

City University of New York (CUNY)

CUNY Academic Works

Computer Science Technical Reports

CUNY Academic Works

2006

TR-2006014: Geometric Considerations for Distribution of Sensors in Ad-hoc Sensor Networks

Ted Brown

Deniz Sarioz

Amotz Bar-Noy

Tom LaPorta

Dinesh Verma

See next page for additional authors

[How does access to this work benefit you? Let us know!](#)

More information about this work at: https://academicworks.cuny.edu/gc_cs_tr/280

Discover additional works at: <https://academicworks.cuny.edu>

This work is made publicly available by the City University of New York (CUNY).
Contact: AcademicWorks@cuny.edu

Authors

Ted Brown, Deniz Sarioz, Amotz Bar-Noy, Tom LaPorta, Dinesh Verma, Matthew Johnson, and Hosam Rowaihy

Geometric Considerations for Distribution of Sensors in Ad-hoc Sensor Networks

Ted Brown,* Deniz Sarioz,*
Amotz Bar-Noy*, Tom LaPorta†, Dinesh Verma‡,
Matthew Johnson*, Hosam Rowaihy†

November 20, 2006

1 Introduction

Ad-hoc sensor networks are networks of sensing devices (and other communicating devices) whose goal is to have the sensors “sense” some aspect of their local environment and communicate those values over the network they have created. These type of networks create a number of challenging problems for the following reasons.

The sensors that are part of an ad-hoc sensor network are most often *low-cost battery powered* devices, capable of sensing some aspect of the environment *near* them. For example, low frequency vibrations, visual light, sound, infrared light, motion, etc. Their batteries have a short life-time and low power. Consequently the sensors they power are severely limited in the range that they can sense and cannot communicate their values far. Furthermore, in typical applications, sensors cannot be positioned accurately. For example, they may be dropped from a plane. The network they use to move information is *ad-hoc*, in that the network needs to be created using the sensors (and other communication devices). The limited power of sensors means that sensor networks may have to make use of multiple hops within their ad-hoc network to move the information they have gathered to a collection point. What is more, to conserve power one may want to turn the power off on a sensor. The communication network may need to be created on the fly as need be as sensors previously used as relay points may be turned off or have died. Additionally, redundancy of sensor coverage in a part or whole of the area may be important so that if a sensor uses up its power, the area it was sensing is still covered by another. All of these difficulties

*The City University of New York Graduate School and University Center, the Ph. D. Program in Computer Science, New York, NY, 10016. Please direct all correspondence via email to Deniz Sarioz, sarioz@acm.org

†The Pennsylvania State University, Computer Science and Engineering Department, University Park, PA, 16802

‡IBM Watson Research Center (Hawthorne), Yorktown Heights, NY, 10598

make studying these networks an interesting topic and there have been many papers in the recent past on examining solutions to these problems.

We examine a coverage problem. There are really two related coverage questions for ad-hoc sensor networks, *sensing coverage* of the area, and *communication coverage*. In this paper, we will investigate some geometric considerations for the placement of sensors in the area so as to provide sensing coverage and redundancy. While in fact, a single sensor may have several sensory modalities (low frequency vibrations, visual light, etc.), with each modality differing in range and in other properties, we will limit ourselves in this paper to one type of sensing coverage and make a number of assumptions: all sensors have the same modality, all sensors have the same radius of coverage (with perfect coverage inside of it), the field that the sensors are to be placed in is flat, with no obstructions, and the communications radius of a sensor is more (maybe two or three times more) than the sensor coverage radius. (The last assumption means that sensing coverage also provides communications coverage.)

Investigations of sensing coverage of a field has been well studied as well. In the sensor literature an approach that has been discussed is iteratively finding a better coverage of the area than the current configuration. Although it is possible to provide global information to all sensors, this is power and time expensive. Consequently, a useful and usual assumption is to assume sensors only know about their local environment based on how far they can communicate. The algorithms at each iterative step then move the sensors to improve their own coverage of the area they see. Methods involving local computations [4] have been shown to converge to local optima, resulting in very good but not perfect coverage. In the scenario where sensors malfunction and need to be replaced, a bidding-based algorithm [3] has been suggested. For the same task, when there is a surplus of sensors a grid-based method [5] has been suggested wherein sensors designated as neighborhood heads communicate with one another to calculate routing of sensors from neighborhoods with surplus sensors to those requiring them.

