UCLA
Papers

Title
Distributed Techniques for Area Computation in Sensor Networks

Permalink
https://escholarship.org/uc/item/35f5c8sh

Authors
Greenstein, Ben
Kohler, Eddie
Culler, D E
et al.

Publication Date
2004-05-05

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Distributed Techniques for Area Computation in Sensor Networks

Ben Greenstein† Eddie Kohler† David Culler‡ Deborah Estrin†

† UCLA, Department of Computer Science

‡UCB, Department of Computer Science

†{ben,kohler,destrin}@cs.ucla.edu, ‡culler@cs.berkeley.edu

Abstract

We study four distributed techniques for computing the area of a region in a sensor network. Area calculation is a fundamental sensor network primitive, and distributed, in-network approaches prove more scalable than centralized collection in terms of energy consumption. The four techniques—Delaunay triangulations, Voronoi diagrams, and two new, simpler algorithms, inverse neighborhood and inverse neighborhood with location—vary in computational complexity, communication cost, and information required from the sensor network. We conclude that when sensors know their physical locations, our simple and efficient inverse-neighborhood approach performs comparably to more systematic, but more expensive, computational geometry algorithms. We also analyze the effects of radio range and deployment density on accuracy, and show that topologies derived from real testbeds behave quite differently from commonly seen random topologies with unit disk connectivity.

1. Introduction

Several extant sensor networks, including deployments at Great Duck Island [14] and James Reserve and the TinyDB [11] and Tiny Diffusion projects, collect sensor data and return it to an external gateway to be analyzed offline. In the logical next step, sensor networks will detect and evaluate distributed phenomena and react to properties of those phenomena in situ, without ever involving a gateway to an external network or a human user. In-network evaluation can clearly detect phenomena faster. More importantly, in-network evaluation can require less communication than external evaluation. Communication in sensor networks is a primary energy cost, and energy can limit how long a network survives. In an offline-evaluation system, all data must be sent to the collection point for later processing. The costs of communicating high-resolution data through the network would quickly deplete any battery-powered device. An in-network-evaluation system, in contrast, can decide when data is interesting and deliver only that data to the external gateway. In-network evaluation also allows different kinds of deployments, and can support in-network actuation, where the sensor network detects a phenomenon (such as a fire) and then acts accordingly (such as by turning on a fire extinguisher).

But deciding when data is interesting is harder than it might seem. How can a set of loosely coupled, energy-poor sensors decide what shape some sensed phenomenon covers? Or how large that phenomenon is? Or how acute it is, or how long it lasts? For instance, a fire might become "interesting" only if it covers a large area, in a roughly linear "fire-front" shape, and its average temperature is above 800°. Each of these questions requires that sensors cooperate on some algorithm and that cooperation should not take more communication than sending all data off the network.

This paper evaluates four techniques for the in situ determination of a homogenous region's area. Our techniques complement other current techniques for in-network evaluation, such as distributed edge detection [3, 16], hierarchical multiresolution storage [6], distributed databases [10, 7], and data modeling.

The first two techniques are derived from traditional computational geometry methods, namely Delaunay triangulation and Voronoi polygonization. They partition the sensed environment into disjoint sectors, each of which is "owned" by one or more nodes; the sensed region's area equals the sum of the areas of the sectors whose owners are in the region. The techniques can provide very low error bounds, at the expense of significantly more computation and communication; for instance, the best results are obtained when all nodes in

the network agree on the division into parts, and this agreement may require global communication.

The second two techniques use a much simpler idea, new to this work. Inverse neighborhood techniques keep track of the number of neighbors accessible within one hop from each sensor in the network. Each sensor then "owns" an area inversely proportional to its neighbor count, since assuming identical radios, more neighbors indicates a denser deployment. Inverse neighborhood techniques are susceptible to error when location information is not known and when density is not constant over the network. However, in realistic simulation scenarios, they perform surprisingly well, while requiring significantly less communication and computation than the computational geometry approaches. In fact, in one realistic simulation scenario, inverse neighborhood with location performs better than any other algorithm. When the average connectivity degree is at and above 20, the inverse neighborhood technique performs with an error of two percent. Voronoi polygonization yields an error between 12 and 21 percent for the same connectivity degrees.

The contributions of this work are the inverse neighborhood algorithms, cheap techniques for area measurement accurate enough for many purposes; a comparison of these algorithms with more precise computational geometry techniques; detailed descriptions of the tradeoffs between these algorithms; and evaluation of all the techniques in the context of real connectivity and topology data derived from a deployed network.

