as v_1, v_2, v_3 with planar coordinates (x_1, y_1),(x_2, y_2), and (x_3, y_3) respectively. Let d_1, d_2, and d_3 be 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 planar coordinates (x_i, y_i) simply by minimizing the mean square error among the distances:

$$\sum_{j=1}^{3} (\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - d_j)^2. \tag{8}$$

It deserves special note that the planar embedding step of the proposed approach is fundamentally different with graph rigidity based localization methods [10]. For an extracted global planar structure, the proposed algorithm computes first the optimal flat metric with discrete Gaussian curvature equals to zero for all interior vertices, which guarantees that the embedding process can be determined at each step for every single edge. While in [10], the extracted structure is embedded to plane by minimizing a least square energy which can't guarantee a global planar embedding and the embedded network can still curve around and self intersect.

C. Discussion

1) Time Complexity and Communication Cost: The time complexity of triangulation a sensor network is linear to the size of the network n. Its communication cost, measured by

the number of messages, is also $O(n)$, since it is a completely local algorithm.

Computing the optimal flat metric is based on discrete Ricci flow. Theoretically, its time complexity (the number of iterations) is independent of the size of the network size, given 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 [14]. Since each vertex only needs to exchange u values with its direct neighbors, the communication cost is linear to the triangular mesh size m, with $O(-C\frac{log\varepsilon}{\lambda}gm)$, where g is the average degree of vertices of the triangular mesh. Note that for a general network, $m << n$, and g is less than eight.

During the isometric planar embedding step, two rounds of flooding are involved to slice holes of a triangular mesh open with communication cost $O(m)$. The embedding of the triangular mesh is a propagation process, with time complexity and communication cost both linear to $O(m)$.

2) Error Propagation: If error is introduced to the estimated or measured metric in a small area, we can estimate the error propagation on the entire network with our algorithm. Here we show that this problem is closely related to the discrete Green's function defined on the triangular mesh with a circle packing metric.

As shown in figure 1 (b), there is a unique circle marked with red orthogonal to all three vertex circles. The center of the circle is called the *power center* for triangle f_{ijk}. Denote the distance from the power center to edge $[v_i, v_j]$ as h_{ij}^k. Suppose $[v_i, v_j]$ is also adjacent to another triangle $[v_j, v_i, v_m]$, then we define the edge coefficient

$$w_{ij} = \frac{h_{ij}^k + h_{ji}^m}{l_{ij}},$$

where l_{ij} is the length of the edge. The discrete Laplace-Beltrami operator for the circle packing metric is defined as $\Delta = (d_{ij})$

$$d_{ij} = \begin{cases} -w_{ij} & [v_i, v_j] \in E \\ \sum_j w_{ij} & i = j \\ 0 & otherwise \end{cases}$$

By direct computation, it shows the differential relation between the curvature $d\mathbf{K}$ and the radii $d\mathbf{u}$ is given by

$$d\mathbf{K} = \Delta d\mathbf{u}.$$

Suppose, there is only one node v_i which has the measurement error, the other nodes have no error, $d\mathbf{K} = \delta v_i$, δ is the Dirac function. We want to estimate the impact on $d\mathbf{u} = \Delta^{-1}\delta(v_i)$, namely, the *discrete Green's function*.

If the mesh and its circle packing metric are given, the discrete Green's function can be computed directly. For general cases, we can only give rough estimations. Assume a planar triangle mesh satisfies the following condition:

$$\frac{1}{C} < \frac{l_{max}}{l_{min}} < C, C > 1,$$

and the diameter for each triangle is less than ε, which is sufficiently small comparing to the diameter of the whole network, then the discrete Green's function is similar to the smooth Green's function

$$G(p,q) = \frac{1}{|p-q|},$$

where p and q are two points far away from the boundary. This formula shows that the impact to point q of a wrong measurement at point p decreases quickly with their distance.

