

On Optimal Arrangements of Binary Sensors

Parvin Asadzadeh, Lars Kulik, Egemen Tanin and Anthony Wirth

National ICT Australia (NICTA), Department of Computer Science and Software Engineering, University of Melbourne, Parkville, Victoria 3010, Australia

Abstract. A large range of monitoring applications can benefit from binary sensor networks. Binary sensors can detect the presence or absence of a particular target in their sensing regions. They can be used to partition a monitored area and provide localization functionality. If many of these sensors are deployed to monitor an area, the area is partitioned into sub-regions: each sub-region is characterized by the sensors detecting targets within it. We aim to maximize the number of unique, distinguishable sub-regions. Our goal is an optimal placement of both omni-directional and directional static binary sensors. We compute an upper bound on the number of unique sub-regions, which grows quadratically with respect to the number of sensors. In particular, we propose arrangements of sensors within a monitored area whose number of unique sub-regions is asymptotically equivalent to the upper bound.

1 Introduction

Geo-sensor networks generate large interest from researchers in spatial information science. They are used to detect, monitor and track continuous environmental phenomena such as toxic plumes or oil spills in seas. Although sensor nodes are usually considered to be inexpensive, large deployments still incur significant costs. In addition, even if the number of sensors is small, the cost of the actual deployment of sensors might still be significant. Thus, it is an important strategy to minimize the costs by identifying optimal arrangements of sensor nodes to cover a monitored area. Our work will analyze an important subclass of geo-sensor networks, those that use binary sensors to detect the presence of a phenomenon. Examples of binary sensors include motion sensors that can detect the presence of movement, RFID (Radio Frequency IDentification) readers that can detect the presence of RFID tags [13] and binary chemical sensors that can detect the presence of chemical compounds in their fields.

Binary sensors can be divided into omni-directional and directional binary sensors. Omni-directional binary sensors can detect the presence of a phenomenon from any direction within a specific distance; whereas, directional binary sensors have a limited range and can only determine the presence of a phenomenon within a sector. Generally, binary sensors are beneficial for movement tracking. RFID antennas, for example, can be used in design of gesture based user interfaces, in which movement of RFID-tagged user hands are detected to enable natural user interaction with computing devices. Moreover, directional RFID

antennas can be installed in a museum to track people carrying RFID-enabled devices to enable personalized recommendations of further exhibits. We propose optimal arrangements for both omni-directional and directional binary sensors. The goal of optimal arrangements of binary sensors is to provide the desired accuracy for a large class of applications with reduced cost.

A number of approaches have suggested the use of binary sensor networks to track phenomena or targets in an area [11][1][12][3]. In general, each binary sensor can only return information regarding a target’s presence or absence within its sensing region. However, one positive detection of a target greatly confines its possible locations, since a positive detection indicates that the target is in the space defined by the sensing region of that sensor.

In binary sensor networks, the results of all sensor detections can be combined to provide a more accurate estimation of the whereabouts of a target at any given time. In such a case, the monitored area is divided into multiple sub-regions so that each sub-region is in the sensing regions of a particular set of sensors. We refer to the technique of using binary sensors to partition a space as *space partitioning*.

Figure 1(a) shows partitioning of a monitored area into eight sub-regions r_1 to r_8 , using three omni-directional sensors S_1 , S_2 and S_3 . These sub-regions are distinguishable in the sense that each sub-region is covered by sensing regions of a different set of sensors (Figure 1(b)).

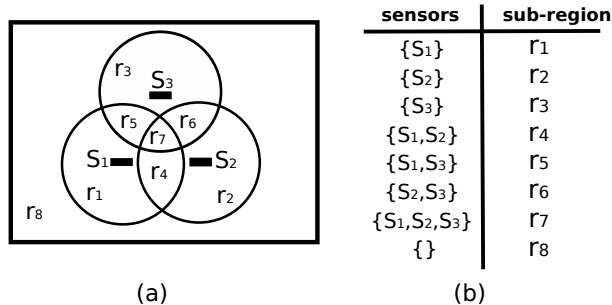


Fig. 1. Sample space partitioning using three omni-directional sensors

In most tracking applications, the sensors are scattered randomly with uniform distribution over a two-dimensional planar monitored area. Shrivastava *et al.* [11] show that for a fixed sensing radius, the accuracy improves linearly with an increasing sensor density. Furthermore, for a fixed number of sensors, the accuracy improves linearly with an increase in the sensing radius because an increase in the sensing radius leads to a finer geometric partition of the field.

Space partitioning using binary sensors can also be used to localize stationary targets in many indoor localization applications. For example, an RFID system could be installed in a library to detect misplacement of books [4]. Medication supply rooms in a hospital can also be equipped with binary sensors to provide

efficiency and security by reducing staff time and frustration in finding what they need faster, and eliminating drug lost or misplacement.

An important goal is to minimize the number of sensors in the monitored area to reduce cost while providing the required accuracy. Therefore, it is essential to find an optimal arrangement of sensors. However, space partitioning does not guarantee that every two sub-regions can be uniquely identified. In Figure 2, for example, the two sub-regions of r_1 and r_2 are in sensing region of the same sensor, i.e., S_3 . Therefore, when a target is detected by sensor S_3 only, it could be in either of the two sub-regions r_1 or r_2 . However, we cannot determine in which of the two sub-regions the target is located. We call the set of created sub-regions that are all distinguishable from each other, *unique sub-regions*.

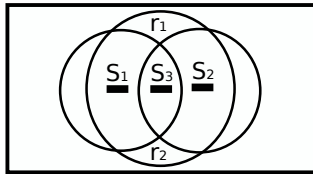


Fig. 2. Sample space partitioning using three omni-directional sensors

Every localization application needs a particular level of accuracy, which determines the resolution by which the monitored area must be partitioned. More precisely, the localization accuracy is limited by the maximum diameter of the created sub-regions. Generally, the number of created sub-regions can be used as an approximate measure to determine the localization accuracy. Since not all sub-regions are distinguishable in target localization, our goal is to maximize the number of unique sub-regions in contrast to the total number of sub-regions.

Our work will (i) provide an upper bound on the number of unique sub-regions a monitored area can be divided into, given a specific number of static binary omni-directional or directional sensors (Sections 5.1 and 6.2). (ii) propose an arrangement of sensors which creates the number of unique sub-regions that is asymptotically equivalent to the calculated upper bound for both omni-directional and directional sensors (Sections 5.2 and 6.3). In our proofs, we assume that the object position can be represented as a point and any value can be assigned to the range and angles of binary sensors. This outcome gives researchers an insight into how many sub-regions can be created using a specific number of sensors as well as the number of sensors required to achieve a certain accuracy.