The paper is organized as follows: Section 2 discusses other work in this area. Section 3 describes the components of area evaluation and Section 4 covers various distributed algorithms for computing area. In Section 5 we analyze the tradeoffs of using the various techniques. We conclude in Section 6 and discuss future work in Section 7.

2. Related Work

There have been a number of developments by the sensor network community related to the efficient detection and evaluation of distributed phenomena.

Applications including boundary estimation and edge detection have also been developed and analyzed. The problem of boundary estimation of homogenous regions is considered in [13]. They explore the tradeoff between MSE and energy consumption as functions of node density and develop a boundary estimation algorithm based on multiscale partitioning methods. Chintalapudi et al. [3] investigate techniques for distributed edge detection in sensor networks. They describe and compare a statistical approach, an image processing approach and a classifier-based approach and show that the last of these performs the best.

Delaunay triangulation and Voronoi polygonization are common computational geometry algorithms and researchers have recently found use for them in sensornets. Meguerdichian et al. [12] investigate minimum exposure paths in sensor networks. As part of their work they argue that the best a target can do to avoid being sensed is to walk along the edges of a Voronoi diagram. Ganeriwal et al. [5] partition a sensor netork into a Voronoi diagram and weigh nodes' sensor values by the size of their respective enclosing Voronoi polygons to effectively compute the average sensor value taken over all space as opposed to over all nodes.

Finally, efforts are underway to make operations over spatial data first order elements of high-level sensornet languages. Welsh, for example, develops the notion of abstract regions, a family of spatial operators that capture local communication within regions of the network. One such region is an approximate planar mesh based on a pruned Yao graph [17].

3. System Design

Communication is often the primary consumer of energy in sensor network deployments [15]. Thus, every effort should be taken to locally compute as much of a partial result as possible before sending information along to a collection sink. In the context of homogenous region area calculation, a straightforward approach to minimizing communication is to have each node in a region estimate its contribution to the total area and to add these estimates together en route to a sink. This requires (1) testing region membership; (2) setting up an maintaining a routing tree; (3) locally generating partial estimates of the area; and (4) adding these estimates together. We briefly describe each of these.

For the purpose of this work, a sensor is said to belong to a homogenous region *iff* it reads a value above a user-defined threshold. Two sensors, s_0 and s_k, are said to be in the same homogenous region *iff* there exists a communication path $\{s_0, ..., s_k\}$ such that all $s_{i,0 \leq i \leq k}$ are above this threshold. Thus inclusion in a region can be determined by a local test of sensor value and a one-hop connectedness evaluation.

Once a node determines that it is in a region, a routing tree must be formed over the region to aggregate partial estimates. Nodes form a spanning tree and elect an arbitrary node in that tree to serve as root. Then, using one of the techniques described in the paper, each node estimates a local contribution to that area. These estimates are transmitted to the root of the tree. To

minimize communication, instead of forwarding each estimate individually, each node forwards the sum of its estimate and its children's estimates. Finally, the root receives these aggregate estimates and adds them together; the result is the area of the region.

In this paper, we consider several techniques for distributed calculation of area. Regardless of technique, however, the other steps in processing (i.e., membership, routing, and aggregation) remain the same.

4. Techniques

The distributed computation of the area of a region amounts to estimating how much each node contributes to this area and summing these contributions over all nodes in the region.

In this section we discuss four methods for estimating a node's local contribution to a region's area. The first two rely on well-known algorithms from computational geometry: the Delaunay triangulation and its dual, the Voronoi diagram. The last two rely on neighborhood information to estimate local node density.

We discuss the potential communication and computation costs for each technique.

4.1. Delaunay Triangulation

Both computational geometry techniques partition the space covered by the sensor network into disjoint sectors. Then the area of a sensed region equals the sum of the areas of the sectors determined to be in the region. In a *triangulation* of the space, each section is a triangle whose vertices are sensors. A triangle is considered to be in the region if and only if all its sensor-vertices are in the region. This method will underestimate most regions' areas, since it essentially computes the area of a polygon inscribed inside the region. For the best results, we clearly prefer compact triangulations, where the three vertices of each triangle tend to be located close to one another; if the three vertices are unnecessarily far apart, the triangle will tend to be left out of most regions.

