An efficient method to verify and assess $(k - \omega)$ coverage in wireless multimedia sensor networks

Nguyen Thi My Binh[1], Chu Minh Thang[1], Huynh Thi Thanh Binh[1]

Abstract

Barrier coverage problems in camera sensor networks have drawn the attention by academic community because of their huge potential applications. Various versions of barrier coverage under wireless camera sensor networks have been studied such as minimal exposure path, strong/weak barrier, 1/k barrier, full view barrier problems. In this paper, based on new $(k - \omega)$ angle coverage model, we study how to achieve $(k - \omega)$ angle barrier coverage problem under uniform random deployment scheme (hereinafter $A(k - \omega)ABC$ problem). This problem aims to juggle whether any given camera sensor networks is $(k - \omega)$ angle barrier coverage. A camera sensor network is called $(k - \omega)$ angle barrier coverage if any crossing path is $(k - \omega)$ angle coverage. The $A(k - \omega)ABC$ problem is useful because it can make balance of the number of camera sensors used and the information retrieved by the camera sensors. Furthermore, this problem is vital for design and applications for camera sensor networks when camera sensor nodes were deployed randomly. Thus, we formulate the $A(k-\omega)ABC$ problem and then proposed an efficient method for solving this problem. An extensive experiments were conducted on random instances, and the results indicated that the proposed algorithm can produce high quality and stable solutions, as well as achieve better results than full-view coverage model.

Keywords: minimal exposure path, wireless sensor networks, stationary sensor networks, mobile sensor networks

1. Introduction

Barrier coverage in WSNs is a critical issue for various sensor network security applications, e.g., boundary surveillance (national border, critical resource protection) and intrusion detection. Barrier coverage is formed by a sensing barrier for detecting intruders crossing a region of interest. Compared with full coverage, barrier coverage requires much less quantity of sensors and energy. Thus, barrier coverage is considered more scalable and attractive in large-scale deployment in reality.

Recently, camera sensor networks have drawn the attention of research community [1-6]. Compared with conventional scalar sensors, camera sensors can harvest much richer information of the environment in the forms of images or videos and thus promise an extremely potential in applications. However, cost of camera sensors is higher than scalar sensor. Camera sensors are randomly scattered for achieving barrier coverage in a large scale is a very challenged. Thus, security surveillance or intrusion detection application is expected to build up a cost efficient camera sensor barrier such that every intruder image can be detected effectively. However, the camera sensors barrier coverage problem is much different and more complicated than the conventional barrier coverage problem. When sensing range of a chain of camera sensors across

Email addresses: binhntm@haui.edu.vn (Nguyen Thi My Binh), mthang129@gmail.com (Chu Minh Thang), binhhtt@soict.hust.edu.vn (Huynh Thi Thanh Binh)

the surveillance region is simply combining, that does not provide effective barrier coverage. Because an intruder may cross the barrier without being recognized, i.e., its face image could not be caught. The fundamental difference between camera and traditional scalar sensors in coverage is that camera sensors may capture very different scenes of the same object if they are from different viewpoints. For instance, when a camera is placed behind the intruder, no face image can be recognized. Researches in the computer vision field show that the object is more likely to be recognized by the recognition system if the image is captured at or near the frontal viewpoint [1], i.e., if the object is facing straight to the camera. As the angle between the object facing direction and the camera viewing direction (defined by the vector from the object to the camera, Figure 1(a)) increases, the detection rate decreases dramatically. Therefore, to keep a high level monitor quality, a good camera sensor barrier should ensure that no matter where the traversing object faces, there is always some camera to effectively capture its face image.

To get more information of the intruders, especially their face identifications, full-view coverage model was first introduced in Wang and Cao [8, 9]. An object is full-view covered if there is always at least one camera covering it no matter which object facing direction and the cameras viewing direction is sufficiently close to the object facing direction. Obviously, we can obtain more information about the intruder in full-view model. In [2], the authors investigate the necessary and sufficient conditions to achieve full view coverage under uniform deployment and Poisson deployment.

Ma et al. [3] studied the minimum camera barrier coverage problem in WCSNs when the camera sensors are deployed randomly in a monitor field. The authors have some extension, the algorithm reduces the number of camera sensors, but it still requires a lot of camera sensors to construct a full viewed barrier coverage. In [4], Xu et. al. also want to maintain the quality of aggregate information with the least camera sensors. Basing on $(k-\omega)$ angle coverage model [5], they produce a new barrier coverage model named by (k,ω)-angle barrier coverage. The object is said $(k-\omega)$ angle coverage when it is simultaneously monitored by at least k sensors, and to ensure that k different sensors monitor different parts of the object, the angle between two arbitrary sensor directions must be equal or greater. The monitored field exists a $(k-\omega)$ angle barrier coverage if it has a connected region across the monitor field such that every point within the region is $(k-\omega)$ angle coverage. The $(k-\omega)$ angle barrier coverage is useful because of a balance of the number of camera sensors used and the information retrieved by the camera sensors.

