2012 32nd IEEE International Conference on Distributed Computing Systems

Connectivity-based and Boundary-Free Skeleton Extraction in Sensor Networks

[1,3]Wenping Liu [1]Hongbo Jiang [2]Chonggang Wang [1]Chang Liu [1]Yang Yang [1]Wenyu Liu [4]Bo Li

[1]Department of Electronics and Information Engineering, Huazhong University of Science and Technology, China
[2]InterDigital Communications, U.S.A. [3]Hubei University of Economics, China
[4]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong
Email:[1]{wenpingliu2009,hongbojiang2004,liucheung1105,yangyang8795}@gmail.com, [2]cgwang@ieee.org, [4]bli@cse.ust.hk

***Abstract*— In sensor networks, skeleton (also known as medial axis) extraction is recognized as an appealing approach to support many applications such as load-balanced routing and location-free segmentation. Existing solutions in the literature rely heavily on the identified boundaries, which puts limitations on the applicability of the skeleton extraction algorithm. In this paper, we conduct the first work of a connectivity-based and boundary-free skeleton extraction scheme, in sensor networks. In detail, we propose a simple, distributed and scalable algorithm that correctly identifies a few skeleton nodes and connects them into a meaningful representation of the network, without reliance on any constraint on communication radio model or boundary information. The key idea of our algorithm is to exploit the necessary (but not sufficient) condition of skeleton points: the intersection area of the disk centered at a skeleton point x should be the largest one as compared to other points on the chord generated by x, where the chord is referred to as the line segment connecting x and the tangent point in the boundary. To that end, we present the concept of ϵ-centrality of a point, quantitatively measuring how "central" a point is. Accordingly, a skeleton point should have the largest value of ϵ-centrality as compared to other points on the chord generated by this point. Our simulation results show that the proposed algorithm works well even for networks with low node density or skewed nodal distribution, etc. In addition, we obtain two by-products, the boundaries and the segmentation result of the network.**

***Index Terms*—Wireless Sensor Networks, Skeleton Extraction, Neighborhood Size, Voronoi Cell.**

I. Introduction

Many algorithms and protocols of wireless sensor networks are intimately related to the underlying geometric space where sensors are deployed. For example, the global topology of a wireless sensor network has a fundamental influence on designing point-to-point routing protocol, data gathering/processing scheme, etc. In the literature skeleton extraction is one of the most widely used approaches to topology discovery; many protocols on top of it, such as localization [14], navigation [5], routing([3], [4], [7], [16]), distributed index for multi-dimensional data and random sampling [18], and the like, have witnessed an improved performance.

For example, load balance has been a compelling objective in geographical routing schemes, considering the limited resource of each sensor. Typical geographical routing protocols not considering the underlying geometric features often lead to the unbalance of loads. As a result, the boundary nodes are heavily overloaded, and then exhaust the batteries quickly. Skeleton-based routing protocols consist of two main steps: naming scheme and routing scheme. Roughly speaking, for naming scheme, we name each sensor node based on its relative position to the skeleton. And each skeleton node has a unique label which represents its relative position. By doing so, virtual coordinates of nodes are obtained. For routing scheme, the routing message is *forced* to follow a direction almost parallel to the skeleton while maintaining an approximately shortest path between the source and destination pair. Skeleton-based routing protocols are desirable due to its avoiding overload along boundaries whereby no node gets overloaded [3]. Another application of skeleton extraction is shape segmentation [18], [12] which is to divide an irregular network into nicely shaped subnetworks, such that traditional algorithms designed for a simple geometric region can be applied. With extracted skeleton graph, nearby skeleton nodes are merged into a sink. Other nodes compute their parents with higher hop-count to the boundaries, "flowing" to the sinks. Those nodes flowing to the same sink are grouped to the same segment finally.

MAP [3], [4] and CASE [10], [11] are two representative studies in the literature for skeleton extraction. Taking boundary information as an input, MAP first identifies medial nodes and connects them into a meaningful representation. MAP then applies the medial axis to name each node; the naming scheme and its coupled routing protocol reflect the network connectivity very well (that is, two faraway nodes in the connectivity graph can not be nearby in the virtual coordinate system). Consequently, local minimum phenomenon can be avoided; and load balance of each node is guaranteed. Unfortunately, as pointed out in [10], [11], MAP suffers from boundary noise, that is, a small bump on the boundary will incur a long branch. As such, CASE [10], [11] is proposed. The beauty of CASE is that by detecting so-called corner points, the boundaries are segmented into several branches; and a node is a skeleton node such that it has two nearest boundary nodes which belong to different boundary branches. The boundary noise can be controlled by the user-defined threshold value for corner point.

An implicit assumption of MAP and CASE, however, is that boundary information is already given. This might limit their applicability since in practical, especially for networks

1063-6927/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDCS.2012.10

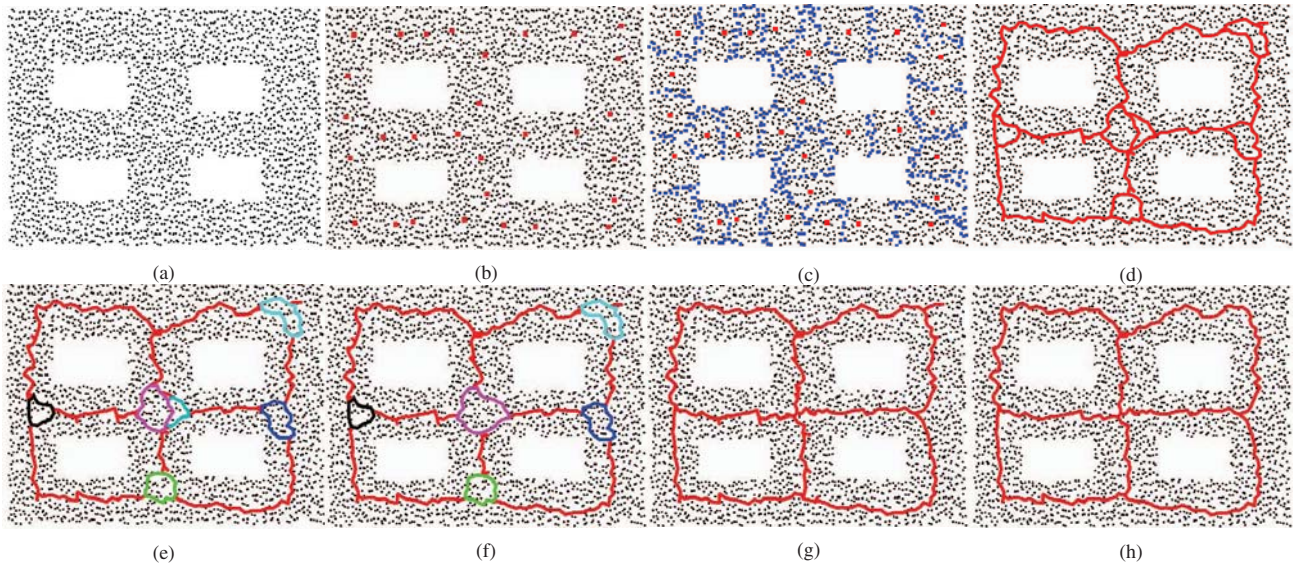


Fig. 1. Skeleton extraction of a Window-shaped network, which has 2592 nodes with avg.deg 5.96. (a) Original network; (b) Skeleton nodes (marked in red); (c) Segment nodes (marked in blue); (d) Coarse skeleton; (e) Identifying loops. Different fake loops are marked in different colors, and genuine skeleton loops are marked in red; (f) Merging fake loops; (g) Deleting fake loops; (h) The final skeleton after pruning.

with low node density, boundary recognition itself is still challenging and most boundary identification algorithms do not work well [13]. In contrast, in this paper we are primarily interested in extracting the skeleton of a wireless sensor network with pure local connectivity without reliance on any constraint on communication radio model or boundary information. Interestingly, the boundaries and the segmentation result are two by-products of our algorithm.