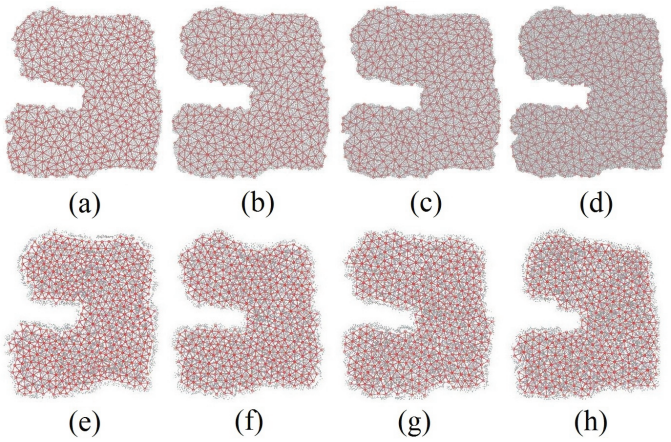


Fig. 2. Networks with variant nodal density: (a)-(d) the original network with increased nodal density; (e)-(h) the embedding results of our algorithm. All the networks have the same communication range and under the same transmission model. (a) Average nodal degree $d = 9.4$, with localization error 0.514 in (e); (b) Average nodal degree $d = 12.6$, with localization error 0.322 in (f); (c) Average nodal degree $d = 15.2$, with localization error 0.28 in (g); (d) Average nodal degree $d = 18.5$, with localization error 0.246 in (h).

IV. Simulations and Comparison

We carry out extensive simulations under various scenarios to evaluate how well our algorithm performs under different topologies and how performance is affected by different factors such as node density, communication model (UDG, quasi-UDG, log-Norm model, and probability model), and non-uniform node distribution. We compare our algorithm with those achieving the highest localization accuracy, which include the centralized MDS approach (MDS-MAP) [1], the distributed MDS approach (MDS-MAP(P)) [2], the centralized (C-CCA) neural network and the distributed (D-CCA) neural network approaches [5]. Our approach exhibits superior performance over other localization schemes. Note that we use red points to represent the selected landmark nodes and grey points for those non-landmark nodes. The localization error used in simulations is computed as the ratio of the average node distance error (all sensors in the network) and the averaged transmission range.

A. Networks with Variant Nodal Density

In general, connectivity based localization schemes favor high nodal density, with hop counts approximating well the true shortest distance. Figure 2 (a)-(d) gives the testing networks: reverse C-shape with the same communication range but average nodal degree increased from 9 to 18. We apply our scheme to localize the selected landmarks, then non-landmark

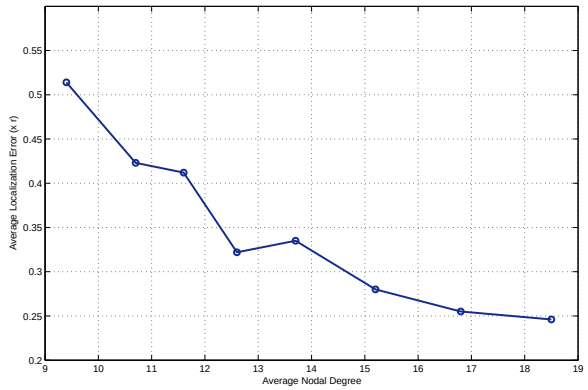


Fig. 3. Localization error for the reverse C-shape network as shown in Figure 2 decreases when the nodal degree is increased.

nodes in the network find their 3 nearest landmarks to compute their own coordinates. The localization error decreases with the increase of nodal density, which is shown in Figure 3.