2 Related Work

Space partitioning using binary sensors have been successfully deployed in a range of indoor applications. Although there are different variants of binary

sensors, they all sense a target’s presence using a physical phenomenon within limited range. Murakita *et al.* [9] have developed a human tracking system, in which floor blocks are fitted with binary pressure sensors. In the Active Badge Location System [14], a network of infrared sensors are placed around a building and detect signals from badges worn by people in order to find the region users are currently located.

Space partitioning has also been successfully applied in many indoor positioning systems using omni-directional RFID readers as binary sensors. In the study by [5], a table surface is equipped with an array of omni-directional RFID antennas and is hence divided into many distinguishable sub-regions. When an object tagged with multiple RFID tags is placed on the table surface, the location of each individual tag is determined by the sub-region containing the tag. Similarly, Bouet and Pujolle [2] as well as Reza and Geok [10] deploy a grid of RFID reader antennas on the floor and ceiling of a building to track RFID tagged objects within that building.

All approaches proposed so far equip the entire monitored area with many low-range omni-directional antennas simply such that the sensing regions of immediate neighbor sensors overlap. However, a comprehensive study that investigates the maximum number of unique sub-regions or finds optimal arrangements of sensors in terms of localization accuracy is still outstanding. Moreover, there is no study of space partitioning using directional sensors, which provide more focused sensor regions than omni-directional sensors.

Mehmood *et al.* [8] employ another variant of space partitioning technique by deploying passive RFID tags in large numbers covering a deployment space. Each RFID tag has an area in which it can be read, which is approximated as a circular disk. In such a deployment, a partition is defined as a non-empty sub-region where a given set of tags can be read by an RFID reader. The location of an agent navigating through the deployment space, is approximated by the closest partition. This approach minimizes the number of used tags for an optimal coverage of space by employing the classical circle covering problem [6]. Kershner [6] has shown that the covering for discs of radius r is optimal when they are placed at the vertices of an equilateral triangular graph overlaying the monitored space. This problem is different to our work because we aim to maximize the number of sub-regions, while in the classical circle covering problem the aim is to cover the whole space with the minimum number of circles.

3 Preliminaries

In this paper, we investigate arrangements of both omni-directional and directional binary sensors. Each omni-directional sensor has a circular sensing region, approximated by a circle in two-dimensional space. Directional sensors, on the other hand, recognize not only the target’s maximal range but also a sector within the circular range around it. Examples of such sensors include cameras, infrared sensors and RFID directional antennas. Generally, the sensing region of a directional sensor can be approximated with a trapezoid in two-dimensional

space (Figure 3(a)). In our proofs, we assume that the object position can be represented as a point and any value can be assigned to the range and angles of binary sensors.

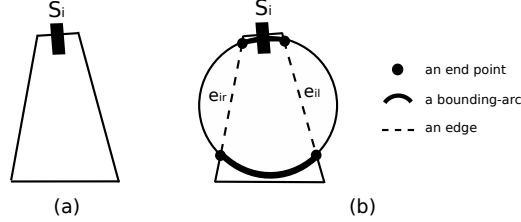


Fig. 3. (a) Approximated sensing region of a directional sensor (b) Sensing region of a directional sensor S_i within a circular monitored area

We assume that all directional sensors are to be installed on the border of a circular monitored area, as shown in Figure 3(b). It is also assumed that the sensing range of each sensor is longer than the monitored area's diameter and hence, the sensing region of the sensor within the circle is represented by two circle chords (dashed lines in Figure 3(b)).

The following notations are used in lemmas and proofs in the remainder of this paper:

- n the number of sensors.
- S_i the i^{th} sensor.
- C_i the corresponding circular region of an omni-directional sensor S_i .
- e_{lr} and e_{rl} at S_i 's position facing the circle's center, the *left* and the *right edges* of directional sensor S_i (shown as two chords within the circle as shown in Figure 3(b)).
- e_{ix} a generic term, to refer to either of the two edges of directional sensor S_i .
- The *bounding-arcs* of S_i** two arcs of the circle within the sensing region of directional sensor S_i (Figure 3(b)).
- The *end points* of S_i** the intersection points of the sensing region of directional sensor S_i with the monitored area (Figure 3(b)).

4 Unique Sub-regions

We denote the set of all created sub-regions by \mathbb{SR} . We then define C as a function that assigns every sub-region an n -bit code, *i.e.*, $C : \mathbb{SR} \rightarrow \{0, 1\}^n$, the k^{th} bit of which is set if a target in sub-region r_i is detectable by the k^{th} sensor. Figure 4(b) shows the codes assigned to each sub-region in the sample sensor arrangement shown in Figure 4(a). Sub-region r_4 , for example, has the code 001 since a target in this sub-region is detectable by S_3 but is not detectable by either of S_1 or S_2 .

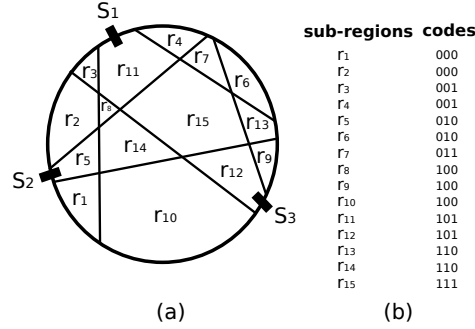


Fig. 4. Sub-region codes in a sample arrangement of three sensors

Two sub-regions r_i and r_j are equivalent if they have the same codes, *i.e.*, $r_i \sim r_j \iff C(r_i) = C(r_j)$. Given the mentioned equivalence relation among sub-regions, $\mathbb{S}\mathbb{R}$ can be divided into different equivalence classes as:

$$[r_i] = \{r_j \in \mathbb{S}\mathbb{R} \mid r_j \sim r_i\}$$

For sensor arrangement in Figure 4(a), for example, there are $2^3 = 8$ equivalence classes: $\{r_1, r_2\}$, $\{r_3, r_4\}$, $\{r_5, r_6\}$, $\{r_7\}$, $\{r_8, r_9, r_{10}\}$, $\{r_{11}, r_{12}\}$, $\{r_{13}, r_{14}\}$ and $\{r_{15}\}$.