We conduct the first work of a connectivity-based and boundary-free skeleton extraction scheme. Our study is motivated by the observation that skeleton nodes are placed medially in the network, therefore, 1) they often have a larger neighborhood size as compared with other nodes; and 2) each neighbor of a skeleton node also has a large neighborhood size. Accordingly, we propose a novel skeleton extraction algorithm. Instead of computing the hop count distance of each node to the boundaries and identifying skeleton nodes based on their hop count distances to the boundaries, the proposed algorithm identifies skeleton nodes purely based on their neighborhood sizes and that of their neighbors. More specifically, each node first collects its k-hop neighborhood size by controlled flooding, followed by collecting the set of k-hop neighborhood sizes of its l-hop neighbors. With these neighborhood sizes, each node computes an index. A node identifies itself as a skeleton node if it has a locally maximal index. Subsequently, these identified skeleton nodes flood in the entire network to build a set of Voronoi cells, within each of which there is only one skeleton node (referred to as a site). Note that two Voronoi cells may be adjacent, at the intersection of which there are some nodes almost equidistant to the two corresponding sites; and we call these nodes as segment nodes. Next we select the segment node with the largest index to connect the two sites in a greedy manner. Eventually, all sites are connected and a coarse skeleton is obtained. As there can be unwanted skeleton branches (i.e., with small length) and/or loops, we finally refine the obtained coarse skeleton. The refinement process includes identifying loops, merging and deleting fake loops, and pruning. After this process, the refined skeleton is established.

In summary, our proposed algorithm distinguishes itself from previous work with the following features:

- **Boundary-free.** Our algorithm does not require identified boundaries.
- **Scalable.** Our algorithm has a linear message complexity and the running time is sub-linear in the network size, which makes it applicable for large-scale networks.
- **Distributed.** Each step of our algorithm is conducted in a distributed manner.
- **Robust to network model and node distribution.** Our proposed algorithm has no constraint on communication radio model or node distribution.

The reminder of this paper is organized as follows. In Section II, we describe the theoretical foundation of our algorithm. Next we present details of our algorithm in Section III. In Section IV, we evaluate the effectiveness of the proposed algorithm through extensive simulations. Section VI presents the related work, and Section VII concludes this paper.

II. Theoretical Foundation

A. Skeleton Node Identification

We assume that a sensor network has n sensors. Further, we assume that nodes are all deployed uniformly at random in the field (We will show later that this technical assumption is not necessary, but here it helps us to simplify our description), and all sensors have the same communication radio range R.

Each node has only knowledge of with whom it can directly communicate. Our question is given as follows: is it possible to extract skeleton using only connectivity information?

As pointed out in [8], a node close to the boundaries often has a smaller neighborhood size (or node degree) as compared to that of an interior node. Since skeleton nodes lie medially, they often have high node degrees. However, we can not simply identify a skeleton node by computing its node degree, because there might be fluctuations in node density due to the random deployment of sensors, and consequently, many nodes may be misinterpreted as skeleton nodes.

Skeleton node identification is the key of our algorithm, and our skeleton node identification is motivated by the following two observations.

Observation 1. A skeleton node often has a large neighborhood size.

Observation 2. The neighbor of a skeleton node also has a large neighborhood size.

To formally state our observations, we first concentrate on the properties of skeleton points in continuous domain in Section II-B. The implementation in discrete networks is deferred later.

B. Skeleton in Continuous Domain: Principle

We first recall the definition of skeleton in continuous domain. Suppose D is a bounded open set in R^2. We denote by $S(D)$ the skeleton of D, and ∂D the boundaries. Let $dist(x, y)$ denote the distance between point x and y of the object D. According to Blum's definition [1], the skeleton (or so-called medial axis) of D is the set of locus of the centers of maximal disks (i.e., circles with at least two tangent points on the boundaries ∂D). Let $D(v, r(v))$ be the disk centered at $v \in D$ with radius $r(v)(> 0)$, then $S(D) = \{v \in D | D(v, r(v))$ is a maximal disk$\}$. Further, we call the line segment, xy, which connects a point $x \in S(D)$ with a tangent point $y \in \partial D$, as a chord; and we say such chord is generated by the skeleton point x. Typically, each skeleton point has at least two chords; and for a point not on the skeleton, it has been proven in [3] that

Lemma 1: For an arbitrary point $v \notin S(D)$, if v is on a chord xy, where $x \in S(D), y \in \partial D$, then y is the only closest boundary point of v.

Lemma 2: For an arbitrary point $v \notin S(D)$, it stays on a unique chord.

Let $D_i(v, r(v))$ be the intersection of $D(v, r(v))$ with D, that is, $D_i(v, r(v)) = D(v, r(v)) \cap D$. Let $\lambda(\cdot)$ be a 2-dimensional volume function (i.e., the area) that assigns a measure to a subset of 2-dimensional Euclidean space. Then, $\lambda(D_i(v, r(v)))$ is the intersection area (short for the area of the intersection) of $D(v, r(v))$ with D; and we have $\lambda(D(v, r(v))) \geq \lambda(D_i(v, r(v))) > 0$.

Theorem 1: Let xy be a chord, where $x \in S(D), y \in \partial D$; and let R be a positive real value which is no greater than $dist(x, y)$. Then for any point $v \in D$ on chord xy, 1) $\lambda(D_i(v, R)) \leq \lambda(D_i(x, R))$ holds if $R < dist(x, y)$, and 2) $\lambda(D_i(v, R)) < \lambda(D_i(x, R))$ if $R = dist(x, y)$.

Proof: Since $R \leq dist(x, y)$, the disk $D(x, R)$, centered at x with radius R lies completely inside D. That is, the intersection area $\lambda(D_i(x, R)) = \lambda(D(x, R)) = \pi R^2$.

Case 1: $R < dist(x, y)$**.** If v is a point on chord xy such that $dist(x, v) \leq dist(x, y) - R$, then the disk centered at v with radius R has at most one tangent point y with ∂D. As a matter of fact, $dist(v, y)$ and $dist(x, y)$ are the Euclidean distance transforms of point v, y respectively. Since $dist(x, v) \leq dist(x, y) - R$, which implies that $dist(x, y) \geq dist(v, y) + R$, the disk $D(v, R)$, centered at v with radius R, must lie inside the disk centered at x with radius $dist(x, y)$ [2]; and thus $D(v, R)$ lies completely inside D. Therefore, $\lambda(D_i(v, R)) = \lambda(D(v, R)) = \pi R^2$.

If v is a point on chord xy such that $dist(x, v) > dist(x, y) - R$, then $D(v, R)$ can not lie completely inside D, otherwise, $D(x, dist(x, y))$ will not be a maximal disk, which contradicts to the fact that x is a skeleton point. Accordingly, $\lambda(D_i(v, R)) \leq \lambda(D(v, R)) = \pi R^2$.