The well-known Delaunay triangulation is both the most compact triangulation and by far the most common in the computational geometry literature. Let A, B, and C be vertices in a graph. Then triangle $\triangle ABC$ is in the graph's Delaunay triangulation if no other vertex exists inside the circle passing through A, B, and C. At the very least, Delaunay triangulations require that each sensor know its location and has the facility to beacon that location. Calculating a Delaunay triangulation and its dual, the Voronoi diagram, using the Guibas–Stolfi divide-and-conquer algorithm [8] or Fortune's plane-sweep algorithm [4] have a reasonable time complexity of $O(n \log n)$.

Unfortunately, a correct triangulation might require the locations of every sensor, since adding one sensor could change every existing triangle. In practice, such degeneracy will not commonly arise, but information about nodes more than one hop away may be required to get a good result. To demonstrate this, Figure 1 characterizes the accuracy of a triangulation using at most k hops of neighbor information. We simulate a randomly deployed network of 100 nodes in a square field; the nodes have isotropic disk communication, meaning each node can communicate with all nodes within a radius r of its location, and with no nodes outside that radius. Each node calculates an independent triangulation, using only the locations of nodes within a k-hop neighborhood; k is plotted on the x axis. The plot shows how the total area of the triangulation, obtained by summing all triangles' areas, changes as we increase k. Each node i performs a triangulation in its k-hop neighborhood, then adds the areas of all triangles whose lowest-ID vertex is i to the total. The result, the area of the full triangulation, should be about 90% of the area of the square field. The communication radius r is set to achieve two different average node degrees, or equivalently, one-hop-neighborhood sizes. The higher degree plot reaches a point of diminishing returns at around 3 hops. The lower degree plot, however, improves its estimate with each successive hop of location knowledge, out to nine hops. We conclude, therefore, that it is only feasible to use Delaunay-based techniques when graphs are dense.

In the rest of this paper, we will consider Delaunay triangulations that use location information from the whole network. This provides the best case for Delaunay triangulation's performance, since a real Delaunay deployment would probably use less information.

4.2. Voronoi Diagram

Voronoi diagrams, like Delaunay triangulations, partition the sensed space into disjoint sectors. Voronoi diagrams, however, consist of one polygonal sector per sensor; every point in a sector is closer to its sensor than to any other sensor. The application to estimating a sensed region's area is obvious: Add a polygon's area to the sum if and only if its sensor detects the phenomenon under study.

Area calculation using Voronoi diagrams offers several advantages over Delaunay triangulation. First, the result is likely to be more accurate; a Voronoi covering, unlike Delaunay triangulation, has no tendency towards underestimation. Second, assuming the dia-

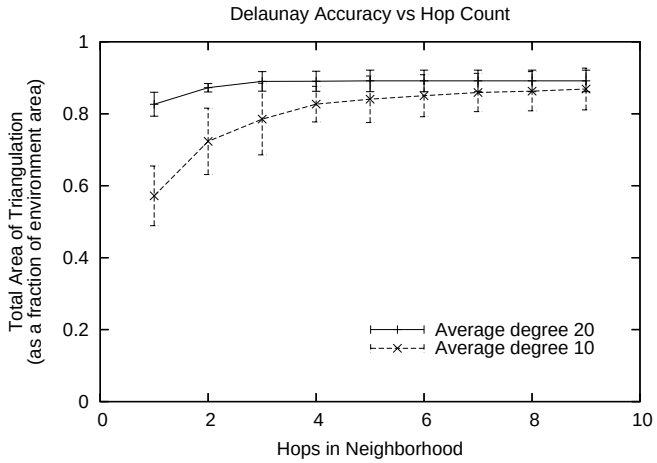


Figure 1. Effects of incomplete location information on Delaunay area estimates. 100 randomly distributed nodes with isotropic disk communication. Average of 5 trials.