However, a fundamental problem of assessing the quality of a $(k-\omega)$ angle barrier has not been addressed efficiently. Currently, in the tradition model of attenuation, when evaluating the coverage quality of a sensor toward a certain point, the coverage value formula is defined to be affected only by the distance from the sensor to the considered point, which may lead to several exceptional inconsistency regarding evaluate the quality of coverage, especially in the problem of $(k-\omega)$ barrier in the sensor network. Firstly, the model can only illustrate the general coverage of a point to a certain sensor. As a result, it is impossible to achieve a metric measuring the amount of information on every direction of an intruder captured by the network of sensors, which is fundamental in several coverage problems such as the $k-\omega$ cover or the full view cover problem. Secondly, this model shows little consistency considering the coverage of a point in a sensor network. In case of evaluating the coverage of a point exposed to numerous sensors, there have been 2 common models, the All-Sensor Field Intensity and the Closest-Sensor Field Intensity model. While the latter failed to illustrate the cooperation of the sensors in sensing the same object, which greatly underrate the real coverage of the network on the intruder, the former model does not take into account the repetition of information captured by numerous sensors, thus exaggerate the value of coverage dramatically.

As a result, we found it necessary to devise a more preferable attenuated model assessing the coverage quality of the sensor network toward a point in the field of interest, the model can later be generalised to evaluate the coverage of the sensor network on a line or a closed region. To formulate the coverage value regarding information from every direction of the intruder, it is necessary to consider its shape. In this article, the problem is tackled generally, which results in we using the rule of symmetric to make an assumption that the intruder is a circle. This geometrical shape is sufficient since the coverage is considered in the 2-D plane only, and the circle itself has parts of the circumference followed every possible direction, which help illustrate the devised model more thoroughly and clearly.

The coverage model implements the idea of Divide-And-Conquer. The intruder is differentiated into several small parts. Each of them is then evaluated separately, then added together to obtain the total coverage of the sensor network on the intruder at the considered position. Each small part of the circle, in turn, is calculated with each sensor that cover it, then the largest coverage value is taken. This operation will prevent the coverage overrated from several sensors having similar position toward the considered point, which in real life will serve no purpose of obtaining more information of the specific part on the intruder.

In this paper, we focus on evaluating a sensing field based on two factors: the probability of successfully forming a $(k - \omega)$ angle barrier, and the average quality of the found barriers in the sensing fields. This problem is vital for optimizing the efficiency of sensor placement in sensor networks when camera sensor nodes were deployed randomly. For instance, camera sensors have to be scattered by plane or artillery if the region of interest is hostile or inaccessible or deficiency on time, manpower or funds prevents careful arrangement of every single sensor.

The main contributions of this article are as follows.

- Formulate achieving $(k - \omega)$ angle barrier coverage problem.
- Propose an efficient method to deterministically verify if a monitored field can be achieved a $(k - \omega)$ angle barrier coverage in by any given set of camera sensors.
- Evaluate the found barriers with the devised metric.
- Conduce experiments in various scenarios to examine the result and the computation time of the proposed algorithm.
- Analyze and estimate the parameters effect on performance of proposed algorithm.

The rest of the paper is organized as follow. Related works are presented in section 2. Section 3 formulates A $(k - \omega)$ angle barrier problem. Section 4 introduces proposed algorithm. Section 5 gives our experiments along with computational and comparative results as well as conclusion in section 6.

2. Related works

The concept of **barrier coverage** was first introduced in [6] in the field of robotics. Up to now, barrier coverage has attracted intensive attention from research community. Earlier, almost researches focus on omni-directional sensors [**? ? ? ?**]. In [**?**], Kumar et al. defined the concept of k-barrier coverage. Two types of barrier coverage were also introduced: weak barrier coverage and strong barrier coverage. Chen et al. [**?**] introduced a new model of barrier coverage called local barrier coverage. It is proved there that sensors can locally determine the existence of local barrier coverage, while it is impossible for global barrier coverage. Under this model, the authors developed a sleep-wakeup algorithm for maximizing the network lifetime and showed that their algorithm can provide close to optimal enhancement. In [**?**], Liu et al. derived critical conditions for strong barrier coverage in two-dimensional rectangular where sensors are deployed uniformly and randomly. They pointed out that if width W and length L of the rectangular satisfies $W = \Omega(L)$, there exist, with high probability, multiple disjoint barriers when sensors density reaches a certain value. Whereas, when $W = O(L)$, with high probability, there will be a crossing path not covered by any sensor regardless of the sensor density. Based on this result, the authors devised an efficient distributed algorithm to construct disjoint barriers in a large sensor network to cover long boundary areas of irregular shapes. In [**?**], based on probabilistic sensing model, Yang et al. studied barrier coverage problem under the assumption that neighboring sensors may collaborate with each other to form a virtual sensor which makes the detection decision depending on combined sensed readings. This is referred to as barrier information coverage.

Later, with the wide use of directional sensor, especially camera sensor, there were more and more interest in Wireless Camera Sensor Networks (WCSNs). Barrier coverage problems in WCSNs are much more difficult and challenging compared to those in traditional scalar WSNs. The biggest difference between WCSNs and scalar WSNs and also the biggest challenge when working with WCSNs is the directional characteristic of camera sensor. Since directionality was taken into consideration, many coverage models have been proposed for improving monitoring performance of sensor networks.

In this paper, we study $(k-\omega)$ coverage model. In the followings, we present some related works studying on some different coverage models which are closely relevant to $(k-\omega)$ coverage model as well as previous works on $(k-\omega)$ coverage model.