Some work too has been done on proper placement without considering moving sensors. Using probabilistic modeling in a random deployment scenario with non-mobile sensors [6] calculates efficient placement of sensors for full coverage. However, the problem of finding good placement and redundant coverage has not been studied.

In this paper we use geometric ideas to look at what would be considered a “good” placement of sensors. By placing sensors into a grid configuration we do not attempt an optimal configuration but one that is easy to deploy and has excellent coverage properties. The results from here can be used as a criterion for the iterative algorithms for movable sensors, or for placing sensors that cannot move.

2 Geometric approach

Well studied for hundreds of years are how best to cover and or pack a two dimensional area using various geometric shapes, including disks (circles with their interiors) into a square area. Solutions have been given by mathematicians, amateur puzzle solvers, and more recently computer scientists [1]. In this literature, the covering problem would be stated as covering completely a square area using disks. As stated in a collection of open problems in discrete geometry [1], only the arrangements of n unit disks that cover the largest possible square have been proved for small values of n .

The problem remains without a general solution. The exact solutions for small n do not provide guidance for a general solution and because they are not totally symmetric, do not result in optimal solutions for larger areas when repeated. Our interest here is to find a solution that will have symmetry, be general in its approach (i.e., not depend on n), and be easy to implement. We are also interested in covering solutions that are redundant.

The paper is organized by first looking at how to use repeating patterns to cover a field with 2 redundancy. We then consider the of k -covering, that is, making each point in the field covered by at least k disks. Multiple covering of an area is important since redundancy allows some sensors to be turned off or malfunction and still cover an area of interest. Redundancy can be important when triangulation is of concern, when occlusion occurs, and it allows 3-D images of events. Some of the earlier iterative algorithms could be improved using these results.

We examine the problem from a geometric point of view. We will use the term “disk” to represent the area that a sensor covers, which is appropriate based on our assumption that all points up to a distance r from a sensor are covered by it. Without loss of generality, we will consider only unit disks, that is, having radius 1.

2.1 Exact solutions with lattice-based arrangements

A field can be covered with disks in a variety of ways. We describe one way that has some advantages described below. The method we shall use is to place the (sensors) center of disks evenly around the circumference of another (sensor) disk. We illustrate using first showing a solution with four disks around the circumference, then three. Together they show the general method. It turns out that the three disk solution is optimal.

2.1.1 Four disk overlap solution

We illustrate the placement algorithm using four disks around the circumference of another. Start with a single disk within a large field. Four disks are added, one centered at 0° on this disk’s circumference, the next at 90° , then 180° , and another at 270° (0° point is arbitrarily chosen.) Fig. 1 (a) shows this basic arrangement, and Fig. 1 (b) the resulting pattern. Notice that every point on

the plane, except the edges, is covered at least twice using this arrangement. (We will ignore the boundary conditions in the rest of this paper, since we assume a large field. They are easily handled, e.g. have disks extend beyond the boundary).

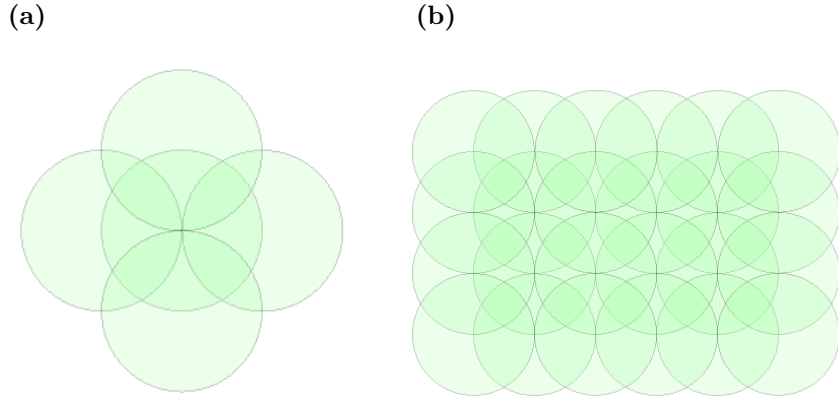


Figure 1: “Four disk overlap” solution

Observe that if the field were infinite and if the origin is placed at any disk, the disk centers fall on exactly the points on the plane with integer coordinates: $\{(i, j) \mid i, j \in \mathbb{Z}\}$. This is sometimes called the integer lattice on the plane.