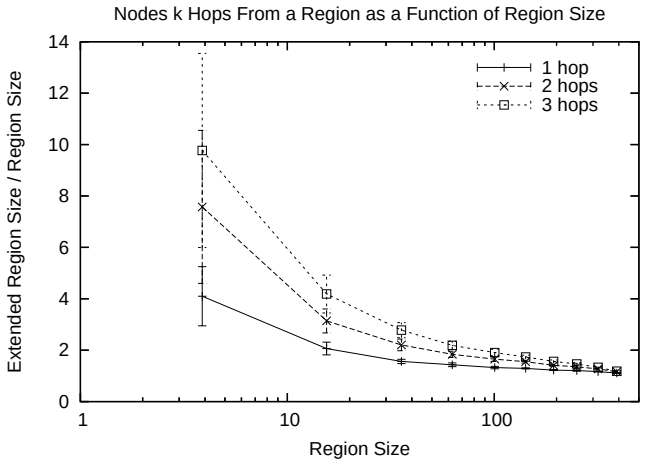


Figure 2. The ratio of sensor count in a region extended by 1, 2, or 3 hops to sensor count in the original region. 500 node random topology in a 1000×1000 network; isotropic disk communication with radius 50.

gram is precomputed, each sensor in the region instantly knows how much area to contribute to the total. Contrast this to the Delaunay technique, where a triangle's area is included only if all three of its vertices are in the region.

The drawback relative to Delaunay triangulation is that where computing area from a triangulation only requires communication between nodes in the region, the polygonization technique requires communication with sensors at least k hops out of the region, to calculate the correct polygon sizes for sensors on the region's border. As discussed earlier, small k may lead to triangulation errors, and therefore Voronoi diagram errors. To demonstrate this disadvantage, Figure 2 shows, for a range of region sizes, the ratio of the number of sensors in an extended region (the region plus 1–3 hops) to the number of sensors in the original region.

4.3. Inverse Neighborhood

Put off by the high costs of these computational geometry techniques, we investigated a much simpler technique that trades less accuracy for less computation. This area estimator relies on a derivation of local density. Assume that a node can guess its coverage area (the area over which it can hear all nodes that are transmitting), and that it keeps track of its 1-hop neighbors (the nodes in that coverage area). It can then estimate the area for which it is responsible by dividing the coverage area by its neighbor count plus one. We call this technique *inverse neighborhood*.

Clearly, inverse neighborhood has significantly less communication and computation overhead than the computational geometry techniques. The sensors must simply keep track of their one-hop neighbors, either through periodic beaconing or through beaconing triggered by a sensed event. The problem is that if the guess of coverage area is bad, the area estimate is bad. Even with a bad estimate of coverage area, however, the technique can still be useful for comparing the relative sizes of two or more regions.

How can we estimate the right coverage area? In the simpler technique, we simply pick a number. Perhaps the radios are configured pre-deployment to transmit robustly within a user-specified range; the area of that range is used for the coverage area. Of all our techniques, only this simple inverse neighborhood measurement does not require that nodes know their locations.

In addition to simple inverse neighborhood, we consider a variant called *inverse neighborhood with location*, where the nodes know their own locations and the locations of their one-hop neighbors. Then each node can make an estimate of its deployed coverage area. For instance, supposing isotropic disk communication, the radius of the coverage disk can be approximated with the distance to the farthest neighbor. Figure 3 shows how this technique can incorrectly calculate the area due to a given sensor. However, while individual nodes can

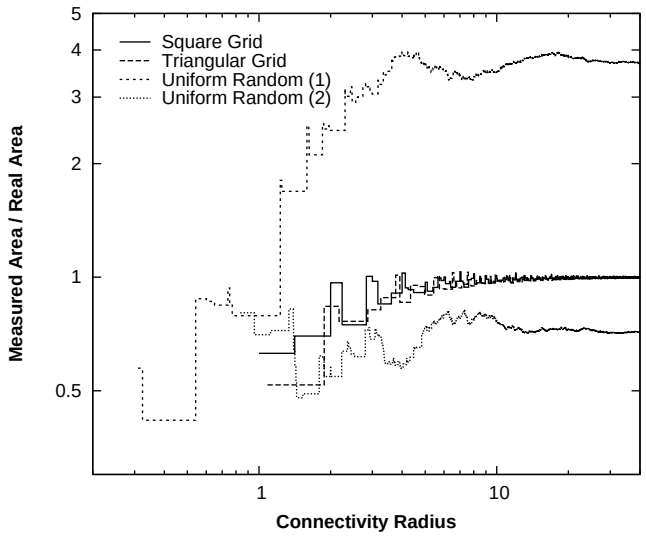


Figure 3. Ratio of a node's area calculated by located inverse neighborhood vs. its true area, calculated by Voronoi polygonalization. Disk connectivity with various radii; various topologies, all with the same density. In "Uniform Random (1)", the chosen node was in a dense part of the graph, leading to area overestimation at large connectivity radius; in "Uniform Random (2)", it was in a sparse part of the graph.

have serious error, the sum of all nodes will tend towards less error; nodes in sparse portions of the graph will underestimate their area, but nodes in dense portions of the graph will overestimate their area.