In order to get more information of the object, especially face regconition, full-view coverage was introduced in [**?**] by Wand and Cao. An object is full-view covered if there is always a camera to cover it no matter which direction it faces and the angle between the camera's viewing direction and the object's facing direction is less than a predefined parameter θ. The authors proposed a method for full-view coverage verification on a target field. After that, they derived an estimation of the sensor density needed for full-view coverage in a random deployment. Based on this work, the authors further studied the problem of constructing a camera barrier in [7]. They proposed a method to select camera sensors from an arbitrary deployment to form a camera barrier and then presented a technique for reducing the number of cameras used since there might be redundant cameras (cameras that can be turned off without breaking the barrier) after barrier is formed. The monitoring region is divided into sub regions where each sub region is a set of points covered by the same set of sensors. Using notion of safe region, the authors can verify whether or not a sub region is full-view covered. After that, they can transform the problem into graph and finding a barrier is converted to finding a path from source node to sink node in the graph. For redundancy reduction, another type of graph is introduced and the problem is equivalent to maximum independent vertex set problem.

Inspired by this work, in [3], Ma et al. proposed a better method for constructing camera barrier. The authors proved that for a sub region which is not full-view covered, it always contains two parts: the full-view covered one and the non full-view covered one. Then, each node in the graph can be a full-view covered sub region or the full-view covered part of a sub region that is not full-view covered. The problem of finding the minimum camera barrier is transformed to the problem of finding the shortest path from source node to sink node in the graph. **However, we see that the algorithm used in this work cannot be used to solve the problem**. This algorithm can be considered as a variant of Dijkstra algorithm. The only two differences are: (1) the edge has no weight but each vertex v has a "weight" denoted by $I(v)$ and (2) the operator used for updating label of a vertex is union instead of addition. The second difference is the reason why the algorithm has prolem. According to the algorithm, when a vertex v is labeled, every previous vertex in the path from source vertex s to v is also labeled. Supposed u is such a vertex between s and v, the shortest path from s to u must be contained in the path from s to v. But this is not true when the operator union is used (figure 1)

Following the above studies, there have been much effort on various problems under full-view coverage model. In [**?**], the authors study the effects of realistic sea surface movements in achieving full-view coverage in CSN. They also addressed the problem of reducing the total transmission power among sensors by proposing a cooperative transmission method. Under the deterministic deployment strategy, the authors in [**?**] proposed an efficient deployment pattern algorithm DPA to compute the critical density and positions of sensors for achieving full-view coverage on the target field. Moreover, they devised a local neighboring-optimal selection algorithm (LNSA) to find a minimum set of camera nodes guaranteeing the region of interest (ROI) can be full-view covered and schedule camera nodes into sleeping or working mode. In [**?**], Liu et al. studied full-view barrier coverage problem in mobile CSN. They defined full-view coverage model of mobile camera sensors. Based on this model, they divided the target field into some connected grids and used these grids to construct a weighted directed graph. The problem of building a full-view barrier with minimum number of sensors was then converted to the problem of finding the shortest path in the constructed graph by using Dijkstra algorithm.

Motivated by the purpose of monitoring the object from multiple perspectives, in [5], Tseng et al. introduced the notion of k-angle coverage. To avoid duplicating information from multiple sensors simultaneously monitoring an object, an angle constraint was added, which guaranteed two sensors cannot appear in an angle range of ω around the object (figure 2). With this constraint, the notion of $k-\omega-$angle coverage was defined. It was pointed out that if an object is $k-\omega-$angle covered, there is no angle larger than $2\pi-(k-1)\omega$ of the object that is not covered by any sensor. This means that an object that is $k-\omega-$angle covered is also full-view covered with parameter $\theta=\dfrac{2\pi-(k-1)\omega}{2}$. Hence, $k-\omega-$angle coverage can be considered as a special case of full-view coverage with the number of camera sensors covering the object is fixed. Under this new coverage model, the paper focused on addressing the problem of $k-\omega-$angle covering maximum number of objects using minimum number of sensors. The sensors can rotate around its location but cannot move. The authors proved that there have only finite possibilities that a sensor can rotate to achieve optimal coverage. Then, a greedy scheme was proposed to gradually turn each sensor into fixed position. Finally, all redundant sensors were set to low-power mode to save energy and optimize network performance.

In [4], the authors studied the problem of constructing k-ω-angle barrier using minimum number of sensors, which is referred to as minimum k-ω-angle barrier coverage problem (MkABC). The paper presented MkABC problem in two deployment scheme. Under deterministic deployment, a geometric method was proposed, which used the feature of regular polygon to construct a $(k-\dfrac{\pi}{k})$-angle barrier. When sensors are randomly deployed in the monitoring region, the MkABC becomes more difficult. In this scenario, the authors proposed a grid-based method, where each grid is judged to be $(k-\omega)$-angle covered or not. MkABC problem is then transformed to finding shortest path in graph. The algorithm used is the same as one used in [3], which has some problems as aforementioned. Besides, the accuracy level and computational cost of this method as well as generic grid-based method depends on grid size. The decreasing in grid size leads to increasing in both computational cost and accuracy level, which is not expected. There must be a trade-off between these two factors.