Definition 1. *The selection of one sub-region from each equivalence class of $\mathbb{S}\mathbb{R}$, arbitrarily, establishes a set of class representatives. We call the sub-regions in the class representatives of $\mathbb{S}\mathbb{R}$, unique sub-regions; all remaining sub-regions are called duplicates.*

For example, we may choose the set of class representatives to be

$$\{r_1, r_3, r_5, r_7, r_8, r_{11}, r_{13}, r_{15}\}$$

from the arrangement shown in Figure 4(a). The sub-region r_8 is then called unique, but r_{10} is a duplicate.

The number of created sub-regions and their uniqueness depends on the shape of the sensor sensing regions as well as the arrangements of the sensors. The main aim of this paper is to determine, for each value of n , the maximum size of the set of class representatives (the maximum number of unique sub-regions), and an arrangement of sensors that would lead to this maximum, for both omni-directional and directional sensors. For every set of n sensors, there are 2^n subsets of sensors, which is therefore an upper bound on the number of unique sub-regions. However, for large values of n , the number of unique sub-regions is also limited by the maximum number of sub-regions that can be geometrically created. In Section 5, we propose an arrangement of omni-directional sensors with maximum geometrically possible number of sub-regions, which are all unique. Then, in Section 6 we prove a tighter upper bound on the

number of unique sub-regions in directional sensor arrangements and propose an arrangement with number of unique sub-regions asymptotically equivalent to the calculated upper bound.

5 Omni-directional Binary Sensors

5.1 Maximum Number of Sub-regions

Space partitioning using circles – representing the sensing region of omni-directional binary sensors – has already been investigated in the literature [7, Problem 137.1]. It has been shown that n circles can divide a plane into $n^2 - n + 2$ sub-regions, if each pair of circles intersects in two points, and no three circles intersect in the same point.

Therefore, $n^2 - n + 2$ is the maximum number of sub-regions that can be created geometrically. However, our aim is to find the maximum number of sub-regions that are all distinguishable – the maximum number of unique sub-regions. In this section, we first introduce Algorithm 1 that creates an arrangement of omni-directional sensors with maximum number of sub-regions, $n^2 - n + 2$. Theorem 1 then proves that all created sub-regions are unique and hence, $n^2 - n + 2$ is also an upper bound on the number of unique sub-regions in a omni-directional binary sensor network.

5.2 Our Proposed Arrangement of Omni-directional Sensors

The following algorithm generates an arrangement of omni-directional sensors that creates maximum number of sub-regions, $n^2 - n + 2$.

<p>Input : n omni-directional sensors</p> <p>Output: An arrangement of n omni-directional sensors with maximum possible number of sub-regions</p> <ol style="list-style-type: none"> 1 Choose X such that $X < r$, where r is the sensing range of the circular sensors. 2 $a \leftarrow X \times \cos \pi/n$ 3 Consider an n-sided convex regular polygon, with the side length equal to a. 4 Place the sensors on the vertices of the regular polygon.

Algorithm 1: Our proposed algorithm for construction of an arrangement of n omni-directional sensors

Figure 5 shows arrangements of four and five omni-directional sensors, with sensing regions C_i , generated by Algorithm 1.

In Theorem 1, we prove that all created sub-regions using Algorithm 1 are unique. We will use the following notations in the proofs in this section. S_i is

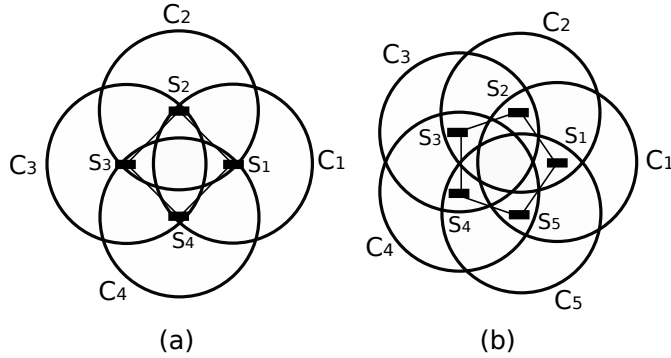


Fig. 5. Our proposed arrangement (a) 4 (b) 5 omni-directional sensors

the i^{th} sensor and C_i is its corresponding circular region. Sensors are indexed from 1 to n , counterclockwise. We denote the boundary of a circle C_i with ∂C_i . Moreover, $\partial C_j \cap \partial C_i$ denotes the rightmost intersection point of the boundaries of two circles C_i and C_j . We also define h as $n/2$, where n is the number of sensors.

Theorem 1 *All created sub-regions in our proposed arrangement generated by Algorithm 1 are unique.*

Proof. Consider a line L_0 passing through the polygon's center, vc , and $\partial C_h \cap \partial C_{h+2}$ and a Line L_1 passing through vc and $\partial C_{h+1} \cap \partial C_{h+2}$, as shown in Figure 6.

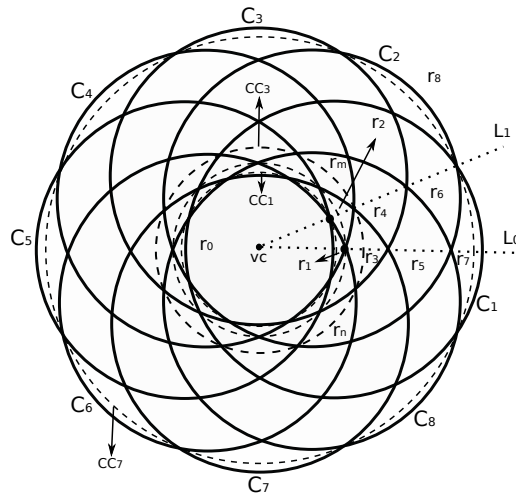


Fig. 6. A proposed arrangement of 8 omni-directional sensors

L_0 intersects h sub-regions by crossing the following circles/circle intersections in the following order:

$$\partial C_{h+1}, \partial C_h \cap \partial C_{h+2}, \partial C_{h-1} \cap \partial C_{h+3}, \dots, \partial C_2 \cap \partial C_n. \quad (1)$$

Similarly, L_1 crosses $h - 1$ sub-regions by crossing the following circle intersections in the following order:

$$\partial C_{h+1} \cap \partial C_{h+2}, \partial C_h \cap \partial C_{h+3}, \partial C_{h-1} \cap \partial C_{h+4}, \dots, \partial C_2 \cap \partial C_1. \quad (2)$$

The sub-region r_0 is within the sensing regions of all sensors. So, all bits in its code bit vector, $C[r_0]$, are set. Starting from r_0 , moving further on lines L_0 and L_1 moving from one sub-region to another, we leave the sensing region of a sensor once the line intersects with its sensing region boundary (in the order defined in 1 and 2). Whenever a circle, describing the sensing region of a sensor, is left, its corresponding bit becomes zero in the code bit vector of the newly met sub-region. Consequently, sub-regions r_1 to r_n have $n - 1$, $n - 2$, ... and 1 bits set, respectively and hence, are all unique.