Overall, if $R < dist(x, y)$, then $\lambda(D_i(v, R)) \leq \lambda(D_i(x, R))$.

Case 2: $R = dist(x, y)$**.** Similarly, if $R = dist(x, y)$, for any point v on chord xy, $D(v, R)$ can not lie completely inside D. We prove this by contradiction. If $D(v, R)$ lies completely inside D, then the distance from x to y is $dist(x, y) = dist(x, v) + R > R = dist(x, y)$, which is a contradiction.

■

Theorem 1 says that the intersection area of the disk, centered at a skeleton point x with radius R, with the object is the largest as compared to that of other points on the chord generated by the skeleton point x, when R is no greater than the radius of the maximal disk centered at x. In other words, Theorem 1 provides a necessary (but not sufficient) condition for detecting skeleton points. One can further deduce that for any R, equations 1) and 2) in Theorem 1 still hold. In fact, this is guaranteed by the continuity of the intersection area of any disk, centered at a point of the same chord, with any radius R.

Theorem 2: For a chord generated by a skeleton point x, the intersection area of the disk, centered at a point v on this chord with radius R, with the object D is a continuous function of $dist(x, v)$.

Proof: To prove this, we show that for any point I on this chord, we can find a point K on this chord such that the distance between I, K approaches 0. Please see Fig. 2. That is, if points I, K are such that $dist(I, K)$ approaches 0, then $|\lambda(D_i(I, R)) - \lambda(D_i(K, R))|$ will approach 0. To that end, we only need to prove that the difference of the areas formed by arcs NHO and MHP approaches 0, and equivalently, $|\angle HKP - \angle HIO|$ approaches 0. Note that $\angle HKP - \angle HIO = (\pi - \angle PKI) - \angle HIO = \angle KQI$. Let a, b, c denote the lengths of the line segments KQ, IQ, KI, respectively. According to Cosine Rule, we have

$$\cos \angle KQI = \frac{a^2 + b^2 - c^2}{2ab} = \frac{a^2 + b^2}{2ab} - \frac{c^2}{2ab}$$

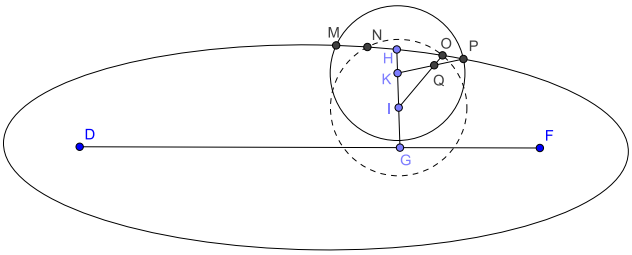


Fig. 2. Illustration of Theorem 2. The line DF represents the skeleton of the eclipse-shaped object, G is a skeleton point and H is G's tangent boundary point. Point I, K are two points of the chord GH, where I is the center of the dashed circle, and K is the center of the solid circle.

Obviously, since $a^2 + b^2 > 2ab$, when $c(= dist(I, K))$ approaches 0, $cos\angle KQI$ will approach 1, that is, $\angle KQI$ approaches 0, which proves the claim. ■

Corollary 1: For two points $v_1, v_2 \notin S(D)$ on the same chord xy, where $x \in S(D), y \in \partial D$. If $dist(v_1, x) > dist(v_2, x)$, then for any positive R, $\lambda(D_i(v_1, R)) \leq \lambda(D_i(v_2, R))$. The equality holds if and only if $R < dist(v_1, y)$.

One direct implication of Corollary 1 is that for a boundary point s, the intersection area of D with the disk centered at s with any positive radius is often small. Such implication has been adopted in [8] to find boundary nodes.

As the skeleton lies medially inside the object, each point on the skeleton is a central point. To quantitatively measure the centrality of a point, we now present the ϵ-centrality of a point s, a metric to describe how central the point s is to the object, as follows:

Definition 1: For any given $\epsilon > 0$ such that $N_\epsilon(s)$, the set of nodes having a distance less than ϵ to s, lies completely inside D. We define the ϵ-centrality (or the average intersection area of the ϵ neighbor) of s, $C_R^\epsilon(s)$, as the ratio of the sum of the intersection areas of all disks, with radius R and centered at each point $v \in N_\epsilon(s)$, with D to the area of $N_\epsilon(s)$. That is,

$$C_R^\epsilon(s) = \frac{\int\int_{v \in N_\epsilon(s)} \lambda(D_i(v, R))dxdy}{\lambda(N_\epsilon(s))}. \quad (1)$$

Based on this definition, we have several claims:

Theorem 3: For any positive R, a point $v \in S(D)$ has the largest ϵ-centrality as compared with that of any point on the chord generated by point v.

Proof: We only need to prove that for any point s on the chord generated by point v, $C_R^\epsilon(s) < C_R^\epsilon(v)$. Principally, the nominator in Equation 1 is a double integral problem, and it is easy to derive that for any point $v_1 \in N_\epsilon(s)$, there is a point $v_2 \in N_\epsilon(v)$ such that $dist(v, v_1) > dist(v, v_2)$; according to Corollary 1, $\lambda(D_i(v_1, R)) < \lambda(D_i(v_2, R))$ holds. Therefore, $\int\int_{w \in N_\epsilon(s)} \lambda(D_i(w, R))dxdy < \int\int_{w \in N_\epsilon(v)} \lambda(D_i(w, R))dxdy$; and $C_R^\epsilon(s) < C_R^\epsilon(v)$, which proves the claim. ■

Corollary 2: For two points $v_1, v_2 \notin S(D)$ on the same chord xy, where $x \in S(D), y \in \partial D$. If $dist(v_1, x) > dist(v_2, x)$, then $C_R^\epsilon(v_1) < C_R^\epsilon(v_2)$.

Corollary 3: For a skeleton point v, $\lambda(D_i(v, R)) > C_R^\epsilon(v)$; for a boundary point w, $\lambda(D_i(w, R)) < C_R^\epsilon(w)$;.

Proof: For the skeleton point v, since

$$\int\int_{s \in N_\epsilon(v)} \lambda(D_i(s, R))dxdy$$
$$< \int\int_{s \in N_\epsilon(v)} \lambda(D_i(v, R))dxdy = \lambda(D_i(v, R)\lambda(N_\epsilon(v))$$

we have

$$C_R(v) < \frac{\lambda(D_i(v, R)\lambda(N_\epsilon(v))}{\lambda(N_\epsilon(v)} = \lambda(D_i(v, R))$$

Similarly, for the boundary point w, we have

$$C_R(w) > \lambda(D_i(w, R))$$

■

In summary, a skeleton point has the largest intersection area of the disk, centered at itself with an arbitrary radius R, with the object and ϵ-centrality of a skeleton point is also the largest as compared to any point of the chord generated by this skeleton point. With these properties, we can construct an index to identify skeleton nodes in discrete domain, which will be addressed next.