B. Networks with Different Transmission Model

We still use the same reverse C-shape network to evaluate the performance of our algorithm under different transmission models, with average nodal degree $d = 16.5$ (under UDG model). For a network with the same number of nodes and the same communication range, different transmission models induce different sets of landmarks (each node has different neighborhoods), thus different triangular meshes are generated. In our experiments, the transmission range of a UDG model is 1. Two nodes are definitely connected when their distance is less than 1. Under the Quasi-UDG model, two nodes are definitely connected when their distance is less than α set to 0.75, definitely not connected when their distance is larger than 1, while they have a probability of ρ set to 0.5 to be connected when the distance is between α and 1. For Log-Norm model [17], since the receiving power is log-normally distributed, we simplify it as when the distance between two nodes is larger than 1, they are not connected; when the distance is less than 1, they have a probability $P(d)$ to be connected, where d is the distance and $P(d)$ satisfies the log-normal distribution with $\alpha = 2$ and $\sigma = 4$. For Probability model, when the distance of two nodes are less then 1, they have a probability equal to constant set to 0.65 to be connected. The results shown in Fig. 4 all give small localization errors.

C. Networks with Non-uniform Nodal Distribution

We also test our algorithm on network with non-uniform node distribution. The reverse C-shape network has the nodal density increased from the bottom to the top with nodal degree ranging from 11.3 to 18.7, as shown in Figure 5. Our algorithm also gives a reasonable good localization result.

D. Comparison with Other Methods on Networks With Different Topologies

A set of representative networks with irregular outside boundary shapes and different topologies are listed in Figure

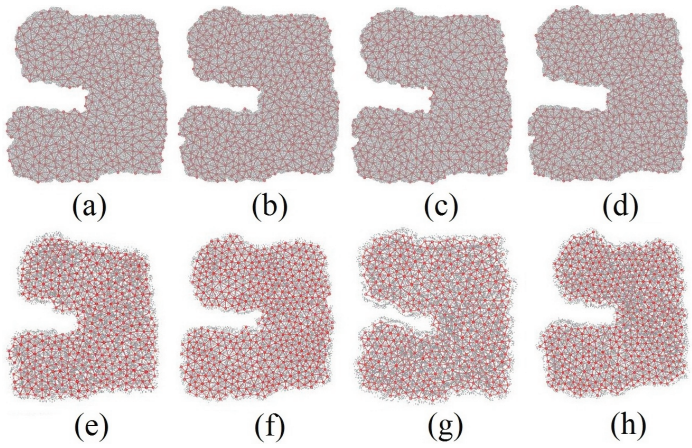


Fig. 4. Networks with different transmission models: (a)-(d) the original networks with different transmission models which result in different nodal degrees and constructed triangular meshes; (e)-(h) the embedding results of our algorithm. All the networks have the same number of nodes and the same communication range. (a) UDG model with transmission range 1, and localization error 0.25 in (e); (b) QUASI-UDG model with $\alpha = 0.75$ and $\rho = 0.5$, and localization error 0.34 in (f); (c) Log-Norm model with $\mu = 0.5$ and $\rho = 4$, and localization error 0.42 in (g); (d) Probability model with $p = 0.65$, and localization error 0.43 in (h).

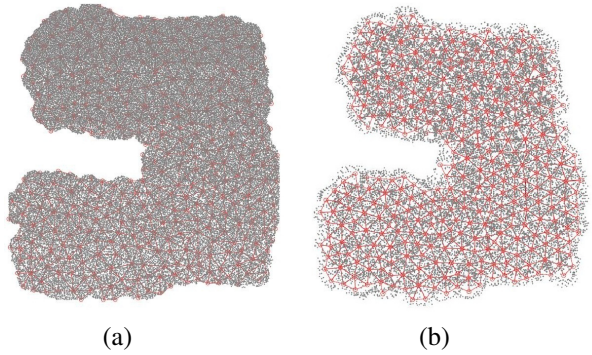


Fig. 5. (a) The original network with non-uniform node distribution. The nodal degrees rang from 11.3 to 18.7. (b) The embedding result of our algorithm with the localization error 0.46.

6. A red line segment is drawn for each node, starting from its real coordinates marked with red and ending at the computed coordinates marked with grey. Clearly, the length of the line segment represents the error of localization at that node. Overall, the more and the longer the red lines are, the worse the performance of the localization is.