A *lattice arrangement* [2] on the plane, also called a *grid lattice*, is the set of all points whose coordinates are a linear combination with integer coefficients of two linearly independent vectors $\mathbf{u} = \langle 1, 0 \rangle$ and $\mathbf{v} = \langle 0, 1 \rangle$. Such a set $\{\mathbf{u}, \mathbf{v}\}$ is called a *basis*, and we shall refer to these vectors as *basis vectors* that *generate* a grid lattice Λ . In precise terms, $\Lambda(\mathbf{u}, \mathbf{v}) = \{i\mathbf{u} + j\mathbf{v} \mid i, j \in \mathbb{Z}\}$.

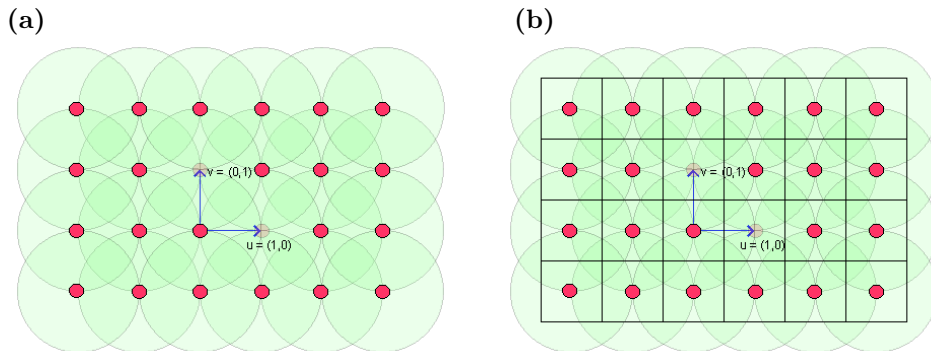


Figure 2: The square grid (a) with basis, (b) with Voronoi neighborhoods

In this language, the integer lattice obtained by placing four sensors on the

circumference of another sensor’s circumference is the grid lattice generated by the basis vectors $\langle 1, 0 \rangle$ and $\langle 0, 1 \rangle$, which is shown more clearly in Fig. 2 (a).

The integer lattice \mathbb{Z}^2 can be decomposed into two square grid lattices based on parity, much of how a checkerboard consists of black squares and white squares. In other words, $\Lambda_{even} = \{(i, j) \mid i + j = 2k : i, j, k \in \mathbb{Z}\}$ and $\Lambda_{odd} = \{(i, j) \mid i + j = 2k + 1 : i, j, k \in \mathbb{Z}\}$. Clearly, $\mathbb{Z}^2 = \Lambda_{even} \uplus \Lambda_{odd}$ where \uplus denotes *disjoint union*.

Notice that the set of unit disks centered at Λ_{even} and that centered at Λ_{odd} can each cover the entire field on their own—thus together they create a 2-cover, a desirable decomposition.

In order to estimate how many sensors will be needed to cover an area with sensors in a certain arrangement, we can find the *coverage density*. The coverage density is the ratio of the sum of the individual disks’ areas to the area of the field being covered. Consider the decomposition of the field into the Voronoi neighborhoods of the sensors. The Voronoi neighborhood of a sensor p is the set of points no closer to any other sensor q than to p . Due to the regularity of the integer lattice (and in fact, all grid lattices), these Voronoi neighborhoods are all of equal size; their boundaries are shown in Fig. 2 (b). Therefore, the coverage density will be exactly the area of a disk divided by the area of its Voronoi neighborhood.

Since a unit disk has area π and each Voronoi neighborhood for this solution is a square with length 1, and has area exactly 1, the coverage density of this solution is π . Clearly, the coverage density is independent of the actual sensory radius r , since every solution can be scaled by r . Using this method, we can cover a large (so that edge conditions can be neglected) field of area A with approximately $\pi A/r^2$ or $3.14A/r^2$ sensors.