Disk connectivity is not a good assumption, however. Real world propagation over wireless channels is unpredictable and usually only very loosely follows the Raleign and/or Rician models. The conditions of the deployment environment can make the coverage area bigger or smaller; change its shape; change its signal-to-noise ratio nonmonotonically with distance; introduce connectivity holes; and so forth. We have found in practice that coverage area is usually less than that of a disk with radius equal to the distance to the farthest neighbor.

5. Results

We investigate the performance of the inverse neighborhood and computational geometry techniques under various node densities, homogenous region sizes, and communication areas. We present two sets of simulations; the first uses an isotropic disk model of communication while the second uses real-world connectivity data taken from a 55-node testbed.

We expected the computational geometry techniques to handily beat our primitive inverse neighborhood algorithms. Instead, to our surprise, inverse neighborhood handily beat the computational geometry algorithms on the simulated topology. This result should be interpreted not as a commendation of inverse neighborhood, but as an indictment of the simplistic random topology/isotropic disk connectivity model, which, though commonly used, is quite far from reality. In simulations using real-world connectivity, the computational geometry mechanisms perform far better, but located inverse neighborhood continues to do almost as well. We conclude that a variant of located inverse neighborhood may be sufficient for real sensor-network area measurements.

5.1. Methodology

In the first set of experiments, node locations were chosen uniformly and at random over a 1000×1000 network. Five topologies were generated. Since communication follows a disk model, a particular average node degree (or, equivalently, one-hop neighborhood size) could be set by adjustment of the disk radius. The disk radii corresponding to average degrees of 5, 10, 15, 20, and 25 were generated for 100 node topologies and average degrees of 25, 50, 75, 100, and 125 were generated for 500 node topologies. In our simulations, we compute the area of circular regions. Regions were given radii of 50, 100, 150, 200, 250, 300, 350, and 400, and were located randomly in the network. Regions overlapping network boundaries were disqualified from consideration.

The inverse neighborhood procedure requires an estimate of the communication area and a count of neighbors. For the unit disk simulations without location information, the communication area is set to the area of the communication disk. For the simulations without location information that are based on experimentally derived connectivity data, the communication area is set to the area of a circle with a radius equal to the average distance from a node to its farthest neighbor.

When location information is available, in the disk connectivity case, the communication area is set to the area of a circle with a radius equal to the distance from a node to its farthest neighbor. In the case using real-world data, the communication area is set to the area of a circle with a radius equal to the distance of the neighbor that is at the 75th percentile of distances from a node. The 75th percentile was found experimentally to be a reasonable point in the distribution to charac-

terize the effective communication area of a real-world MICA transmitting indoors.

Future work might apply similar empirically-derived constant factors to the results of the computational geometry techniques. This might improve their already-good accuracy. Our goal here is different: we are looking for simple ways to get better accuracy from computationally-inexpensive mechanisms for area computation. Inverse neighborhood won't usually outperform the computational geometry techniques in terms of accuracy, but it will perform reasonably well while remaining simple in terms of computational and message complexity.

The Delaunay triangulation of a region was performed assuming complete knowledge of the locations of all nodes in a region by all the nodes in that region. Likewise, the Voronoi diagram is computed assuming complete knowledge of the locations of all nodes in the region under evaluation, and the locations of any nodes outside the region's border that are necessary to enclose every region node with its Voronoi polygon.

5.2. Random Topology, Isotropic Disk Connectivity

Figures 4(a) and 4(b) compare our four techniques for networks of 100 and 500 uniformly randomly placed nodes, respectively, using isotropic disk connectivity. The radius of the region of interest varies from 200 to 400. The expected number of nodes per region is given by this table:

	100 nodes	**500 nodes**
Radius 200	12.6	62.8
Radius 400	50.3	251.3

In these simulations, all nodes have equal transmission radii. Even without location information, the unrealistic connectivity model helps the inverse neighborhood technique produce accurate results. So long as the communication disk size is known and static across all nodes, inverse neighborhood performs with error below 15 percent. This is because the average density over the area of the disk is known precisely, and, consequently, so is the average contributing area of any node in the disk.