Now, consider $n - 1$ circles, CC_1 to CC_{n-1} (dashed circles in Figure 6), each CC_i defined as follows: CC_i is a circle centered at vc that intersects with two vertices of sub-region r_i . Therefore, by construction, Circles CC_1 to CC_{n-1} pass through the following $n - 1$ intersection points.

$$\partial C_{h+1} \cap \partial C_{h+2}, \partial C_h \cap \partial C_{h+2}, \partial C_h \cap \partial C_{h+3}, \partial C_{h-1} \cap \partial C_{h+3}, \dots, \partial C_3 \cap \partial C_n, \partial C_2 \cap \partial C_n, \partial C_2 \cap \partial C_1.$$

By symmetry, each of circles CC_1 to CC_{n-1} crosses n sub-regions – sub-regions r_1 to r_n , respectively, and their $n - 1$ counterparts. Sub-regions r_1 to r_n and their counterparts have $n - 1$ to 1 bits set in their code bit vectors.

Table 2 shows the code bit vectors of r_3 and its counterparts - sub-region intersected by CC_3 , which all have only three zero bits in their code bit vectors. The pattern of zeros and ones in the code bit vectors of the intersected sub-regions, as shown in Table 2, will be rotated right one bit position at a time; thus, making all of them unique. Therefore, all sub-regions in the proposed arrangement of circular sensor generated by Algorithm 1 are unique.

sub-region no.	1	2	3	...	h	$h + 1$	$h + 2$	$h + 3$...	$n - 1$	n
r_3	1	1	1		0	0	0	1			1
r_m	1	1	1		1	0	0	0			1
...	1	1	1		1	1	0	0			1
...											
...	0	0	0		1	1	1	1			1
...	1	0	0		1	1	1	1			1
...											
...	1	1	1		0	1	1	1			1
r_n	1	1	1		0	0	1	1			1

Table 1. Code bit vectors of the sub-regions intersected with CC_3 (Figure 6)

6 Directional Binary Sensors

In Section 6.1, we demonstrate that n directional sensors placed on the border of a circular monitored area divide the area into at most $2n^2 + 1$ sub-regions. Then, in Section 6.2 we prove an upper bound on the number of unique sub-regions in any arrangement of directional sensors. Finally, Section 6.3 introduces our proposed arrangements of directional sensors.

6.1 Maximum Number of Sub-regions

It is clear that one sensor divides the circle into three sub-regions (Figure 7(a)). Adding any additional sensor such that its sensing region does not intersect with any existing sensing region, introduces two additional sub-regions (r_4 and r_5 in Figure 7(b)). Therefore, if no pair of n sensor sensing regions intersect within the circle, then the number of sub-regions is $3 + 2(n - 1) = 2n + 1$.

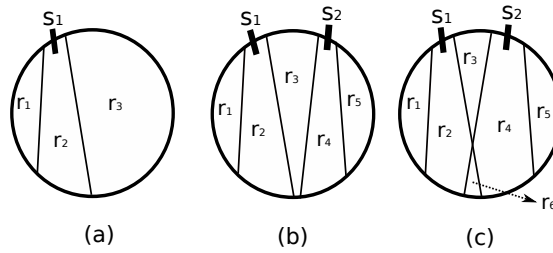


Fig. 7. Intersections between sensing regions

Now, assuming that no more than two edges have the same intersection point, each intersection point between two sensing regions generates one new sub-region (r_6 in Figure 7(c)). Therefore, we have:

$$n_r = 2n + 1 + n_m, \quad (3)$$

where n_r and n_m denote the total number of sub-regions and intersections within the circle, respectively.

Since a sensor region can intersect another sensor region at at most four points, the maximum number of intersections is $n_m^{max} = \frac{4(n-1).n}{2} = 2n^2 - 2n$, where the division by two accounts for each intersection being counted twice. Thus, by Equation 3, the maximum number of sub-regions will be:

$$n_r^{max} = 2n + 1 + n_m^{max} = 2n + 1 + 2n^2 - 2n = 2n^2 + 1. \quad (4)$$

Therefore, n directional sensors placed on the border of a circular monitored area divide the area into at most $2n^2 + 1$ sub-regions.

6.2 An Upper Bound on the Number of Unique Sub-regions

Notations defined in Section 3 are all used in the lemmas and proofs in this section. To maximize the number of created sub-regions, we make the following assumptions:

Assumption 1 Sensor edges do not overlap.

Assumption 2 No two edges intersect on the monitored area.

Assumption 3 No more than two edges have the same intersection point within the monitored area.

Figure 8 (a-c) illustrate a violation of the Assumptions 1-3, respectively.

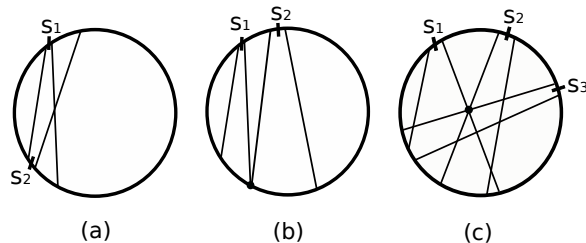


Fig. 8. Violating assumption 1 (a), 2 (b), and 3 (c)

Definition 2 and Lemmas 1 to 3 are used in the upper bound proof in Theorem 2. We first divide the sub-regions into *boundary* and *inner* sub-regions (Definition 2). Lemmas 1 and 2 prove the number of boundary sub-regions and the number of bits set in their codes, respectively. In Lemma 3, we calculate the maximum number of created sub-regions when the codes of the boundary sub-regions are known. The three lemmas are then used to prove an upper bound on the number of unique sub-regions in Theorem 2.

Definition 2. *Boundary sub-regions and inner sub-regions.* We create the multiset of boundary sub-regions, $\mathbb{S}\mathbb{R}_b$, by traversing the circle's perimeter clockwise or counterclockwise until the same point is reached and including the intersected sub-regions in $\mathbb{S}\mathbb{R}_b$. All sub-regions that are not in $\mathbb{S}\mathbb{R}_b$ are called *inner sub-regions*.