So far, we have discussed different types of barrier coverage such as: weak/strong barrier coverage, k-barrier coverage, full-view camera barrier, k-ω-angle camera barrier. Most of the existing works in barrier coverage focused on two main problems: (1) achieving barrier coverage, which answer the question whether a sensors deployment forming a barrier in the monitoring region and (2) minimize the number of sensors using to construct a barrier. Chen et al. [8] first mentioned the problem of measuring the quality of barrier coverage. According to that, just consider whether or not a sensor network providing barrier coverage, which is equivalent to measuring its quality as either 0 or 1, is not enough since there might be many different levels of quality. Using the notion of L-local k-barrier coverage, the authors proposed to measure the quality of k-barrier coverage for a belt region as the maximum value of L that the belt is L-local k-barrier covered. A belt region is said to be L-local k-barrier covered if every zone of length L in the region is k-barrier covered. Two other metrics that they had considered are: (1) the number of sensors needed to make the belt k-barrier covered and (2) the probability that an intruder following a randomly picked path being detected by at least k sensors. All these three metrics can effectively measure the coverage quality, however, they always provide the same result when sensor network has already achieved k-barrier coverage, i.e, the probability of detecting the intruder by k sensors is always 100%.

In this paper, we investigate the $A(k-\omega)ABC$ problem. After considering many related works, we see that previous approaches to $A(k-\omega)ABC$ problem are not yet efficient and there are rooms for improvement. Therefore, we proposed a grid-based method called **Dynamic Partition** to solve this problem. Furthermore, we want to measure the quality of object's information recorded by sensors network when it crosses the barrier. Since the metrics proposed in [8] is for k-barrier coverage model, it cannot apply to our $(k-\omega)$ coverage model. Moreover, this metrics only works when sensors network has not provided k-barrier coverage yet. In contrast, we need a metrics for measuring quality of the $(k-\omega)$ barrier, which means the barrier must have been already constructed. These prompts us to come up with a new metrics called **Differentiation Exposure**.

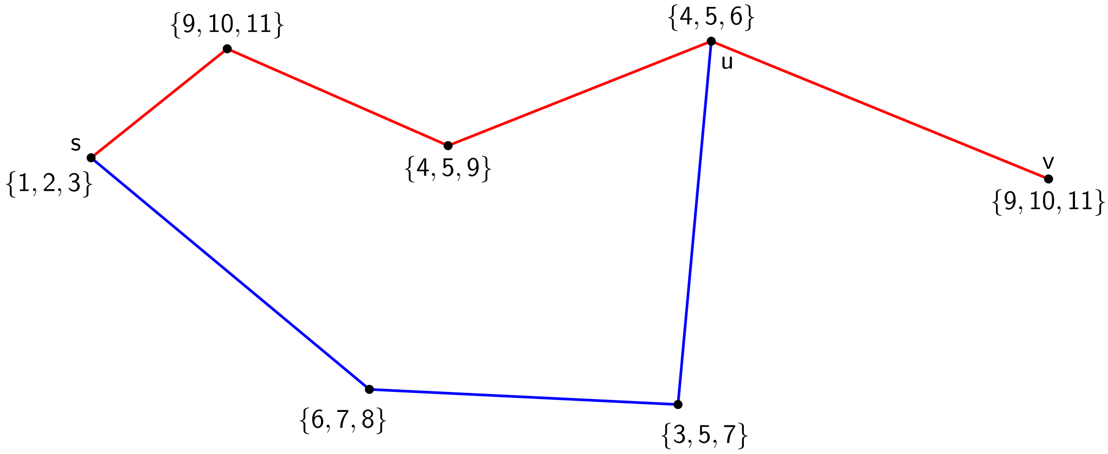


Figure 1: Red line is shortest path from s to v. Blue line is shortest path from s to u

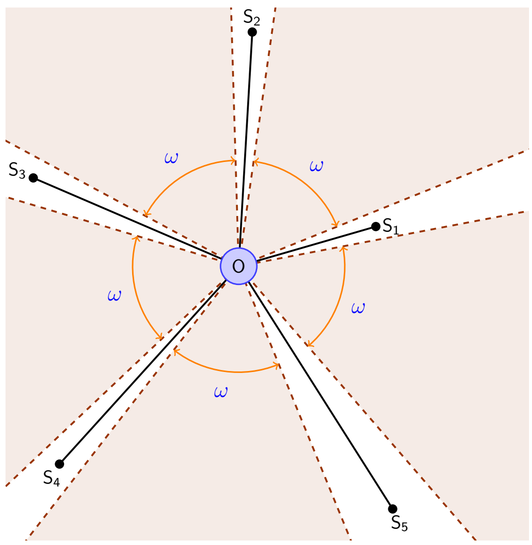


Figure 2: O is 5-ω-angle covered by $\{S_1, S_2, S_3, S_4, S_5\}$

3. Preliminaries and problem formulation

3.1. Preliminaries

3.1.1. $(k-\omega)$ region of a k-sensor list

Definition 1. *$(k-\omega)$ coverage*

- A point P is said to be $(k-\omega)$ covered if there exists a list of k sensors $L = \{S_1, S_2, ..., S_k\}$ ordered in counter-clockwise order around P, such that $\omega < (\overrightarrow{PS_i}, \overrightarrow{PS_{i+1}}) < \pi, \forall i \in \{1, 2, ..., k\}$.
- A region R is said to be $(k-\omega)$ covered if every point in R is $(k-\omega)$ covered.

Definition 2. *Inner safe region of a polygon's side.*

Given a polygon POL. Let AB is a side of POL. The way to determine the inner safe region is as follows:

- Choose the midpoint M of AB.
- Draw an arrow from M toward the inner region of POL.
- The safe region of AB consists 2 symmetrical parts splitted by line AB. The part that the arrow points to is the inner safe region of AB.

Figure 3 is an illustration of inner safe region of S_1S_2.