As can be observed in Figs. 6 a(1) and c(1), a(3) and c(3), C-CCA and MDS-MAP both yield large distortions for the nodes on those branches. This is because the pair wise hop count approximated distances among those nodes are much longer than their actual Euclidean distances due to the reversed C and the star shapes, which leads to noticeable errors around those areas.

MDS-MAP(P) and D-CCA compute local maps first, and then merge them to a global map. Since local maps are "smooth" and do not have large "tentacles" in general, the shortest paths are free of significant distortions. Therefore,

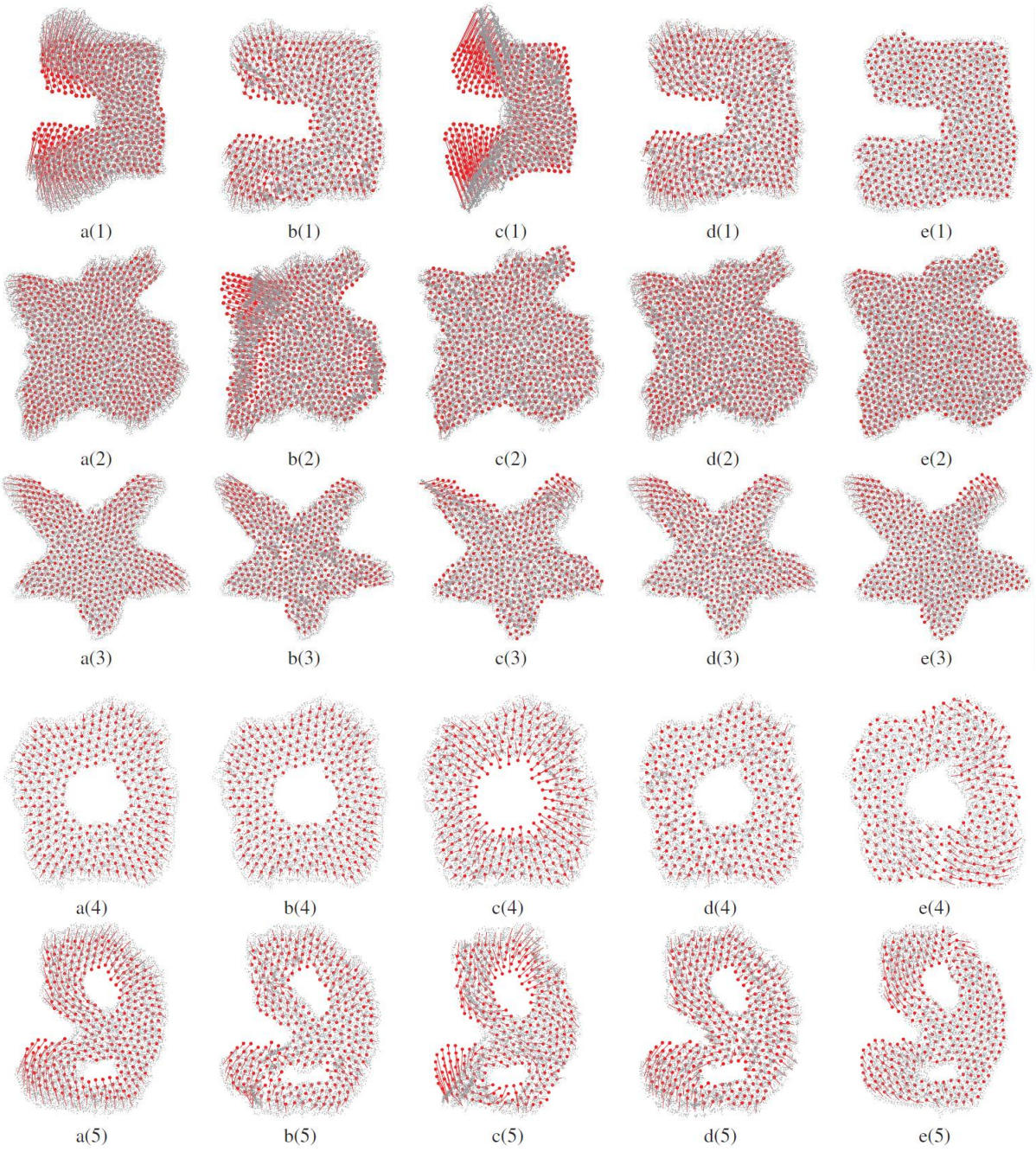


Fig. 6. Comparison of different localization approaches on networks with general topologies. a(1)-a(5): C-CCA scheme; b(1)-b(5): D-CCA scheme; c(1)-c(5): MDS-MAP scheme; d(1)-d(5): MDS-MAP(P) scheme; e(1)-e(5): Ricci scheme. A red line segment is drawn for each node, starting from its real coordinates marked with red and ending at the computed coordinates marked with grey.

Scenario	C-CCA	D-CCA	MDS-MAP	MDS-MAP(P)	Ricci
Topology 1 (Fig. 6 a(1)-e(1))	2.10	0.88	2.52	0.89	0.29
Topology 2 (Fig. 6 a(2)-e(2))	0.71	0.69	0.56	0.68	0.32
Topology 3 (Fig. 6 a(3)-e(3))	0.72	0.64	0.62	0.75	0.48
Topology 4 (Fig. 6 a(4)-e(4))	0.78	0.70	1.18	0.61	0.55
Topology 5 (Fig. 6 a(5)-e(5))	1.17	0.8	1.27	0.99	0.63

TABLE I
AVERAGE LOCALIZATION ERRORS WITH DIFFERENT APPROACHES ON NETWORK MODELS SHOWN IN FIGS. 6.

both methods achieve better performance (i.e. less distortion and lower errors) than the centralized approaches in networks with irregular boundary conditions. This can be clearly seen in Figs. 6 b(1) and d(1). However, if a network has smooth boundary (e.g., Figs. 6 a(2)-e(2)), the shortest paths are not seriously distorted and thus the centralized schemes can perform better since they utilize more constraints to localize the nodes.