For arrangement in Figure 9(a), for example,

$$\mathbb{S}\mathbb{R}_b = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}\}.$$

$\mathbb{S}\mathbb{R}_b$ is a multiset because a sub-region might be encountered twice in this traversal. For example, in Figure 9(b), sub-region r_{10} is intersected twice and hence, included twice in $\mathbb{S}\mathbb{R}_b = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{10}\}$.

Lemma 1 *The cardinality of the multiset of boundary sub-regions $\mathbb{S}\mathbb{R}_b$ is $4n$.*

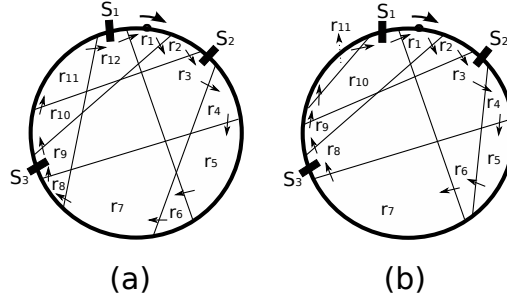


Fig. 9. Boundary sub-regions

Proof. The sensing region of each sensor intersects the circle's perimeter at four end points. Therefore, by Assumptions 1 and 2, there is a total of $4n$ intersection points on the circle's perimeter, leading to the size of multiset \mathbb{SR}_b also being $4n$.

Lemma 2 *The number of sub-regions in \mathbb{SR}_b whose codes have an odd number of bits set equals the number of sub-regions in \mathbb{SR}_b with an even number of bits set.*

Proof. By Assumption 1, each pair of neighboring boundary sub-regions have exactly one edge in common. Therefore, each pair of neighboring boundary sub-regions has only a one-bit difference in their codes. Thus, if a boundary sub-region has an odd (even) number of bits set in its code, its clockwise neighbor has an even (odd) number of bits set in its code, which proves the lemma.

Lemma 3 *In any arrangement of n sensors, $n \geq 2$, there are at most $(2n^2 + n + 1) - (\sum_{i=1}^n i \cdot SR_b^i)/2$ sub-regions within the circle, where SR_b^i is the number of elements of multiset \mathbb{SR}_b with i bits set in their codes.*

Proof. When $n \geq 2$, for each end point located in the bounding-arcs of S_i , there is an intersection between an edge e_{jx} and an edge of S_i that is missed ((e_{jr}, e_{ir}) and (e_{jl}, e_{il}) in Figure 10(a) and (e_{jr}, e_{il}) and (e_{jr}, e_{ir}) in Figure 10(b)). Moreover, if no end point of e_{jx} ends up in the bounding-arcs of S_i , e_{jx} might still intersect with none of the edges of sensor e_{ix} (e_{jl} Figure 10(c)).

Therefore, if we denote the number of sensor edges that do not intersect with edge e_{ix} by $m_{e_{ix}}^-$ and the number of end points located in the bounding-arcs of S_i with n_e , then we have:

$$m_{e_{il}}^- + m_{e_{ir}}^- \geq n_e. \quad (5)$$

On the other hand, the number of boundary sub-regions covered in the bounding-arcs of S_i equals $n_e + 2$, because each sensor has two bounding-arcs and each end point creates a new boundary sub-region within those bounding-arcs. These are also the only boundary sub-regions that have a bit set in the i^{th}

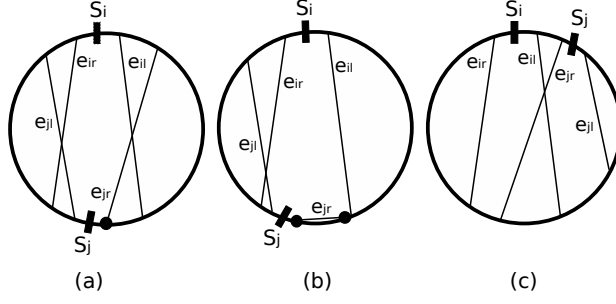


Fig. 10. Different positions of edges of sensor S_j with respect to sensor S_i

position of their codes. Therefore, denoting the number of boundary sub-regions with a bit set at the i^{th} position of their codes by $SR_b^{i=1'}$, using Equation 5:

$$m_{e_{il}}^- + m_{e_{ir}}^- + 2 \geq SR_b^{i=1'}. \quad (6)$$

Moreover, the total number of bits set in all boundary sub-regions codes is $SR_b^1 + 2SR_b^2 + 3SR_b^3 + \dots = \sum_{i=1}^n i \cdot SR_b^i$, where SR_b^i is the number of elements of multiset SR_b with i bits set in their codes. Therefore, we have:

$$\sum_{i=1}^n (m_{e_{il}}^- + m_{e_{ir}}^- + 2) \geq \sum_{i=1}^n i \cdot SR_b^i. \quad (7)$$

On the other hand, the number of intersections within the circle is:

$$(2n^2 - 2n) - \sum_{i=1}^n ((m_{e_{il}}^- + m_{e_{ir}}^-)/2),$$

where the first part, $2n^2 - 2n$, is the number of intersections when all edges intersect with one another. The second part, $\sum_{i=1}^n ((m_{e_{il}}^- + m_{e_{ir}}^-)/2)$, is the number of missed intersections; the sum is divided by two, as each possible intersection is counted twice.

Therefore, by Equation 7, the maximum number of intersections within the circle is:

$$2n^2 - 2n - \left(\frac{\sum_{i=1}^n i \cdot SR_b^i}{2} - n \right). \quad (8)$$

Using Equations 8 and 3 (by Assumption 3), the maximum number of partitions is calculated as:

$$2n + 1 + 2n^2 - 2n - \left(\frac{\sum_{i=1}^n i \cdot SR_b^i}{2} - n \right) = 2n^2 + n + 1 - \frac{\sum_{i=1}^n i \cdot SR_b^i}{2}. \quad (9)$$

Theorem 2 *An upper bound on the number of unique sub-regions for any arrangement of n directional sensors is $2n^2 - 3n + 2$, $n \geq 2$.*

Proof. As in Definition 2, the created sub-regions in any arrangement of sensors are divided into two groups of boundary and inner sub-regions. An upper bound on the number of unique sub-regions can be the maximum value of $SR_t - SR_{bd} - SR_{id}$, where SR_t is the total number of sub-regions and SR_{bd} and SR_{id} are the number of duplicate sub-regions among boundary and inner sub-regions, respectively. Therefore, $SR_{upper} = SR_t^{max} - SR_{bd}^{min} - SR_{id}^{min}$.