The Delaunay triangulations are computed assuming each node has knowledge of every region node's location. Triangulation performs well only when there are many nodes in the region of interest. This is to be expected, since each included triangle must have all three vertices in the region; when a region contains only 12 nodes, this leaves a lot of data out. Voronoi polygonization performs well across region sizes, but tends to overestimate area in these simulations.

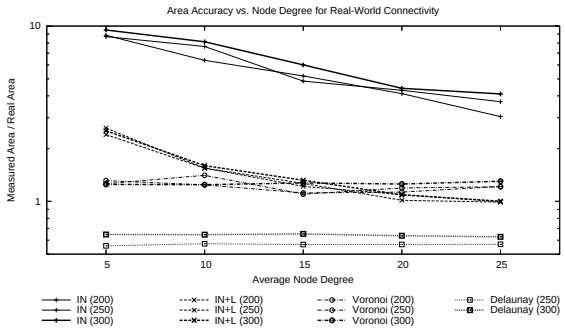


Figure 5. Accuracy in area calculation as a function of node density for various circular regions of interest. Real-world topology and connectivity data from our testbed. Averages of 5 trials.

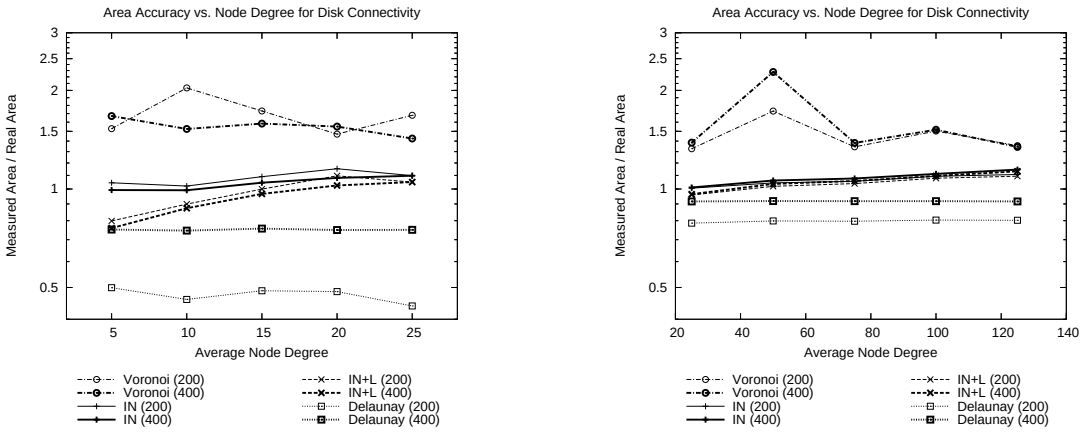
5.3. Real-World Topology and Connectivity

Disk connectivity is not realistic. Real-world connectivity is probabilistic and noisy, and although it generally decreases with d^2 to d^4, obstacles and interference make it very hard to predict. It is nonisotropic. Rather than attempt to more accurately model the communication channel (which is a serious research effort unto itself), we use real connectivity data from a 55-node MICA testbed with TR1000 radios [1], configured to cover a 1000×1000-unit area. The radius of the region of interest varies from 200 to 300 units. The expected number of nodes per region is given by this table:

Radius	**200**	**250**	**300**
Nodes in Region	6.9	10.8	15.6

The connectivity data was collected using the following procedure: Each node in the topology sent 200 TOS_Msg packets and the percentage of packets received was collected for all pairs. This experiment was repeated for various radio transmission powers. We set a threshold percentage, such that two nodes A and B are considered neighbors of each other if the minimum reception percentage of A to B and B to A is greater than or equal to that threshold. We vary the threshold to generate topologies with varying average degrees of connectivity. Each point in our results represents the average of trials with different power levels and thresholds, but the same average node degree.

Figure 5 shows the results, which differ dramatically from the random topology/disk connectivity model. Inverse neighborhood without location information does particularly poorly. Recall that we assign connectivity area based on a circle whose radius equals the aver-



(a) 100 nodes total

(b) 500 nodes total

Figure 4. Accuracy in area calculation as a function of node density for various circular regions of interest. Uniform random topology, isotropic disk connectivity. "IN" is inverse neighborhood, "IN+L" is located inverse neighborhood. The numbers in parentheses are the radii of the circular regions of interest. Averages of 5 trials.

age longest distance between neighbors over the whole network. This is a wild overestimate, since real connectivity area is not a disk; for instance, nodes might have much greater connectivity in a preferred direction than in any other direction. This overestimate will be worse at lower power rates or higher loss thresholds—for instance, connectivity in the preferred direction might be preserved, keeping the connectivity radius high, even as connectivity in other directions drops, reducing the connectivity area. Thus, the error in area is worse at lower connectivity; but even at high connectivity, it is a factor of three too high.