C. Skeleton in Discrete Networks: Implementation

Now we study how to identify skeleton nodes in discrete networks. We regard the deployment of sensors as a sampling from the underlying geometry, thus we can easily extend the properties of skeleton points in continuous domain to the discrete networks. Let R be the communication radio range of each sensor. Let the k-hop neighbors of node p, denoted by $N_k(p)$, be the set of nodes at most k hops from p. Note that we have assumed that sensors are uniformly deployed in the region D. Then for any node $p \in D$, the probability, $P(p, kR)$, of the event that a node s falls into the intersection of the disk $D(p, kR)$ with D, denoted by $D_i(p, kR)$, equals $\frac{\lambda(D_i(p,kR))}{\lambda(D)}$; and the number, $N(p, kR)$, of senors falling into $D_i(p, kR)$ (i.e., the k-hop neighbors of p), follows a binomial distribution $b(n, P(p, kR))$, with expected value $E(N(p, kR)) = nP(p, kR)$ and variance $\sigma^2(N(p, kR)) = nP(p, kR)(1 - P(p, kR))$.

Note that here the k-hop neighbors of node p is actually the discrete analog of $D_i(p, kR)$ in continuous domain. As mentioned in Subsection II-B, a skeleton point v has the largest $\lambda(D_i(v, R))$ as compared with other points on the chord generated by v, therefore, the number of nodes falling into $D_i(p, kR)$ (i.e., the k-hop neighborhood size of p) will also be the largest in the expected sense. Thus, it is possible for us to identify a skeleton node according to its neighborhood size; and the node with locally maximal k-hop neighborhood size must be on the skeleton. As such, we have

Definition 2: If node p has a locally maximal k-hop neighborhood size, we call p a skeleton node.

Due to the discrete nature of wireless sensor networks and the existence of rounding errors, by Definition 2, lots of nodes may claim themselves as skeleton nodes. It is noted that, as depicted by Theorem 3, a skeleton point has the largest ϵ-centrality among other points of the chord generated by this

skeleton point. As such, we introduce another metric, say, l-centrality of a node p, as follows.

Definition 3: For a node p and a positive integer-valued constant l, its l-centrality $c_l(p)$ is defined as the average of the k-hop neighborhood sizes of its l-hop neighbors.

Obviously, p's l-centrality $c_l(p)$ is the discrete analog of $c_\epsilon(p)$ in continuous domain. Therefore, we have

Lemma 3: For a skeleton node p, its l-centrality is the largest as compared with the nodes on the chord generated by p.

As such, for a skeleton node, both of its k-hop neighborhood size and its l-centrality are locally maximal; our skeleton node identification is based on the combination of these two metrics, and we find that by doing so, we can eliminate noises more greatly than using only one metric, e.g., the k-hop neighborhood size.

Definition 4: For each node p, we define the index of p, denoted by $i(p)$, as the average of p's k-hop neighborhood size and its l-centrality, namely, $i(p) = \frac{|N_k(p)|+c_l(p)}{2}$.

As a matter of fact, one can also define the index of p as $i(p) = f(|N_k(p)|) + g(c_l(p))$, where both $f()$ and $g()$ are monotonically increasing functions. Obviously, the index $i(p)$ can also describe how central p is. A skeleton node should have the largest index as compared to other nodes on the same chord. In other words, if a node has a locally maximal index, it must be a skeleton node. Therefore, we have

Definition 5: For a node p, if $i(p)$ is locally maximal, then we call p a critical skeleton node.

Note that two parameters, namely k and l, are used in our skeleton node identification process. Intrinsically, our algorithm is not sensitive to these two parameters (detailed in Subsection V-B); and in our simulations, we simply set $k = l = 4$. It is noted that the skeleton nodes identified by Definition 5 are generally disconnected. We thus propose a series of operations to connect them properly, and finally the skeleton is obtained.

III. Skeleton Extraction Algorithm

In this section, we show how to extract skeleton based on mere neighborhood size of each node, without reliance on boundaries. The key step of our algorithm is to identify skeleton nodes. Since boundaries are unknown, we can not calculate the hop count distance of each node to the boundaries, and skeleton nodes can not be identified in the way described in [3], [4], [10], [11]. As mentioned earlier, our method is based on neighborhood size. Due to the discrete nature of wireless sensor networks, only a part of skeleton nodes identify themselves. Consequently, it is still challenging to connect these identified skeleton nodes to form a meaningful representation.

We first give an outline of our algorithm, and the details of each step are presented later.

1) **Skeleton Node Identification**. We first let each node have knowledge of its k-hop neighborhood size, and the k-hop neighborhood sizes of its l-hop neighbors. Subsequently, each node computes its index. The node with locally maximal index identifies itself as a critical skeleton node.

2) **Voronoi Cells Construction**. With these identified critical skeleton nodes, we let them perform a limited flooding; and each node then keeps record of its nearest critical skeleton node and the path from the nearest critical skeleton node to itself. As a consequence, a set of Voronoi cells are constructed, within each of which there is only one critical skeleton node. Further, for two adjacent Voronoi cells, there are a few nodes who have a small difference (e.g., 1 in our implementation) between its hop count distances to the two critical skeleton nodes; we call these nodes as *segment nodes*, and these two Voronoi cells are called *adjacent*.

3) **Coarse Skeleton Establishment**. After the construction of Voronoi cells, for two adjacent Voronoi cells, we choose a segment node between these two cells. The chosen segment node should have the largest index among all segment nodes associated with these two cells. Subsequently, the chosen segment node sends a message along the reverse path it has kept during the process of Voronoi cells construction; and two paths from the segment node to its two nearest critical skeleton nodes are built. Accordingly, a coarse skeleton is established.

4) **Final Clean-up**. The obtained coarse skeleton may have unwanted branches and/or loops, which need further treatment. The presence of loops is either caused by obstacles, which indicates there may exist holes and the loops are genuine, or by incorrectly connecting three or more adjacent Voronoi cells and generating a fake loop. The latter case should be avoided, otherwise the obtained skeleton is not homotopic to the original network. As such, in the final stage, we conduct a series of operations to identify loops, merge and delete fake loops, and trim the skeleton branches with small length. Finally, the refined skeleton is generated.

A. Skeleton Node Identification

In this subsection, we exploit local connectivity information to identify critical skeleton nodes. To this end, we let each node have knowledge of its k-hop neighborhood size, and that of its l-hop neighbors. This can be done with two rounds of limited flooding. In the first round, each node v floods a message containing the ID of v, and a counter which implies the hop count the message has traveled. When an intermediate node q receives the flooded message, it executes the following rules:

- if the counter in the message is less than the given threshold, say k, q keeps record of the ID of v, increases the counter by one, and forwards the updated message to its neighbors;
- otherwise, q simply discards the message.

This way, each node has knowledge of its k-hop neighbors; and it is trivial to compute its k-hop neighborhood size.

In the second round, each node broadcasts its k-hop neighborhood size within its l-hop neighbors. After that, each node q computes the index $i(q)$; and the node with locally maximal index identifies itself as a critical skeleton node. Please see Fig. 1 (b) as an example.

B. Voronoi Cells Construction

It is noted that the identified critical skeleton nodes are generally disconnected, so we need to connect them properly to form a set of skeleton curves. To that end, we first construct a series of Voronoi cells, which are fundamental for the establishment of a coarse skeleton; and we will present the details of the coarse skeleton establishment in next subsection.