Assume that the number of elements of multiset SR_b with i bits set in their codes is denoted by SR_b^i . Since there are at most n unique boundary sub-regions with one bit set in their codes and only one unique boundary sub-region with code zero, the minimum number of duplicate boundary sub-regions SR_{bd}^{min} is $(SR_b^1 - n) + (SR_b^0 - 1)$. To calculate an upper bound, we can also assume that all inner sub-regions are unique, or $SR_{id}^{min} = 0$. Therefore, an upper bound on the number of unique sub-regions for any arrangement of n directional sensors is:

$$SR_{upper} = SR_t^{max} - ((SR_b^1 - n) + (SR_b^0 - 1)). \quad (10)$$

Then, by Lemma 3:

$$SR_{upper} = 2n^2 + n + 1 - \left(\frac{\sum_{i=1}^n i \cdot SR_b^i}{2}\right) - ((SR_b^1 - n) + (SR_b^0 - 1)). \quad (11)$$

By Lemma 1, $\sum_{i=0}^n SR_b^i = 4n$. Therefore, we have:

$$SR_{upper} = 2n^2 + n + 1 - \left(\frac{\sum_{i=1}^n i \cdot SR_b^i}{2}\right) - \left(4n - \left(\sum_{i=2}^n SR_b^i\right) - n - 1\right).$$

or,

$$SR_{upper} = 2n^2 - 2n + 2 - \left(\frac{\sum_{i=1}^n SR_b^i}{2}\right) - \left(\frac{\sum_{i=1}^n (i-1) \cdot SR_b^i}{2}\right) + \left(\sum_{i=2}^n SR_b^i\right).$$

Using Lemma 1, we have:

$$SR_{upper} = 2n^2 - 2n + 2 - \frac{(4n - SR_b^0)}{2} - \left(\frac{\sum_{i=1}^n (i-1) \cdot SR_b^i}{2}\right) + \left(\sum_{i=2}^n SR_b^i\right).$$

The upper bound is therefore:

$$SR_{upper} = 2n^2 - 3n + 2 + \left(\frac{SR_b^0 + SR_b^2}{2} - n\right) - \left(\frac{\sum_{i=4}^n (i-3) \cdot SR_b^i}{2}\right). \quad (12)$$

From Lemmas 1 and 2, we know that $SR_b^0 + SR_b^2 \leq 2n$, i.e., the value of $((SR_b^0 + SR_b^2)/2 - n)$ is not greater than zero. Thus, we conclude that an upper bound on the number of unique sub-regions for any arrangement of n sensors is $2n^2 - 3n + 2$.

To show that the calculated upper bound is tight, we propose, in the next section, an arrangement of sensors whose number of unique sub-regions is asymptotically equivalent to the calculated upper bound.

6.3 Our Proposed Arrangements of Directional Sensors

We have proved an upper bound on the number of unique sub-regions for any arrangement of n sensors, $2n^2 - 3n + 2$, $n \geq 2$. In this section, we provide an algorithm to construct a regular arrangement of n sensors that has $2n^2 - 5n + 1$ unique sub-regions for even n and $2n^2 - 5n + 5$ unique sub-regions for odd n . Using a probabilistic method, we will also show that we can in fact generate arrangements that have a greater number of unique sub-regions than the regular arrangement and are very close to the calculated upper bound in Section 6.2.

Arrangement I We describe in Algorithm 2 the construction of our regular arrangement of n directional sensors, Arrangement I. For an even number of sensors, the sensors are placed equidistant around the circle with an angle of $2\pi/n$ apart from each other. If n is odd, the sensors are placed $2\pi/(n+1)$ apart, which leaves one position empty. For a given sensor S_i , we define its left (right) angle β_i (α_i) as the angle between e_{il} (e_{ir}) and the ray pointing from S_i to the centre of the circle (see Figure 11(a) for example for sensor S_1). In any Arrangement I, all α_i s and β_i s are equal to a given α and β , respectively. Figure 11 shows such arrangements for five and six sensors, computed by Algorithm 2. The sensors are positioned on a circular monitored area and are numbered anticlockwise from 1 to n (see Figure 11).

Input : n sensors, a circular monitored area, α, β
Output: An Arrangement I of n sensors

- 1 $\alpha \leftarrow \pi/n + \epsilon$
- 2 $\beta \leftarrow \pi/n - (\epsilon + \epsilon')$
- 3 $\forall i \in \{1, \dots, n\} : \alpha_i \leftarrow \alpha$
- 4 $\forall i \in \{1, \dots, n\} : \beta_i \leftarrow \beta$
- 5 **if** n is even **then**
- 6 Place the sensors equidistant $2\pi/n$ apart.
- 7 **end**
- 8 **if** n is odd **then**
- 9 Place the sensors equidistant $2\pi/(n+1)$ apart, leaving one position empty.
- 10 **end**

Algorithm 2: Our proposed algorithm for construction of an Arrangement I of n directional sensors

We will initially assume that n is even and we define h as $n/2$. Moreover, arithmetic is modulo n on the domain $\{1, \dots, n\}$. Given the values of $\alpha = \pi/n + \epsilon$ and $\beta = \pi/n - (\epsilon + \epsilon')$ for small ϵ and ϵ' as in Algorithm 2, Sensor S_i 's left edge aims slightly to the left of sensor S_{i+h+1} , while its right edge aims slightly to the left of sensor S_{i+h-1} . Lemma 4 is used to compute the number of unique partitions in any Arrangement I.