For networks with holes, similar as the reverse C-shape network, MDS-MAP(P) and D-CCA perform better (shown in Figs. 6 b(4) and d(4), b(5) and d(5)) compared with their centralized counterparts MDS-MAP and C-CCA (shown in Figs. 6 a(4) and c(4), a(5) and c(5)). But they have to pay the cost to merge different subnetworks together.

The proposed discrete Ricci flow based approach, on the contrary, always achieves the least overall localization errors in all simulated scenarios as demonstrated in Figs. 6 e(1), e(2), e(3), e(4) and e(5). The average localization errors with different approaches on models shown in Figs. 6 are summarized in Table I. The distribution of localization errors on the reverse C-shape network is illustrated in Fig. 7 for different approaches. Since the results under other networks show similar statistics, we omitted them here. As can be seen, the localization errors of the Ricci-based approach are nicely distributed at the lower range.

E. Testing of Error Propagation

We conduct the following experiments to test the propagation rate of errors resulting from bad estimation or wrong measurements. Two triangular networks are shown in the right top of Figure 8 (a) and (b). If all the edge lengths can be accurately measured and considered as input to construct the initial circle packing metric, the triangular network can be embedded to plane with very high localization accuracy, with errors $2.8e-5$ and $1.64e-5$ for the two networks respectively. While if the measurement around one vertex is