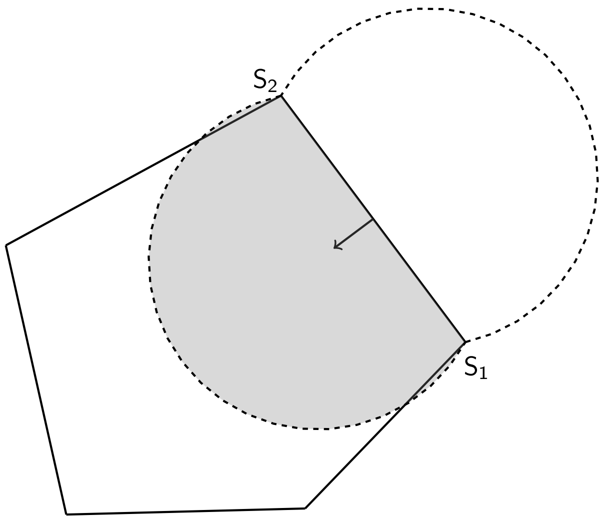


Figure 3: inner safe region of S_1S_2

Definition 3. *$(k-\omega)$ region of an k-sensor ordered list*
Given a k-sensor ordered list $L = \{S_1, S_2, ..., S_k\}$. $(k-\omega)$ region of L is the locus of points that are $(k-\omega)$ covered by L.

Theorem 1. *$(k-\omega)$ region of an k-sensor ordered list $L = \{S_1, S_2, ..., S_k\}$ is the intersection of inner safe region of every line segment S_iS_{i+1} and the sensing range of all sensors in L.*

Proof. Let R denote the $(k-\omega)$ region of L. We only consider the case that R is not empty. Suppose that P is a point inside R, then P is $(k-\omega)$ covered by L. From definition 1, we have $(\overrightarrow{PS_i}, \overrightarrow{PS_{i+1}}) > \omega$ (1) and $(\overrightarrow{PS_i}, \overrightarrow{PS_{i+1}}) < \pi$ (2), $\forall i \in \{1, 2, ..., k\}$. Condition (1) means that P is inside the safe region of S_iS_{i+1}. This safe region has 2 symmetrical parts splitted by line S_iS_{i+1}. Condition (2) forces P to be located at only one special part which satisfies $(\overrightarrow{PS_i}, \overrightarrow{PS_{i+1}}) < \pi$. And this part is always the inner safe region of S_iS_{i+1}. Hence, P is inside the intersection of inner safe region of every line segment $S_iS_{i+1},\ i = \overline{1,k}$ and the sensing region of all sensors in L (call this intersection $\mho$) (*).

On the other hand, if P is inside $\mho$, P satisfies the condition (1) and (2). Thus, P is $(k-\omega)$ covered by L, which deduce to P is inside R (**).

From (*) and (**), we have theorem 1 proved.

Theorem 2. *$(k-\omega)$ region of an k-sensor ordered list $L = \{S_1, S_2, ..., S_k\}$ is a convex region.*

Proof. The intersection of two convex regions is a convex region. Hence, the intersection of any limited sets of convex region is a convex region. From theorem 1, it is obviously that $(k-\omega)$ region of a list L is intersections of convex regions and thus, we have theorem 2 proved.

Figure 2 is an illustration of this theorem. By that, the blue hatching area is the $(k-\omega)$ region of $L = \{S_1, S_2, S_3, S_4\}$ and $\omega = 60$. We provide some preliminaries used in the following sections of this paper.

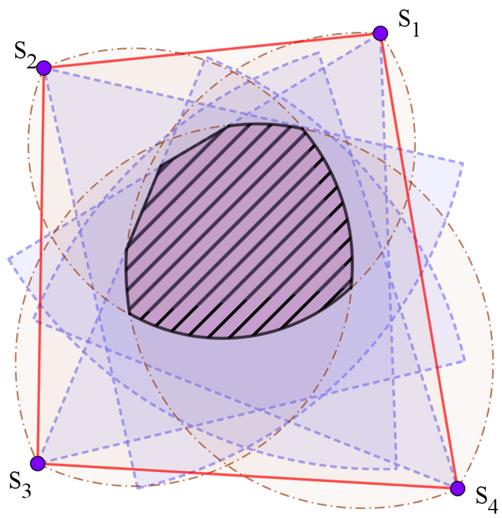


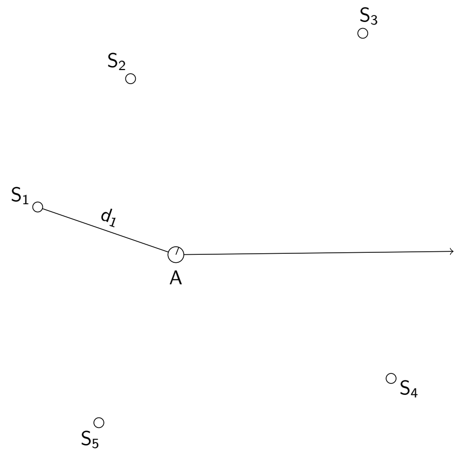
Figure 4: $(k-\omega)$ region of $L = \{S_1, S_2, S_3, S_4\}$ and $\omega = 60$