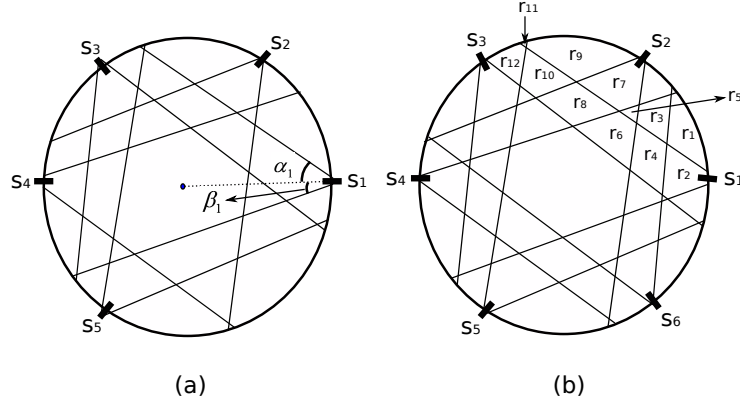


Fig. 11. Our proposed arrangements of (a) five and (b) six sensors

Lemma 4 *Edge e_{1r} intersects the following edges in the following order:*

$$e_{(h+3)r}, e_{2l}, e_{(h+4)r}, e_{3l}, \dots, e_{nr}, e_{(h-1)l}, e_{(h+1)l}, e_{2r}, e_{(h+2)l}, e_{3r}, e_{(h+3)l}, \dots, e_{(h-1)r}, e_{(n-1)l}.$$

Proof. Consider two edges: e_{1r} (from sensor S_1 to sensor S_h) and e_{il} (from sensor S_i to sensor S_{i+h+1}), for $i \in \{2, 3, \dots, n-1\}$. By symmetry, if there were no ϵ and ϵ' , these two edges would meet at a point that lies on the diameter from sensor $S_{(i+1)/2}$ to sensor $S_{(i+1)/2+h}$. Therefore, as i increases, the intersection point of e_{il} with e_{1r} moves further and further counterclockwise, and therefore in the order given by i . Note that none of e_{hl} , e_{hr} , e_{nl} intersects e_{1r} . By construction, edge e_{il} is almost coincident with $e_{(h+i+1)r}$, but due to the effect of the ϵ and ϵ' deflections, $e_{(h+i+1)r}$ intersects e_{1r} slightly closer to sensor $h+i+1$ than e_{il} does (for $i \neq h$).

Using Lemma 4, we enumerate the neighboring sub-regions of edge e_{1r} , which are the sub-regions whose boundaries are formed by parts of edge e_{1r} . In Figure 11, for example, the neighboring sub-regions of edge e_{1r} are sub-regions r_1 to r_{12} . Sub-region r_2 is the start sub-region of sensor S_1 .

Table 2 shows the enumeration of neighboring sub-regions of edge e_{1r} . This enumeration is analogous for edges e_{ir} : there will be corresponding sub-regions for each edge of e_{ir} . Each row i in Table 2 shows the code bit vector of the i^{th} neighboring sub-region of edge e_{1r} . When counting neighboring sub-regions of e_{2r} , e_{3r} etc., the pattern of zeros and ones in the bit vectors in Table 2 will be rotated right one bit position at a time. Thus, we can establish symmetries and identify when sub-regions are not unique. In particular, row $2n-5$ is equivalent to row $2n-10$, row $2n-4$ is equivalent to row $2n-8$, row $2n-3$ to row $2n-11$ and row $2n-2$ to row $2n-9$. In fact, this process will continue, repeating with period four, due to the symmetries, except that lines $2n-7$, $2n-6$, $4n-13$ have no counterparts and line $4n-12$ is a duplicate of the first line. The number of unique sub-regions among neighboring sub-regions of edge e_{1r} is then $2n-5$. The

sub-region in the fields of all sensors is not included in Table 2. Hence, the total number of unique sub-regions is $(2n - 5)n + 1 = 2n^2 - 5n + 1$.

Our discussion assumed that n is even. If n is odd, we calculate the number of unique sub-regions in the $n + 1$ -size arrangement: $2(n + 1)^2 - 5(n + 1) + 1$, because $n + 1$ is even. We subtract the neighboring sub-regions of the $(n + 1)^{th}$ sensor's edges: there are $2(n + 1) - 5$ intersections on its left edge, $2(n + 1) - 7$ intersections on its right edge, each corresponding to one neighboring sub-region of S_{n+1} 's edges. We must also exclude the start sub-region of sensor S_{n+1} . In summary: the number of unique sub-regions is $2(n + 1)^2 - 5(n + 1) + 1 - ([2(n + 1) - 5] + [2(n + 1) - 7] + 1) = 2n^2 - 5n + 5$.

sub-region no.	1	2	3	4	5	...	$h - 2$	$h - 1$	h	$h + 1$	$h + 2$	$h + 3$	$h + 4$...	$n - 1$	n
1	0	0	0	0	0		0	0	0	1	1	0	0		0	0
2	1	0	0	0	0		0	0	0	1	1	0	0		0	0
3	0	0	0	0	0		0	0	0	1	1	1	0		0	0
4	1	0	0	0	0		0	0	0	1	1	1	0		0	0
5	0	1	0	0	0		0	0	0	1	1	1	0		0	0
6	1	1	0	0	0		0	0	0	1	1	1	0		0	0
$2n - 11$	0	1	1	1	1		1	0	0	1	1	1	1		1	0
$2n - 10$	1	1	1	1	1		1	0	0	1	1	1	1		1	0
$2n - 9$	0	1	1	1	1		1	0	0	1	1	1	1		1	1
$2n - 8$	1	1	1	1	1		1	0	0	1	1	1	1		1	1
$2n - 7$	0	1	1	1	1		1	1	0	1	1	1	1		1	1
$2n - 6$	1	1	1	1	1		1	1	0	1	1	1	1		1	1
$2n - 5$	0	1	1	1	1		1	1	0	0	1	1	1		1	1
$2n - 4$	1	1	1	1	1		1	1	0	0	1	1	1		1	1
$2n - 3$	0	0	1	1	1		1	1	0	0	1	1	1		1	1
$2n - 2$	1	0	1	1	1		1	1	0	0	1	1	1		1	1
$2n - 1$	0	0	1	1	1		1	1	0	0	0	1	1		1	1
$2n$	1	0	1	1	1		1	1	0	0	0	1	1		1	1
$4n - 17$	0	0	0	0	0		0	1	0	0	0	0	0		1	1
$4n - 16$	1	0	0	0	0		0	1	0	0	0	0	0		1	1
$4n - 15$	0	0	0	0	0		0	0	0	0	0	0	0		1	1
$4n - 14$	1	0	0	0	0		0	0	0	0	0	0	0		1	1
$4n - 13$	0	0	0	0	0		0	0	0	0	0	0	0		0	1
$4n - 12$	1	0	0	0	0		0	0	0	0	0	0	0		0	1

Table 2. Enumeration of code bit vectors of neighboring sub-regions of edge e_{1r} .

Arrangement II This type of sensor arrangement is constructed using Algorithm 2, but with new constraints for values of α and β , i.e., $\alpha + \beta < 2\pi/n$ and $\alpha, \beta < \pi/n$. We assume again that the sensors are anticlockwise ordered from 1 to n . Therefore, each field of sensor S_i covers only sensor $S_{i+n/2}$. Figure 12 shows a sample of Arrangement II for six sensors.