2.1.2 Three disk overlap solution

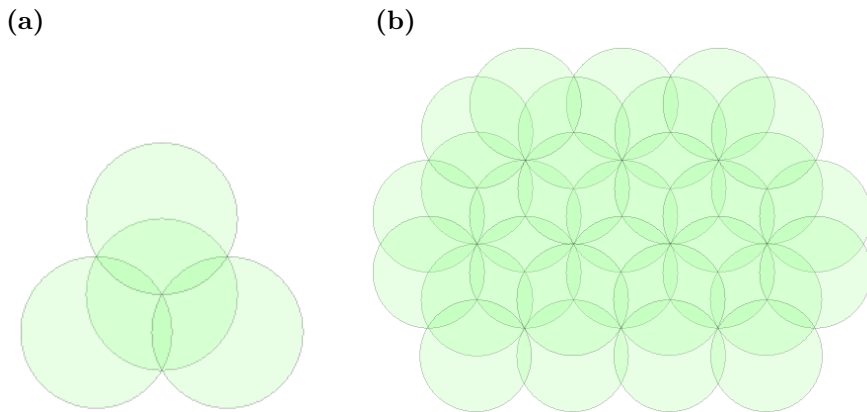
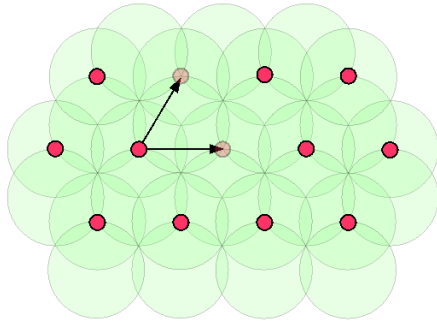


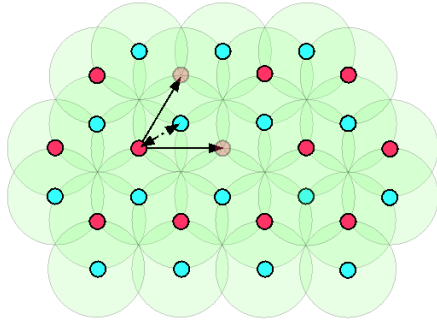
Figure 3: “Three disk overlap” solution

In the same way we can create a grid pattern using three disks around the circumference of another. Starting with a single disk within the field, three disks are added, one centered at an arbitrarily chosen 0° point on its circumference, the next at 120° , and another at 240° . Fig. 3 (a) shows this basic arrangement. And for each neighbor of the central disk, add (if not there already) disks centered around their circumference, this time at 180° , -60° , and 60° . The resulting pattern is shown in Fig. 3 (b). Notice that every point on the plane (except at the boundaries) is like the four disk solution covered at least twice using this arrangement.

(a)



(b)



(c)

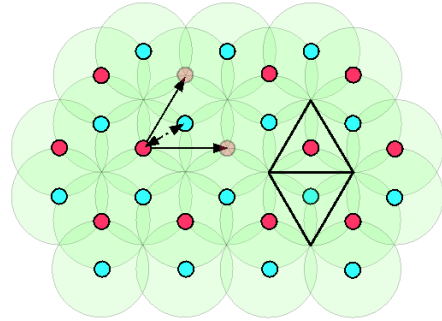


Figure 4: “Three disk overlap” solution with grid lattice

The disk centers constitute what is sometimes called the hexagonal grid, which we shall denote as Λ_{hex} , having a honeycomb-like structure, perhaps seen more clearly by the small light and dark circles at the centers of disks in Fig. 4 (b).

This arrangement is not a grid lattice in the sense defined in the previous section, since it does not admit a characterization as being simply the linear combination with integer coefficients of some basis vectors. However, it can be seen that the subset of this grid that is only the dark circles does indeed constitute a grid lattice with the basis vectors, which is shown in Fig. 4 (a).

More specifically, $\Lambda_{dark} = \Lambda(\mathbf{u}, \mathbf{v})$ where $\mathbf{u} = \langle \sqrt{3}, 0 \rangle$ and $\mathbf{v} = \langle \sqrt{3}/2, 3/2 \rangle$. Notice that both basis vectors as well as their difference $\mathbf{v} - \mathbf{u}$ have magnitude $\sqrt{3}$. This is commonly referred to as the (equilateral) triangular lattice, as the origin and the two heads of the basis vectors drawn with tails at the origin demarcate an equilateral triangle.