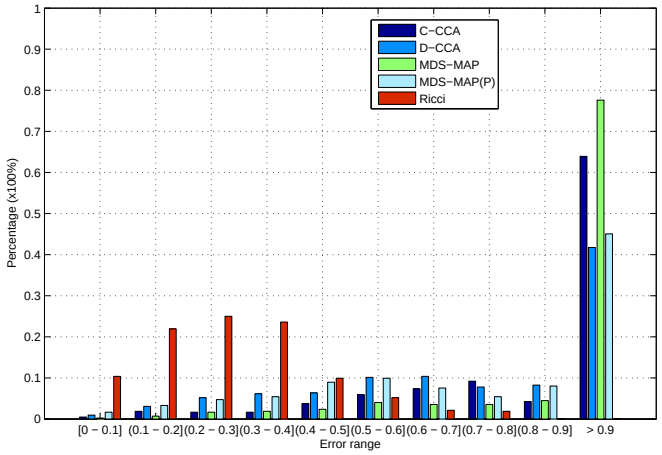


Fig. 7. The distribution of localization errors on the reverse C-shape network with different approaches: C-CCA, D-CCA, MDS-MAP, MDS-MAP(P) and Ricci.

wrong, for example, much slower response of the node to its neighboring nodes' signals will result in a much longer distance approximation. The input error will affect not just this node, but also other nodes in the network (e.g., error will be propagated). We introduce such measurement error at one selected vertex (marked with green in the two triangular networks) by multiplying some constant K (K=2.5 in the tests) with the lengths of its neighboring edges (marked with green too), such that the vertex is no longer planar based on the wrong measurement. We measure the effect of the "one vertex measurement error" by comparing the change of the circle radius of each vertex with the correct input and with the distorted input. Figure 8 (a) and (b) shows the changes of circle radii of vertices of the two testing networks respectively. The closer a vertex is to the distorted vertex, the bigger the change of its circle radius is. Green dots are vertices with changed radii, while red dots are those without any change of radii. It is obvious that the error propagation decreases dramatically with the increase of the distance to the distorted vertex. The localization error of the two triangular networks with distorted vertices are 0.0107 and 0.0081 respectively.

F. Computing Time

As we analyzed in Section III-C1, the time complexity of our overall algorithm is linear to size of the network. The convergence rate of computing flat metric using Ricci flow is exponentially decreasing with the curvature error. In our experiments, we set the step length to 0.1, and the error threshold to $1e-5$. Figure 9 gives the convergence of Ricci flow for part of our simulation networks.

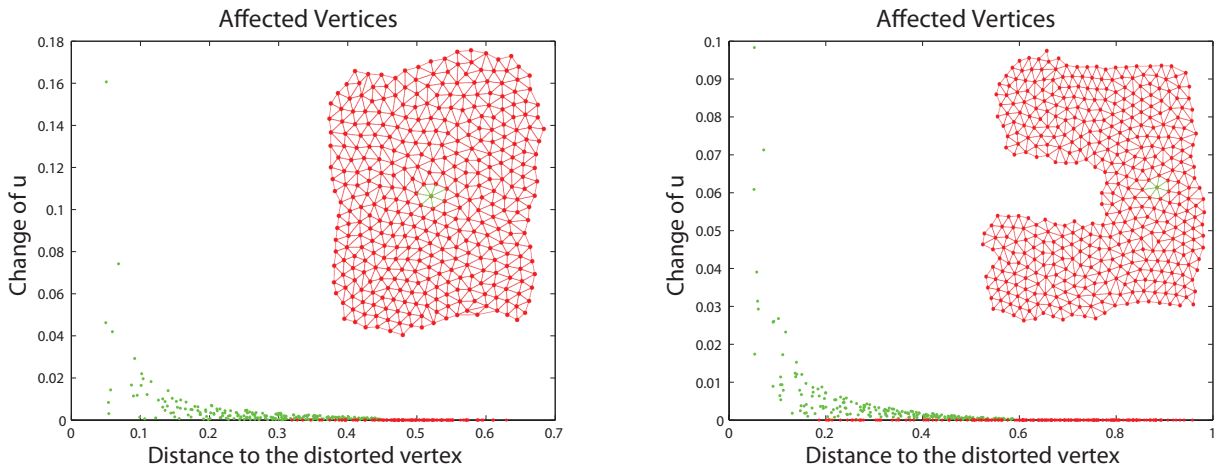
V. Conclusion and Future Works

This work proposes a novel localization method based on mere connectivity. The method is theoretically guaranteed and numerically stable. The computation is fully distributed and highly scalable, with its computation time and communication cost linear to the size of the network. If there is some error occurs at a node, the error propagation decreases dramatically with respect to the distance from the node to others.