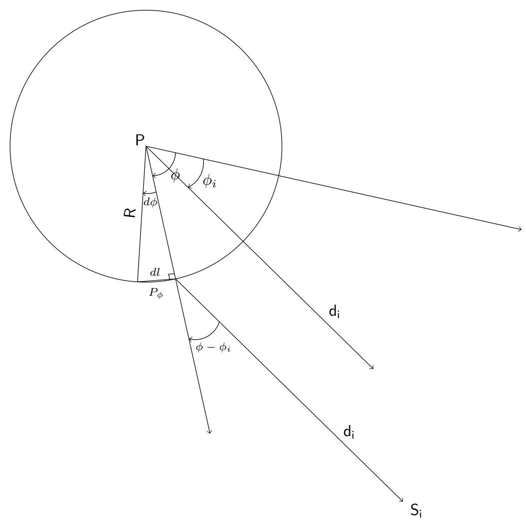
3.1.2. *Differentiation Coverage Model*

In the tradition model of attenuation, the algorithm evaluating the coverage from a sensor network toward a point may lead to several exceptional inconsistency, especially in the problem of $k-\omega$ barrier in the sensor network. As a result, in this article we are going to devise a more preferable attenuated model assessing the coverage quality of the sensor network toward a point in the field of interest, the model can later be generalised to evaluate the coverage of the sensor network on a line or a closed region.

Consider a point lying in the sensing range of a certain omnidirectional or directional sensor. Let us name the sensor S_i, the penetration object P with radius R and the considered part P_ϕ being positioned at

ϕ and has length dl. Call the distance from S_i to P_ϕ as d_i, the direction of the sensor compared to the pivot direction as ϕ_i. Because R is usually inconsiderable compared to d_i, we can approximately use ϕ_i and d_i as constants with variable ϕ, the model will be now illustrated as followed.





Firstly, the coverage value of a sensor to a part is directly affected by the distance between the sensor and the part, and the angle at which the part is viewed by the sensor, this results in the formula:

$$\max(\frac{A\cos(\phi-\phi_i)}{Rd_i^\lambda}, 0)dl \tag{1}$$

with $\frac{A}{R}$ is a constant coefficient, note that the coverage would fall below 0 if the angle between the direction of the part and the direction of the sensor is larger than $\frac{\pi}{2}$, so we need to set it to 0 in that case. Rewrite $dl = Rd\phi$, we have:

$$\max\left\{\frac{A\cos(\phi-\phi_i)}{d_i^\lambda},\ 0\right\} d\phi \tag{2}$$

However, it is obviously unnecessary to obtain too much detailed information from the object in the sensing field. This leads to the existence of a constant E_{max} which corresponds to the maximum necessary coverage on a part with unit length of the circle. To isolate the value from the relative constant A, we rewrite it to the Minimum sensing radius $E_{\mathsf{max}} = \frac{A}{d_{\mathsf{min}}^\lambda}$. As a result, our coverage formula could be rewritten as:

$$\max\left\{0,\ \min\Big\{\frac{A}{d_{\mathsf{min}}^\lambda}, \frac{Acos(\phi-\phi_1)}{d_i^\lambda}\Big\}\right\} d\phi \tag{3}$$

As a result, the coverage on a part P_ϕ of several sensors S_i, as illustrated above, is the maximum of the coverage on that part of every covered sensor:

$$E_\phi(P) = \max\left\{0,\ \min\Big\{\frac{A}{d_{\mathsf{min}}^\lambda},\ \max_{S_i}\{\frac{A\cos(\phi-\phi_i)}{d_i^\lambda}\}\Big\}\right\} d\phi \tag{4}$$

In short, the coverage on a part of a certain set of sensors is calculated from the largest value of $\dfrac{Acos(\phi-\phi_1)}{d_i^\lambda}d\phi$ across all sensors, the result then will be a value in the close interval $[0, E_{\mathsf{max}}d\phi]$ that closest to the above computed value. The total coverage on the intruder is the sum of the coverage on its small parts. Combined with the differentiated form of the formula above, the total coverage would be the integral on all of its parts. As a result, we receive the formula for the total coverage on the intruder at a certain point in the sensing field:

$$E(P) = \int_{0}^{2\pi} \mathsf{max}\left\{0,\ \mathsf{min}\left\{\frac{A}{d_{\mathsf{min}}^{\lambda}},\ \max_{S_i}\left(\frac{A\cos(\phi-\phi_i)}{d_i^{\lambda}}\right)\right\}\right\} d\phi$$

In conclusion, a new model of coverage is devised which may prove to be exceptionally effective in measuring the coverage efficiency of sensor networks in not only the tradition coverage problem but also in more complex ones such as the problem of $full\ view$ or $k-\omega$ coverage. The new model is proposed with detailed and precise logical progress, successfully adapts the strong points of both the All-Sensor Field Intensity and the Closest-Sensor Field Intensity model, handling preferably the cooperation of multiple sensors in the network without overrating the repetition of captured information.

3.1.3. Coverage of a barrier

Definition 4. $(k-\omega)$ *barrier*
A $(k-\omega)$ barrier B is a connected region from the left side to the right side of the monitoring region and satisfies that B is $(k-\omega)$ covered.

Definition 5. $(k-\omega)$ *barrier coverage*
A region achieves $(k-\omega)$ barrier coverage if there exists a $(k-\omega)$ barrier in that region.

A $k-\omega$ barrier (a barrier) is a region connects the left and right side of the sensing field in which all the points are $k-\omega$ covered. Typically, a barrier is fairly narrow, and penetration objects usually intersect the barrier only at a small part on their paths. As a result, a proper metric to assess the efficiency of the barrier would be the coverage density of it.