It is clear that $\Lambda_{hex} = \Lambda_{dark} \uplus \Lambda_{light}$, which renders Λ_{hex} the double lattice arrangement [2]. In addition, just the sensors placed on Λ_{dark} can cover the entire field, as well as the sensors placed on Λ_{light} . In fact, the arrangement of sensors on either are optimal for covering a field [1, 2] in terms of density.

Like in the previous section, we can determine the coverage density of disks in this arrangement with the ratio of the area π of a unit disk, to the area of the Voronoi neighborhood of a disk center. The Voronoi neighborhoods in the hexagonal grid are bounded by congruent equilateral triangles, where those around the sensors in Λ_{dark} are vertical mirror images of those around the sensors in Λ_{light} , an example for each of which is shown in Fig. 4 (c). Further observe that the sides of these equilateral triangles have the same length as the basis vectors generating Λ_{dark} . Each has a base of length $\sqrt{3}$ and height of length $3/2$, yielding an area of $3\sqrt{3}/4$. Hence, the density here is $\pi/(3\sqrt{3}/4) = 4\pi/(3\sqrt{3}) = 4\pi\sqrt{3}/9$, approximately 2.418.

This means that to cover a field of large area A (so that edges are of no concern) with sensors of radius r using this arrangement would take approximately $4\pi A\sqrt{3}/(9r^2)$ or $2.42A/r^2$ sensors. This is a factor of 1.3 more efficient compared to the arrangement described in the previous part.

In fact, since there exists a decomposition into two lattice arrangements, each of which is known to cover the field optimally, this is the best arrangement in which every point in the field is covered at least twice, and is decomposable into two lattice arrangements each of which covers the field.

The arrangements presented in these subsections can be thought of as placing m sensors on the sensing circumference of one, $360^\circ/m$ apart from one another. We explained the results for $m = 4$ and $m = 3$, in that order. A value of $m = 6$ would result in a triangular grid, similar to what is shown in Fig. 4 (a), but with disk centers closer together. It can be shown that no repeating 2-dimensional pattern is possible for values of m other than 3, 4, and 6.

2.2 k -covering

A k -covering of a field is an arrangement of sensors such that every point of the field can be sensed by *at least* k sensors. Every covering is a 1-covering. The two arrangements presented in the previous subsections are 2-coverings.

The primary objective is to k -cover at the same time minimizing the number of sensors needed. Another desirable property is for *all* the sensors to be spaced as far apart from one another as possible. This is subtly different from the primary objective. We deal with the second here. In general, we can take the triangular grid in Fig. 4 (a) which provides an optimal 1-covering, and superimpose k copies of it, one on top of another, so that every position would have k sensors. Clearly, this would be a legitimate k -covering. Moreover, this is a

decomposable covering. However, this would not satisfy the secondary objective since many sensors would have to be piled at exactly the same location!

We can see that the 2-covering solution provided in Fig. 4 (b) satisfies the secondary criterion well by placing sensors at the centers of the equilateral triangles defined by the basis vectors.

The problem of given a general k , how best to obtain a decomposable k -covering such that the secondary criterion is also satisfied remains to be investigated.

References

- [1] P. Brass, W. O. J. Moser, and J. Pach. *Research Problems in Discrete Geometry*. Springer, Berlin-Heidelberg, 2005.
- [2] J. Pach and P. K. Agarwal. *Combinatorial Geometry*. Wiley-Interscience, New York, 1995.
- [3] G. Wang, G. Cao, and T. F. LaPorta. A bidding protocol for deploying mobile sensors. In *Proc. IEEE ICNP*, 2003.
- [4] G. Wang, G. Cao, and T. F. LaPorta. Movement-assisted sensor deployment. In *Proc. IEEE Infocom*, 2004.
- [5] G. Wang, G. Cao, T. F. LaPorta, and W. Zhang. Sensor relocation in mobile sensor networks. In *Proc. IEEE Infocom*, 2005.
- [6] Y. Zou and K. Chakrabarty. Uncertainty-aware and coverage-oriented deployment for sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):788–798, 2004.