Inverse neighborhood with location does much better. Calculating connectivity area on a per-node basis and using the 75th-percentile neighbor distance as a connectivity radius both filters out topological effects, and reduces overestimation caused by unequal connectivity. Given this good estimator of communication range, located inverse neighborhood performs comparably to the computational geometry techniques. Consider the plots in which the region radius is 250. Again, inverse neighborhood's overestimation problem is worse at lower node degrees, and at average node degree 5, the inverse neighborhood technique yields an area that is 2.6 times the real area on average. However, as node degree increases so too does area accuracy. At degree 15, the area is only 1.26 times the true area; at degree 20, it is almost exactly the true area, with an overestimation factor of just 1.01. Voronoi polygonalization, our most theoretically precise technique, does appreciably better only for node degrees below 20. One advantage of the Voronoi technique is that its accuracy is independent of node degree; the overestimation factor hovers around 1.2 regardless of degree.

The lowest lines are Delaunay triangulations. Delaunay triangulation of nodes within a region only performs well when there are many nodes in the region. In a 55-node network, this is never the case.

5.4. Larger Regions

Figure 6 depicts the influence of region area on accuracy in simulations with real-world connectivity. We constructed a 378-node graph from our 55-node grid experimental results using the following procedure: First, we take a 6×7 node rectangular subset of the 55 nodes and create nine 42-node tiles, which we place in a 3×3 arrangement. Connectivity between any of the 42 nodes of a single tile is determined by the original experimentally derived measurements. Connectivity between nodes on adjacent tiles is assigned by mirroring connectivity from within a tile. For example, if node A is near the edge of a tile and node B is 1 meter in from the edge, then the connectivity between A and B would also be assigned to a node C that is in an adjacent tile and is 1 meter from A. Specifically, mirroring is performed about the x-axis and about the y-axis.

As expected, the Delaunay triangulation technique

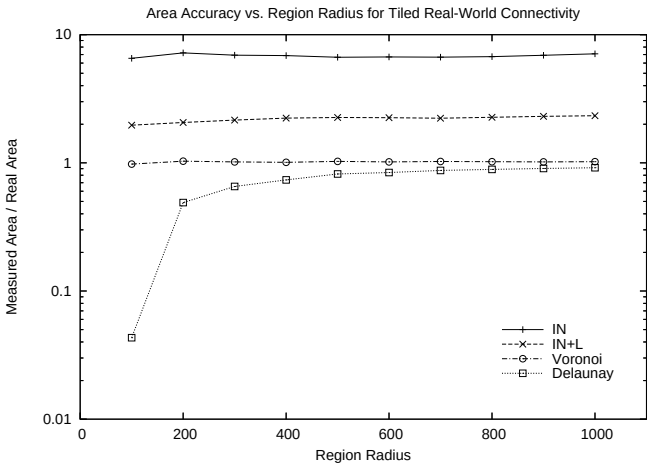


Figure 6. Accuracy in area calculation as a function of region radius. Real-world connectivity data, extended to a larger size through tiling and mirroring. Average node degree 25.

significantly improves its estimate as the number of nodes in a region increases. The other techniques' estimate accuracies do not change significantly with region area.

6. Conclusions

In this paper, we investigated four algorithms for the distributed computation of a region's area.

Delaunay triangulation and Voronoi polygonization are expensive in terms of communication cost, because computation usually requires more than a single hop of location knowledge. Furthermore, the $\Omega(n \log n)$ algorithms used involve floating point arithmetic and extensive manipulations of complicated data structures; both could strain a mote.

We presented two simple inverse neighborhood algorithms, which use the inverse of local density, computed by taking the communication area of a node and dividing it by the number of neighbors the node has, to estimate a node's covering area. Without location information, inverse neighborhood can only give a very coarse indication of node area, but can be used to determine the relative sizes of multiple phenomena. With location information, inverse neighborhoood performs well both under the unrealistic assumption of disk connectivity as well as with real connectivity data. In fact, and surprisingly, located inverse neighborhood can perform comparably to the best of the hugely more complex computational geometry algorithms.