The Voronoi cells are constructed in a distributed fashion as follows. First, we let all skeleton nodes flood in the network to build a shortest path tree, rooted at each skeleton node. The flooded message from the skeleton node p includes the ID of node p, and a counter indicating how many hops the message has traveled. Let $dist(p_1, p_2)$ denote the hop count distance between nodes p_1 and p_2. When an intermediate node, say q, receives the message from p, it makes its decision according to the following rules:

- if q has not received such a flooded message before, q joins the tree rooted at p, increases the counter by one, and forwards the updated message to its neighbors;
- else if q has received message(s) from other critical skeleton node(s) before (q may have received more than one message before. Without loss of generality, we assume that the nearest one to q is $p\prime$), and the absolute difference of hop count distances (indicated by the counter in the message) of q to the two critical skeleton nodes, namely, $|dist(q,p) - dist(q,p\prime)|$, is less than a given threshold α (in our implementation, $\alpha = 1$), q keeps record of the two nearest skeleton nodes, and does not forward the message from p to its neighbors;
- otherwise, q simply discards the message.

If the identified critical skeleton nodes flood at roughly the same time, and the message travels at approximately the same speed, the message cost will be very low. Eventually, many skeleton trees rooted at critical skeleton nodes will be constructed. Each node keeps track of its nearest critical skeleton node(s) and the distance to the critical skeleton node(s). Those nodes who have the same nearest critical skeleton node naturally form a sub-region. Note that ties are allowed here so that there may be a few nodes which are equidistant to two or more skeleton nodes.

let m denote the number of the identified critical skeleton nodes, $L = \{l_i, i = 1, 2, ..., m\}$ be the set of identified skeleton nodes $l_i (i = 1, 2, ..., m)$, and $C = \{c_i, i = 1, 2, ..., m\}$ be the set of sub-regions, where c_i is generated by l_i, that is, for any node $p \in c_i$, $dist(p, l_i) = \min_{j=1,2,...m, j\neq i} dist(p, l_j)$. We have

Theorem 4: For any sub-region $c_i (i = 1, 2, ..., m)$, it is connected.

Proof: We only need to prove that for any node $q \in c_i$, the shortest path between q and its nearest critical skeleton node l_i lies entirely in c_i. We prove this by contradiction. Assume that there is a node $v \notin c_i$ on the path from q to l_i. That is, the nearest critical skeleton node of v is not l_i. Let $l_j (j \neq i)$ be $v's$ nearest critical skeleton node. Obviously, $dist(v, l_j) < dist(v, l_i)$. According to the triangle inequality, we have $dist(q, l_j) \leq dist(q, v) + dist(v, l_j) < dist(q, v) + dist(v, l_i) = dist(q, l_i)$. This contradicts to the fact that $q \in c_i$ which implies that $dist(q, l_i) = \min_{j=1,2,...m, j\neq i} dist(q, l_j)$. ■

Corollary 4: The set $C = \{c_i, i = 1, 2, ..., m\}$ is a Voronoi diagram, where each region c_i is a Voronoi cell.

As such, the network has been partitioned into a set of Voronoi cells. In this sense, we also call the identified critical skeleton nodes as *sites*. Note that between two Voronoi cells, there are a few nodes which are almost equidistant to the two sites (here the word "almost" is used to indicate exceptions where a node may not be strictly equidistant to two sites, thus we introduce a parameter α, as mentioned above); we call these nodes as *segment nodes*, as shown in Fig. 1 (c).

C. Coarse Skeleton Establishment

With these Voronoi cells formed, we now connect all sites properly to form a coarse skeleton. We first identify the segment nodes between pairwise Voronoi cells. The identification of segment nodes is very simple: if a node has kept record of two sites during the process of Voronoi cells construction in subsection III-B, it marks itself as a segment node. For two Voronoi cells, if there are many segment nodes between them, we call these two Voronoi cells *edge adjacent*; if there is only one segment node, we call these cells *point adjacent*. If two Voronoi cells are either edge adjacent or point adjacent, we call these two Voronoi cells (and their sites) *adjacent*. Note that there might be three or more Voronoi cells mutually adjacent. For this case, we call the nodes which are almost equidistant to the sites of these Voronoi cells as *Voronoi nodes*, which are the discrete analogue of Voronoi vertices in continuous domain.

With segment nodes identified, we then generate the coarse skeleton as follows. We first select a segment node that has the largest index among all segment nodes which are almost equidistant to the same two sites. Next, the selected segment node sends a message along the reverse paths to its nearest sites it has kept during the process of Voronoi cells construction in subsection III-B; and two paths from the segment node to its two nearest sites are built. As a result, these two sites are connected, and a coarse skeleton is established, as shown in Fig. 1 (d). Without ambiguity, we call each node on the coarse skeleton as a skeleton node.

D. Final Clean-up

After the establishment of the coarse skeleton, an important step is to refine the coarse skeleton, since there may be unwanted skeleton branches (i.e., with small length), and more importantly, there might be fake skeleton loops. Generally, the existence of loops in the coarse skeleton may be caused by obstacles (or nodes failure, etc) in the sensing field, which indicates that there exist holes in the network and the skeleton loops are genuine. For this case, the loops should not be deleted, so that the skeleton is homotopic to the original network. The presence of fake skeleton loops is due to the fact that during the process of connecting pairwise adjacent

sites, three or more Voronoi cells may be mutually adjacent. As a result, a fake skeleton loop will be generated. In principle, the skeleton of an object D is homotopic to the original object D [6], [15]. That is, we can "continuously deform" the object D to obtain the skeleton. With the presence of fake skeleton loops, however, the skeleton is no longer homotopic to D. In this subsection, we focus on how to refine the obtained coarse skeleton; the refinement process consists of the following sub-steps:

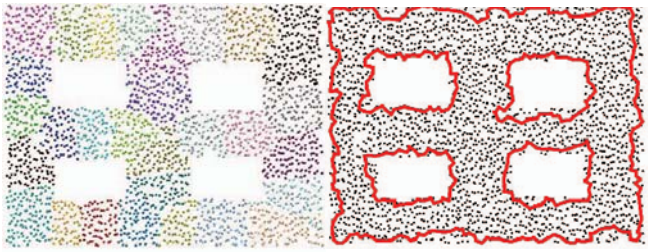
Identify Genuine Skeleton Loops and Fake Skeleton Loops. As discussed above, the presence of skeleton loops is either because that there are obstacles inside the network, or because that three or more adjacent sites are mistakenly connected; and the latter case should be avoided. To that end, we first identify loops via limited flooding within the obtained coarse skeleton. More specifically, for each pairwise adjacent sites, we choose two farthest segment nodes (referred to as *end nodes*) associated with these two sites. Intuitively, each end node is either a boundary node, or a Voronoi node. Let the end nodes issue a limited flooding without crossing the coarse skeleton. Doing so, we obtain many end node loops formed by the shortest paths between pairwise end nodes and *bounded* by the coarse skeleton. If the number of nodes on an end node loop is small, indicating that there is at least one Voronoi node, then the skeleton nodes on the path between pairwise nearest sites of these end nodes form a fake loop; otherwise, the skeleton loop is a genuine skeleton loop. Please see Fig.1 (e) for an example. After this, each skeleton node knows whether it is on a loop (or fake loop) or not.