With the same set of sensors considered, in the range of coverage of all elements of that set, the coverage function is always continuous. Since a barrier is consisted of several separated parts each of which is $k-\omega$ covered by a common set of sensors, the coverage density of the barrier can be defined as the quotient of the total coverage in the barrier and the area of that area, with the total coverage being formulated as the integral of the coverage function over the barrier region. Call the barrier region B with area S_B, the coverage density over B, which is D_B can be formulated as

$$E(B) = \iint_B E(x,y)dxdy.\frac{1}{S_B} \tag{5}$$

3.2. Problem formulation

3.2.1. Verify the $(k-\omega)$ barrier cover

The problem is formulated as following. Given a set of n sensors $S = \{S_1, S_2, ..., S_n\}$ and a rectangular region Ω with the length of L and the width of W. Ω is called the monitoring region and camera sensors in S are deployed according to uniform deployment scheme in Ω to serve the purpose of observation in Ω. The uniform deployment scheme means that total n sensors are deployed randomly, uniformly and independently.

The objective of the problem is to verify if the monitoring region Ω achieves $(k-\omega)$ barrier coverage. In other words, we need to determine if there exists a $(k-\omega)$ barrier B in the monitoring region Ω. If there is none, Ω will not guarantee security requirements and the sensors need to be re-deployed.

Unlike scalar sensor, which only provides information about detection of the object, camera sensor is directional sensor and can be used to obtain multimedia information of the object. Each camera sensor can be denoted by a 4-tuple $\{S_i, R, \alpha, \varphi_i\}$, where S_i is the location of sensor i, R is the sensing radius and α is half of the sensing angle. We assume that all sensors have the same sensing radius and sensing angle. In reality, sensing range of camera sensor is usually less than π, so we also have an assumption that $\alpha < \frac{\pi}{2}$. The last parameter of a camera sensor, φ_i, is the facing direction of sensor i, which is uniformly distributed in $[0, 2\pi]$

Figure **??** shows information of sensor S_i.

The input and output of the problem are followings:

Input

- L, W: Length and width of the monitoring region Ω respectively.
- n: Number of camera sensors.
- $S = \{S_1, S_2, ..., S_n\}$: Set of camera sensors. S_i denotes the i-th camera sensor and also denotes the location of that sensor.
- R: Radius of camera sensors.
- α: Half of the sensing angle of camera sensors.
- φ_i: Orientation angle view of S_i where $i = \overline{1, n}$.
- k, ω : The conditional parameter of the problem.

Output

- The yes/no answer that the monitoring region achieves $(k - \omega)$ barrier coverage.

3.2.2. Evaluate the quality of a $(k - \omega)$ barrier

The problem is formulated as following. Given a barrier B in a sensing field containing several connected regions B_i from the left to the right edge of the field. Each B_i is a closing field that is $(k - \omega)$ covered by a k-list of sensors P_i.

The objective of the problem is to find the coverage of the barrier regarding our devised metric. The process is to assess the quality of the found barrier and compare the result with other settings of parameters to analyse the effect of each parameter to the quality of the sensing field and find the best combination of settings to achieve our desire.

The input and output of the problem are followings:

Input

- $\{B_i\}$: The set of closing region connect the left and the right edge of the sensing field.
- $\{P_i\}$: The set of k-list of sensors, the P_i is known to $(k - \omega)$ cover the region B_i.

Output

- The coverage value of the $(k - \omega)$ barrier.

4. Proposed algorithm

4.1. *Verify the $(k-\omega)$ barrier cover*

To solve this problem, the monitoring region is partitioned into several small rectangles using the proposed Dynamic Partition method. After that, we try to figure out if there is a continuous barrier from the left side to the right side of the region consisting of rectangles which are $(k-\omega)$ covered. The details are shown in the subsequent sections.

4.1.1. $(k-\omega)$ *verification on a rectangle*

Based on theorem 2, we can conclude that in order to verify the $(k-\omega)$ coverage on a rectangle, it is only necessary to check whether all four vertices of that rectangle is $(k-\omega)$ covered by an common list of sensors. Using this conclusion, we propose an algorithm to verify whether a rectangle is $(k-\omega)$ covered or not, and for the optimisation of the second problem, we try to find the list that $(k-\omega)$ cover the rectangle with the largest coverage value toward it. The idea of our algorithm can be described as follows:

Step 1: First, we find a set of sensors G that cover four vertices of the rectangle $ABCD$.

Step 2: Choose a list of k sensors from G satisfies that the point A is $(k-\omega)$ covered by these k sensors.

Step 3: If found any, check all of those lists and take the ones that also $(k-\omega)$ cover the 3 remaining vertices of the considered rectangle.

Step 4: Among all the list taken, return the one with the largest coverage value toward the considered rectangle.

The key to implement this idea is at step 2. Our approach to this problem is very natural. First, sort G in counter-clockwise order around A. Then, we consider each sensor in G sequentially. If the sensor being considered satisfies some conditions, we put it into a list (call this list L). We do that until size of L is equal to k. Then, L is called a valid list. Figure 5 illustrates how to choose a valid list. In figure 5, black vector denotes the sensor that is chosen to put into the list, while red vector denotes candidates to be chosen.