A limitation of the proposed solution is that it works only for a target field that is flat everywhere. In our future work we will relax our assumption and develop new algorithms for a large-scale sensor network deployed on general 3D surface instead of ideal 2D plane. In addition, the accuracy of our proposed solution depends on the quality of the extracted triangular mesh from a sensor network. We are working on distributed algorithms for building highly reliable triangulation mesh for general sensor networks.

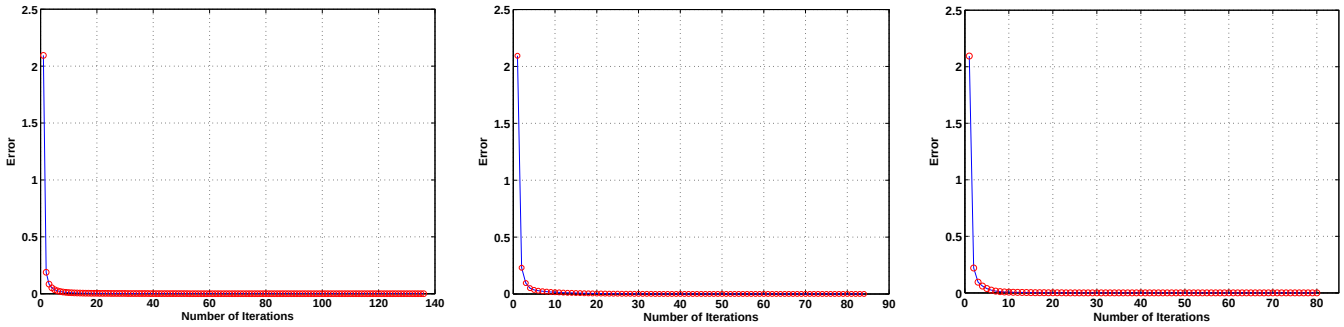
Acknowledgements.

M. Jin, S. Xia and H. Wu are partially supported by NSF Nets-1018306, X. Gu is partially supported by NSF Nets-1016829.



(a) (b)

Fig. 8. The change of the circle radius of a vertex decreases dramatically with its distance to the distorted vertex. Dots marked with red represent those vertices not affected at all by those wrong measurements.



(a) (b) (c)

Fig. 9. Convergence rate of the proposed discrete Ricci flow based approach under different networks, with step length 0.1, error threshold $1e-5$. (a) The inverse C shape network with the number of landmarks 424, and the convergence time 2 seconds; (b) The network with one hole, the number of landmarks 283, and the convergence time 1 second; (c) The network with two holes, the number of landmarks 297, and the convergence time 1 second.

REFERENCES

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

[2] Y. Shang and W. Ruml, "Improved mds-based localization," in *Proc. of INFOCOM*, pp. 2640–2651, 2004.

[3] 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.

[4] G. Giorgetti, S. Gupta, and G. Manes, "Wireless localization using self-organizing maps," in *Proc. of IPSN*, pp. 293–302, 2007.

[5] 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.

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

[7] 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.

[8] H. Lim and J. Hou, "Distributed localization for anisotropic sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 2, pp. 11–37, 2009.

[9] 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.

[10] 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.

[11] 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.

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

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

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

[15] 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.

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

[17] C. Bettstetter and C. Hartmann, "Connectivity of wireless multihop networks in a shadow fading environment," *Wireless Networks*, vol. 11, no. 5, pp. 571–579, 2005.