Merge and Delete Fake Loops. As mentioned earlier, the presence of fake loops incurs that the obtained skeleton is not homotopic to the original network, and further treatments are needed. It might be possible that two fake loops are adjacent, namely, some skeleton nodes belong to two fake loops. For this case, we let these nodes give up their identities of skeleton nodes and be ordinary nodes; and consequently, a larger fake loop will be formed, as shown in Fig. 1 (f). After this, within each fake loop, we regard the nodes on the fake loops as outer boundary nodes, and adopt existing solutions (e.g., CASE [10], [11]) to extract the skeleton of the sub-region bounded by each fake loop. Finally, all nodes (except those that have three or more neighboring skeleton nodes) on each fake loop give up their identities of skeleton nodes. As a result, we obtain a skeleton without fake loops, shown in Fig. 1 (g).

Pruning. After deleting fake loops, there might be some branches with small length. We now prune these skeleton branches. The pruning process is similar with that described in [10], [11], so we omit the details here. Finally, we obtain the refined skeleton, as shown in Fig. 1 (h).

E. By-products

Note that during the process of skeleton extraction, we also obtained two by-products of our algorithm, namely, segmentation result and network boundaries. In the process of Voronoi cells construction, we have decomposed the network into a set of Voronoi cells, as shown in Fig. 3 (a); and in the final clean-up stage, after the process of identifying loops, we obtained the network boundaries, as shown in Fig. 3 (b).



(a) (b)

Fig. 3. Two by-products of our algorithm on Fig.1. (a) Segmentation result; (b) Network boundaries.

IV. Performance Evaluation

In the previous sections, we have presented a distributed and scalable skeleton extraction algorithm based on connectivity information only, without reliance on boundaries. We now evaluate the effectiveness of our skeleton extraction algorithm through a series of simulations.

In our simulation settings, nodes are deployed uniformly in the field. We assume that all nodes have the same communication radio range R. The communication radio model is Unit-Disk Graph (UDG) by default. That is, two nodes are connected if and if their separation is no greater than R. The default parameters for skeleton node identification are $k = l = 4$ unless otherwise stated.

We first examine the performance of our algorithm on various network topologies, and then we study the behavior of our algorithm under different node densities. Next, we test the robustness of our algorithm by using Quasi-Unit Disk Graph (QUDG) model and log-normal communication radio model. We finally examine the performance of our algorithm on networks with skewed node distribution.

A. Simulation Results under Different Scenarios

We study the performance of our algorithm under different scenarios, with average node degrees ranging from 5.75 to 9.60, as shown in Fig. 4. In Fig. 4 (a), the network has a concave hole, and the average node degree is only 6.54. We can clearly see that, even without boundary nodes, our algorithm extracts a medially placed skeleton. In Fig. 4 (b), the average node degree is much lower (only 5.75), but we still obtain a good abstract of the network. From Fig. 4 (c)-(j), we can see that the obtained skeletons are all desirable and they capture very well the global geometric and topological features of the original networks.

B. Effect of Node Densities

We vary the radio range of the Window-shaped network in Fig. 1 to adjust the average degree of nodes, as shown in Fig. 5. We have already seen that our algorithm obtained a good skeleton for the Window-shaped network with low node density in Fig. 1 (h); and here we further study four different

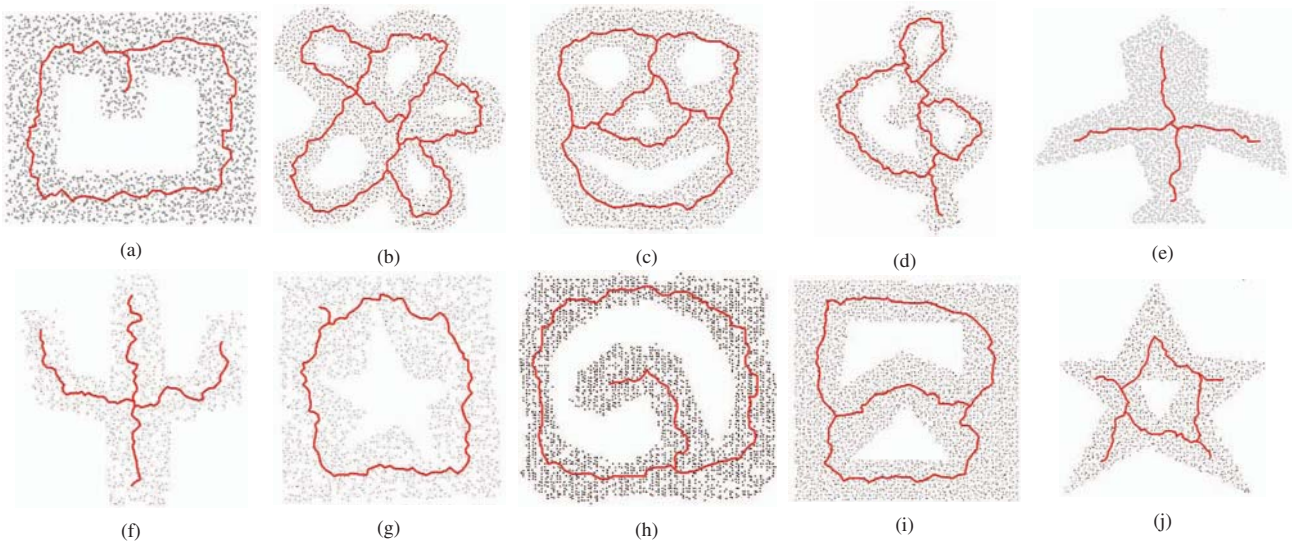


Fig. 4. Skeleton extraction on various scenarios. (a) One hole network of 2734 nodes, with avg.deg 6.54; (b) Flower-shaped network of 2422 nodes, with avg.deg 5.75; (c) Smile-shaped network of 2924 nodes, with avg.deg 6.35; (d) Music-shaped network of 1301 nodes, with avg.deg 6.5; (e) Airplane-shaped network of 2157 nodes, with avg.deg 7.86; (f) Cactus-shaped network of 2172 nodes, with avg.deg 6.70; (g) Star-shaped hole network of 2893 nodes, with avg.deg 8.99; (h) Spiral-shaped network of 2812 nodes, with avg.deg 9.60; (i) Two holes network of 3346 nodes, with avg.deg 6.79; (j) Star-shaped network of 1394 nodes, with avg.deg 6.59.

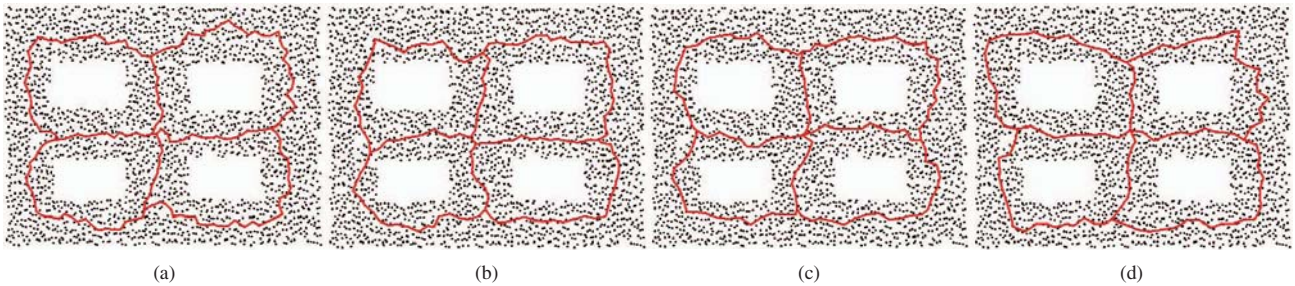


Fig. 5. Skeleton extraction under different node densities. (a) Avg.deg is 9.95; (b) Avg.deg is 14.24; (c) Avg.deg is 19.23; (d) Avg.deg is 22.72.

average degrees, namely, 9.95,14.24, 19.23 and 22.72. As to be expected, with the increase of node density, our algorithm produces very stable skeletons, showing that our algorithm is robust to node density.