We conclude that variants of located inverse neighborhood are sufficiently precise and robust for practical use as inexpensive, distributed area estimators for sensor networks.

We also ran experiments on two topologies, one derived from common assumptions (uniformly randomly distributed nodes with isotropic disk connectivity) and one derived from real-world data. The results, while not shocking, are significant: The real-world-based experiments produced results dramatically different from random topologies with disk connectivity. We recommend the avoidance of such random topologies except in exceptional circumstances.

7. Future Work

One strain of future work would be to move towards even more realistic connectivity assumptions to evaluate the effects of non-uniform deployment densities on accuracy; to study how location error and correlation in location error contributes to area error; and to study operation with non-circular regions of interest.

It may be possible to improve the performance of our inverse neighborhood algorithms without adding significant complexity. For example, the tendency to overestimate connectivity area might be addressed by changing the threshold for connectivity radius, or by moving away from the circular-connectivity assumption.

We plan also to evaluate the performance of our algorithms, and Welsh's planar mesh code, when running on mote hardware. In particular, we seek to derive bounds on minimum detection latency and on duty period. Given the constrained resources of the target platform, it would be beneficial to carefully characterize the performance of Voronoi polygonization, Delaunay triangulation, and inverse neighborhood in terms of CPU cycles expended and bits transmitted.

References

[1] A. Cerpa, N. Busek, D. Estrin, SCALE: A Tool for Simple Connectivity Assessment in Lossy Environments. In *CENS Technical Report 0021*, September, 2003.

[2] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, Habitat monitoring: Application driver for wireless communications technology. In *ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, 2001.

[3] K. Chintalapudi, R. Govindan, Localized Edge Detection in Wireless Sensor Networks. In *Proceedings of the IEEE ICC Workshop on Sensor Network Protocols and Applications*, April 2003.

[4] S. Fortune, A Sweepline Algorithm for Voronoi Diagrams. In *Algorithmica*, 2:153-174, 1987.

[5] S. Ganeriwal, C. Han, M. B., Srivastava, Going beyond nodal aggregates: Spatial average of a continuous physical process in sensor networks. In *Poster in Sensys*, 2003.

[6] D. Ganesan, D. Estrin., and J. Heidemann, DIMENSIONS: Why do we need a new Data Handling architecture for Sensor Networks? In *Proceedings of the First Workshop on Hot Topics In Networks (HotNets-I)*, Princeton, New Jersey. October 28-29 2002.

[7] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, S. Shenker, DIFS: A Distributed Index for Features in Sensor Networks. In *Proceedings of First IEEE International Workshop on Sensor Network Protocols and Applications*, Anchorage, AK. May 2003

[8] L. Guibas and J. Stolfi, Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams. In *ACM Transactions on Graphics*, 4:74-123, April, 1985.

[9] G. Leach, Improving Worst-case optimal Delaunay triangulation algorithms. In *Fourth Canadian Conference on Computational Geometry*, 1992.

[10] X. Li, Y. J. Kim, R. Govindan, W. Hong, Multidimensional Range Queries in Sensor Networks. In *Proceedings of the ACM Sensys*, November 2003.

[11] S. Madden, M. Franklin, J. Hellerstein, and W. Hong., Tag: a tiny aggregation service for ad-hoc sensor networks. In *OSDI*, Boston, MA, 2002.

[12] S. Meguerdichian, F. Koushanfar, G. Qu and M. Potkonjak, Exposure in wireless Ad-Hoc sensor networks. In *ACM Mobile Computing and Networking*, 2001.

[13] R. Nowak and U. Mitra, Boundary Estimation in Sensor Networks: Theory and Methods. In *2nd International Workshop on Information Processing in Sensor Networks*, Palo Alto, CA, April 22-23, 2003.

[14] J. Polastre, Design and Implementation of Wireless Sensor Networks for Habitat Monitoring. Master's Thesis, University of California at Berkeley, Spring 2003.

[15] G. Pottie and W. Kaiser, Wireless integrated network sensors. In *Communications of the ACM*, 2000.

[16] T. Schoellhammer, Distributed Pattern Matching in Sensor Networks. Poster presented at CENS Opening Ceremony, Los Angeles, October 2002.

[17] M. Welsh, Exposing Resource Tradeoffs in Region-Based Communicaton Abstractions for Sensor Networks. In *Proceedings of the 2nd ACM Workshop on Hot Topics in Networks (HotNets-II)*, November 2003.