For each n ranging from 6 to 20, the number of unique sub-regions is then computed by doing simulations for every value of angles α and β with angles change step of 0.1° .

The achieved maximum number of unique sub-regions for such arrangements (Maximum for Arrangement II) are shown in Table 3 and in Figure 13. We compare these numbers with the maximum number of unique sub-regions from

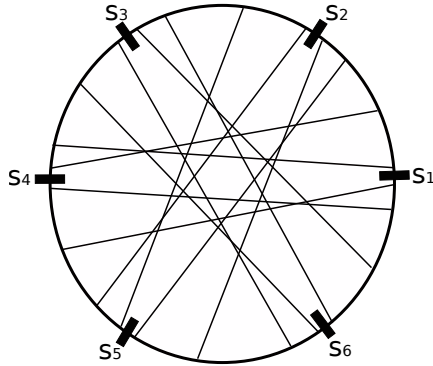


Fig. 12. Arrangement II for 6 sensors

Arrangement I (Section 6.3) and the upper bound calculated in Section 6.2. The number of unique sub-regions in both Arrangements I and II are asymptotically equivalent to the upper bound.

Table 3. The achieved maximum number of unique sub-regions for n ranging from 6 to 20

n	Maximum for Arrangement I	Maximum for Arrangement II	Upper Bound
6	43	53	56
7	67	75	79
8	89	102	106
9	121	132	137
10	151	167	174
11	188	205	213
12	229	248	256
13	274	294	301
14	323	344	353
15	376	399	407
16	433	457	466
17	494	518	529
18	559	584	596
19	628	654	667
20	701	727	742

7 Conclusions and Future Work

Binary sensors can be used to partition an area into unique sub-regions and hence, provide localization functionality. We calculated an upper bound on the

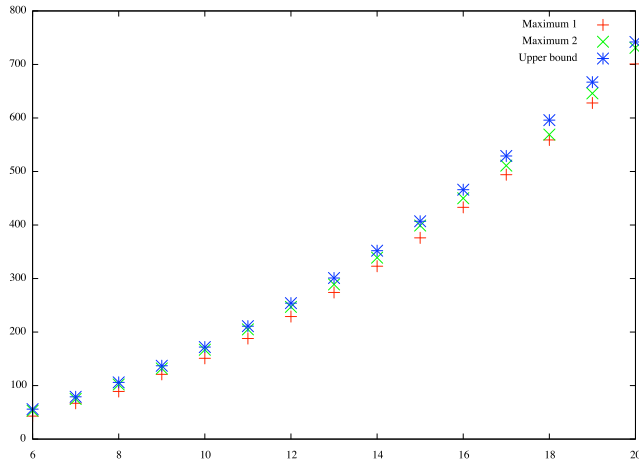


Fig. 13. Upper bound and maximum on the number of unique sub-regions in our proposed arrangements

number of unique sub-regions that a set of omni-directional or directional sensors can achieve. We also proposed regular arrangements for both omni-directional and directional sensors whose number of unique sub-regions is asymptotically equivalent to our calculated upper bound. This outcome gives researchers an insight into how many sub-regions can be created using a specific number of sensors as well as the number of sensors required to achieve a certain accuracy. Finding a constructive algorithm to generate an arrangement of n sensors whose number of unique sub-regions equals the calculated upper bound is still an open problem for directional sensors. Moreover, since the size of the created sub-regions has impact on the localization accuracy, we are currently looking at partitioning schemes that lead to sub-regions that are as equally-sized as possible so that we achieve a uniform resolution.

Acknowledgement. This work was funded by National ICT Australia (NICTA). NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

1. Aslam, J., Butler, Z., Constantin, F., Crespi, V., Cybenko, G., Rus, D.: Tracking a Moving Object with a Binary Sensor Network. In: Proceedings of the first International Conference on Embedded Networked Sensor Systems. pp. 150–161 (2003)
2. Bouet, M., Pujolle, G.: 3-D Localization Schemes of RFID Tags with Static and Mobile Readers. In: Networking. pp. 112–123 (2008)

3. Djurić, P.M., Vemula, M., Bugallo, M.F.: Target Tracking by Particle Filtering in Binary Sensor Networks. *IEEE Transactions on Signal Processing* 56(6), 2229–2238 (2008)
4. Ehrenberg, I., Floerkemeier, C., Sarma, S.: Inventory Management with an RFID-equipped Mobile Robot. In: *International Conference on Automation Science and Engineering*. pp. 1020–1026 (2007)
5. Hinske, S.: Determining the Position and Orientation of Multi-Tagged Objects Using RFID Technology. In: *IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW)*. pp. 377–381 (2007)
6. Kershner, R.: The Number of Circles Covering a Set. *American Journal of Mathematics* 61, 665–671 (1939)
7. Konhauser, J.D., Velleman, D., Wagon, S.: *Which Way Did the Bicycle Go?: And Other Intriguing Mathematical Mysteries*. Mathematical Association of America (1996)
8. Mehmood, M.A., Kulik, L., Tanin, E.: Autonomous Navigation of Mobile Agents Using RFID-enabled Space Partitions. In: *ACM SIGSPATIAL GIS*. pp. 173–182 (2008)
9. Murakita, T., Ikeda, T., Ishiguro, H.: Human Tracking using Floor Sensors based on the Markov Chain Monte Carlo Method. In: *International Conference on Pattern Recognition*. pp. 917–920 (2004)
10. Reza, A.W., Geok, T.K.: Investigation of Indoor Location Sensing via RFID Reader Network Utilizing Grid Covering Algorithm. *Wireless Personal Communications* 49, 67–80 (2009)
11. Shrivastava, N., Mudumbai, R., Madhow, U., Suri, S.: Target Tracking with Binary Proximity Sensors. *ACM Transactions on Sensor Networks* 5(4), 30:1–30:33 (2009)
12. Wang, Z., Bulut, E., Szymanski, B.: Distributed Energy-Efficient Target Tracking with Binary Sensor Networks. *ACM Transactions on Sensor Networks* 6(4), 32:1–32:32 (2010)
13. Want, R.: An Introduction to RFID Technology. *Pervasive Computing* 5, 25–33 (2006)
14. Want, R., Hopper, A., Falc, V., Gibbons, J.: The Active Badge Location System. *Information Systems* 10(1), 91–102 (1992)