C. Effect of Communication Radio Models

We first examine the robustness of our algorithm by using Quasi-unit disk graph (QUDG) model (see Fig. 6) characterized by a parameter $0 \leq \alpha < 1$. Under QUDG, there exists a link between pairwise nodes if their separation is less than $(1-\alpha) \times R$; and a link exists with probability $0 < p < 1$ between pairwise nodes when the separation is between $(1-\alpha) \times R$ and $(1+\alpha) \times R$; no link exists when the separation is greater than $(1+\alpha) \times R$.

In our settings, $p = 0.3, \alpha = 0.4$, and we enlarge the radio range so that the network is overall connected. Compared with Fig. 1 (h), the skeleton in Fig. 6 (a) captures the correct topological features, even though it is a slightly rougher than that in Fig. 1 (h). Similarly, the skeleton in Fig. 6 (b) is not as smooth as that in Fig. 4 (j), but it still lies medially and captures the right topology.

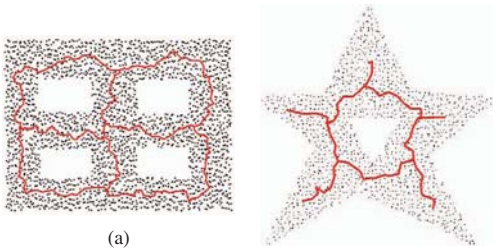
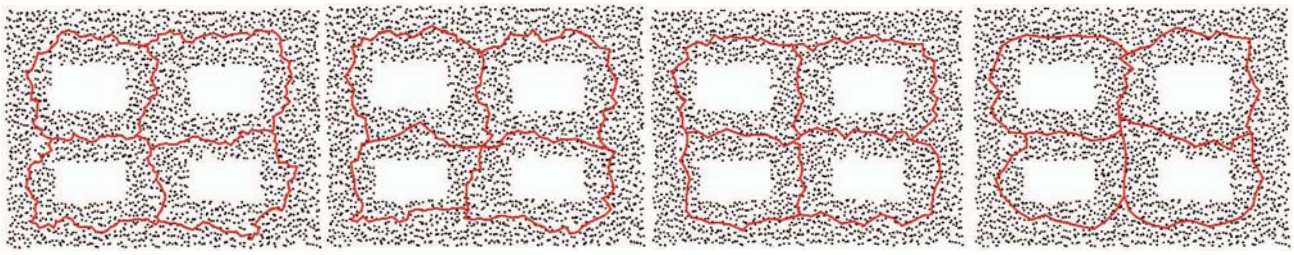


Fig. 6. Skeleton extraction under QUDG. $\alpha = 0.4, p = 0.3$. (a) Window-shaped network in Fig. 1 (a); (b) Star-shaped network in Fig. 4 (j).

We next study the performance of our algorithm when the communication radio model is log-normal (see Fig. 7). In this case, the probability that a link exists between nodes i and j is based on a log-normal shadowing radio model [9]:

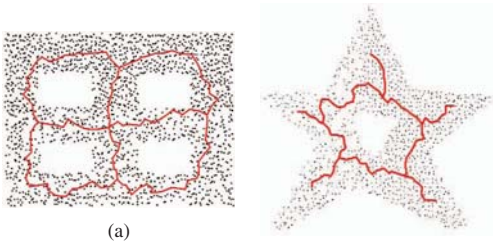
$$p(\hat{r}) = \frac{1}{2}[1 - erf(\alpha \frac{log\hat{r})}{\xi})], \xi \triangleq \sigma/\eta \qquad (2)$$

where $\hat{r}$ is the normalized distance between nodes i and j,



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

Fig. 7. Skeleton extraction of Window-shaped network following log-normal communication radio model. (a) $\xi = 0$, and avg.deg is 5.19; (b) $\xi = 1$, and avg.deg is 6.92; (c) $\xi = 2$, and avg.deg is 11.54; (d) $\xi = 3$, and avg.deg is 20.69.



(a) (b)

Fig. 8. Skeleton extraction of networks with skewed node distribution. (a) Window-shaped network, avg.deg 8.15; (b) Star-shaped network, avg.deg 7.16.

$\alpha = 10/(\sqrt{2}log10)$ is a constant, σ is the standard deviation of shadowing, and η is the pathless exponent. Empirically, ξ may vary between 0 and 6 [9]. An important feature of log-normal radio model is that the link between two nodes whose normalized distance is less than 1 may not exist, and the link between nodes whose normalized distance is larger than 1 exists with a nonzero probability.

We test our algorithm on Window-shaped network with log-normal radio model for four different values of ξ, namely, 0, 1, 2 and 3, as shown in Fig. 7. As can be seen, with the increase of ξ, the average node degree increases accordingly, but our algorithm achieves very stable results. In addition, for larger ξ, the skeleton is smoother than that for smaller ξ. This is due to the fact that for larger ξ, the probability of having a link between two faraway nodes is larger than that for smaller ξ.

D. Effect of Node Distribution

Finally, we study the behavior of our algorithm on networks with skewed node distribution, see Fig. 8. The nodes in Fig. 8 (a) are a sample drawn from Fig. 1 (a), where the upper part is denser than the lower part. However, the obtained skeleton is comparable with that in Fig. 1(b). In Fig. 8 (b), nodes in left part are drawn from Fig.4 (j) with probability 0.65, and the nodes in right part are drawn with probability 1.00. An interesting observation is that in the upper part, the node distribution is non-uniformly, that is, the left upper part is sparser than the right part; and as a result, the skeleton branch bends toward right.

V. Discussions

A. Complexity Analysis

Theorem 5: Our algorithm has a time complexity of $O(\sqrt{n})$ and message complexity of $O((k + l + 1)n)$, where n is the number of sensor nodes, and k, l are the parameters for identifying skeleton nodes.

Proof: We first prove the time complexity is $O(\sqrt{n})$. First, the time complexity of skeleton node identification is $O(1)$. Second, Voronoi cells construction has a time complexity of at most $O(\sqrt{n})$. Third, the time complexity of connecting two sites is $O(1)$, thus the coarse skeleton establishment has a time complexity of at most $O(\sqrt{n})$. Finally, the refinement of the obtained coarse skeleton has a time complexity of at most $O(\sqrt{n})$. Overall, the time complexity of our algorithm is $O(\sqrt{n})$.

We next prove the message complexity is $O((k+l+1)n)$. The dominating factors of message complexity are skeleton node identification and Voronoi cells construction. For skeleton node identification, each node will obtain its k-hop neighborhood size and that of its l-hop neighbors, and this incurs a message complexity of $O((k+l)n)$; during the process of Voronoi cells construction, if the identified skeleton nodes flood at roughly the same time, and the message travels at approximately the same speed, the message complexity will be about $O(n)$. In total, our algorithm has a message complexity of $O((k + l + 1)n)$. ■

B. Effect of Parameters

The parameters k and l determines how many critical skeleton nodes will be identified. Generally speaking, the smaller the two parameters are, the more critical skeleton nodes will be identified, thereby more fake loops are created after connecting adjacent critical skeleton nodes. However, the mechanism of merging adjacent fake loops guarantees that the obtained skeleton is homotopic to the original network and captures the main topology features. As such, one does not need to choose k and l very carefully.

VI. Previous Work

It has been shown that skeleton (also known as medial axis) enables efficient point-to-point routing [3], [4], navigation [5] and segmentation [18], etc. While these algorithms target at the applications of skeleton, however, few literatures focus on

skeleton extraction itself; perhaps the most representative ones are MAP [3], [4] and CASE [10], [11].

In [3], [4], boundary nodes are firstly identified correctly, either manually or by using existing solutions on boundary recognition, e.g., [17]. It then identifies those nodes, who have equidistance to two distinct boundary nodes, as medial nodes. To control boundary noise, those unstable medial nodes, whose nearest boundary nodes are on the same boundary cycle and their separation is small, are not allowed. Next, MAP connects these identified medial nodes in a proper way so that the genuine geometric features of the network are kept; and a medial axis is formed. A direct application of the obtained medial axis is the design of a skeleton-aided naming and routing scheme, where each node is assigned a name by its relative position to the medial axis. The implementations show that MAP improves routing performance in terms of load balance and stretch factor, since the skeleton-aided naming and its coupled routing scheme reflect the network connectivity very well (that is, two faraway nodes in the connectivity graph can not be nearby in the virtual coordinate system); and boundary nodes will not be overloaded.

Jiang *et al.* [10], [11] proposed CASE, a novel skeleton extraction algorithm based on only connectivity information. Similar with MAP, CASE also assumes that the boundary nodes are known. The novelty of CASE is that it segments boundaries into a set of boundary branches; and the nodes equidistant to two or more boundary nodes that are not on the same boundary branches identify themselves as skeleton nodes. CASE then constructs a series of skeleton components, within each of which a skeleton arc is generated; connecting these skeleton arcs forms a coarse skeleton. The final skeleton is obtained by pruning the coarse skeleton via limited flooding.

VII. Conclusion

In this paper, we described a simple, distributed and scalable algorithm that extracts the skeleton of a wireless sensor network based on pure local connectivity information, without reliance on boundary information. We first identify a few skeleton nodes by the computation of an index, which is related to the neighborhood size of each node and that of its neighbors. With these identified skeleton nodes, we construct a set of Voronoi cells, within each of which there is only one skeleton node. Subsequently, we identify segment nodes, which are almost equidistant to two skeleton nodes; and we select the segment node with the largest index to connect its two nearest skeleton nodes and form a coarse skeleton. We finally refine the coarse skeleton by identifying loops, merging and deleting fake loops, and then trimming the skeleton branches with small length; and the final skeleton is established. To our knowledge this is the first work that realizes connectivity-based and boundary-free skeleton extraction.

In skeleton extraction, an open issue is that there is lack of understanding for what is a good skeleton so far. Perhaps an appropriate choice is based on its application. For example, when the skeleton is used for routing improvement, load balance is an important consideration. This will be one direction of our future work.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 60803115, Grant 61073147, and Grant 61173120; by the National Natural Science Foundation of China and Microsoft Research Asia under Grant 60933012; by the Fundamental Research Funds for the Central Universities under Grant 2011QN014; by the National Natural Science Foundation of Hubei Province under Grant 2011CDB044; by the CHUTIAN Scholar Project of Hubei Province; by the Youth Chenguang Project of Wuhan City under Grant 201050231080; by the Scientific Research Foundation for the Returned Overseas Chinese Scholars (State Education Ministry); and by the Program for New Century Excellent Talents in University under Grant NCET-10-408 (State Education Ministry). The corresponding author is Hongbo Jiang.

References

[1] H. Blum. Transformation for extracting new descriptors of shape, models for the perception of speech and visual form. *MIT Press*, pages 363–380, 1967.

[2] J. W. Brandt and V. R. Algazi. Continuous skeleton computation by voronoi diagram. *CVGIP:Image Understanding*, 55(3):329–338, 1992.

[3] J. Bruck, J. Gao, and A. A. Jiang. Map: Medial axis based geometric routing in sensor networks. In *Proc. of ACM MOBICOM*, 2005.

[4] J. Bruck, J. Gao, and A. A. Jiang. Map: Medial axis based geometric routing in sensor networks. *Wireless Networks*, 13(6):835–853, 2007.

[5] C. Buragohain, D. Agrawal, and S. Suri. Distributed navigation algorithms for sensor networks. In *Proc. of IEEE INFOCOM*, 2006.

[6] H. I. Choi, S. W. Choi, and H. P. Moon. Mathematical theory of medial axis transform. *Pacific Journal of Mathematics*, 18(1):57–88, 1997.

[7] Q. Fang, J. Gao, L. Guibas, V. de Silva, and L. Zhang. Glider: Gradient landmark-based distributed routing for sensor networks. In *Proc. of IEEE INFOCOM*, 2005.

[8] S. P. Fekete, A. Kroller, D. Pfisterer, S. Fischer, and C. Buschmann. Neighborhood-based topology recognition in sensor networks. In *Proc. of the 1st Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks*, 2004.

[9] R. Hekmat and P. V. Mieghem. Connectivity in wireless ad-hoc networks with a log-normal radio model. *Mobile Networks and Applications*, 11(3):351 – 360, 2006.

[10] H. Jiang, W. Liu, D. Wang, C. Tian, X. Bai, X. Liu, and W. Liu. Case: Connectivity-based skeleton extraction in wireless sensor networks. In *Proc. of IEEE INFOCOM*, 2009.

[11] H. Jiang, W. Liu, D. Wang, C. Tian, X. Bai, X. Liu, and W. Liu. Connectivity-based skeleton extraction in wireless sensor networks. *IEEE Trans. TPDS*, 21(5):710–721, 2010.

[12] H. Jiang, T. Yu, C. Tian, G. Tan, and C. Wang. Consel: Connectivity-based segmentation in large-scale 2d/3d sensor networks. In *Proc. of IEEE INFOCOM*, 2012.

[13] H. Jiang, S. Zhang, G. Tan, and C. Wang. Cabet: Connectivity-based boundary extraction of large-scale 3d sensor networks. In *Proc. of IEEE INFOCOM*, 2011.

[14] S. Lederer, Y. Wang, and J. Gao. Connectivity-based localization of large scale sensor networks with complex shape. In *Proc. of IEEE INFOCOM*, 2008.

[15] A. Lieutier. Any open bounded subset of $\mathcal{R}^n$ has the same homotopy type than its medial axis. In *Proc. of the eighth ACM symposium on Solid modeling and applications*, 2003.

[16] G. Tan, M. Bertier, and A. M. Kermarrec. Convex partition of sensor networks and its use in virtual coordinate geographic routing. In *Proc. of IEEE INFOCOM*, 2009.

[17] Y. Wang, J. Gao, and J. S. B. Mitchell. Boundary recognition in sensor networks by topological methods. In *Proc. of ACM MOBICOM*, 2006.

[18] X. Zhu, R. Sarkar, and J. Gao. Shape segmentation and applications in sensor networks. In *Proc. of IEEE INFOCOM*, 2007.