As aforementioned, when considering a sensor in G, it must satisfies some conditions to become candidate to be put into the list L. Suppose that at some point of the finding process, the list has *index* elements and *L[index]* $=$ *G[cur]*, $1 \leq$ *cur* $\leq n$, n is size of G. If *G[next]* is chosen to be the next element in L, it must satisfy two conditions:

- $(\overrightarrow{PG[\mathsf{cur}]}, \overrightarrow{PG[\mathsf{next}]}) > \omega \qquad (1)$
- $(\overrightarrow{PG[\mathsf{next}]}, \overrightarrow{PL[1]}) > (k - \mathsf{index})\omega \qquad (2)$

From definition of $(k-\omega)$ coverage, condition (1) is clearly necessary. However, it's not sufficient for *G[next]* to become a candidate for the next position in L. If L is a valid list, we have $(\overrightarrow{PL[i]}, \overrightarrow{PL[i+1]}) > \omega, i = \overline{1,k}$ (consider $k+1=1$). Hence, $(\overrightarrow{PL[\mathsf{index}+1]}, \overrightarrow{PL[1]}) = \sum_{i=\mathsf{index}+1}^{k} (\overrightarrow{PL[i]}, \overrightarrow{PL[i+1]}) > (k-\mathsf{index})\omega$. Since we are choosing candidate for $(\mathsf{index}+1)$-th element in L, $G[\mathsf{next}]$ corresponds to $L[\mathsf{index}+1]$. Thus, (2) is also a necessary condition.

Algorithm 1 and Algorithm 2 shows the details of our algorithm. The function in Algorithm 2 is a support function for Algorithm 1. `RecurFinding(cur)` is a recursive function that finds candidate for $(\mathsf{index}+1)$-th position in L knowing that there is a set cur containing the chosen sensors, in which the last sensor has index

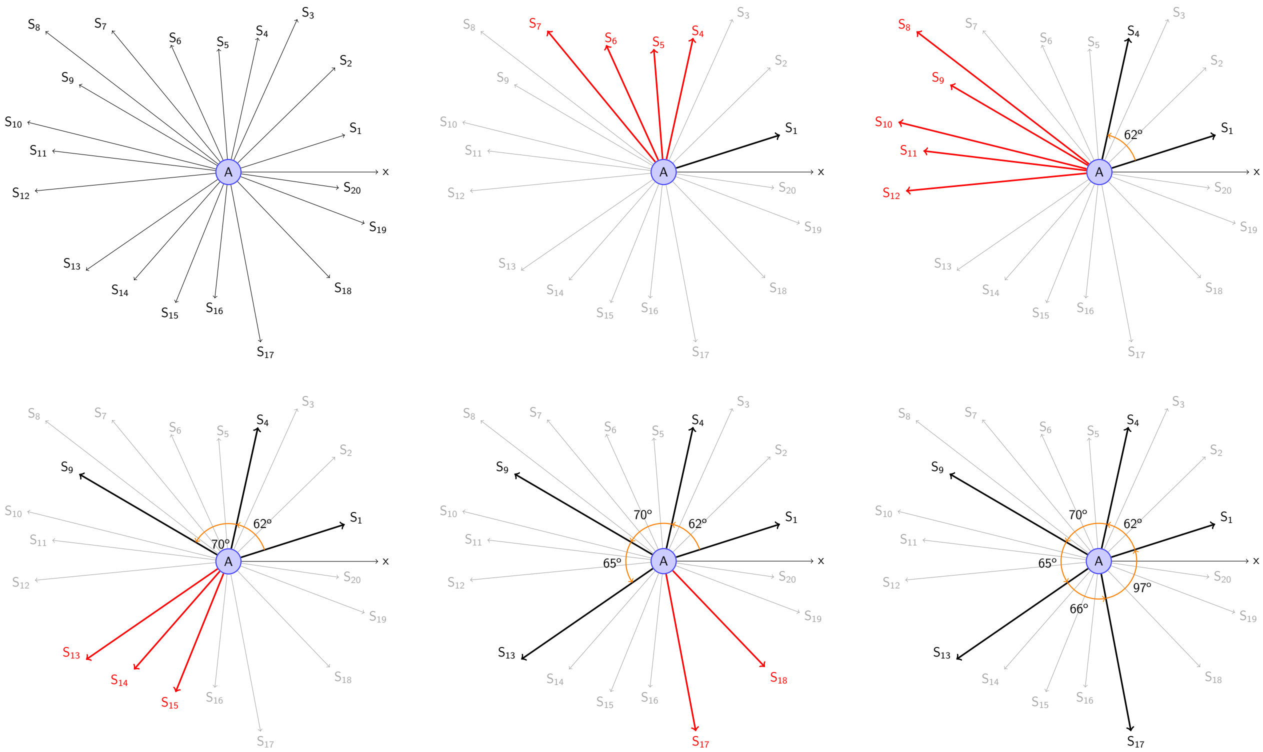


Figure 5: $L = \{S_1, S_4, S_9, S_{13}, S_{17}\}$ is a valid list with $k = 5, \omega = 60^o$

last. It considers elements in G sequentially from (last + 1)-th element and checks if these elements satisfy condition (1) and (2). The return value of `RecurFinding(cur)` is a set containing all the lists of sensors that $(k-\omega)$ cover the point P which takes the current sublist $(\{L[1], L[1], ..., L[\text{index}]\})$ as its first index elements exists. This return value is used to support the recursion process of the algorithm.

Algorithm 1: Find all lists of k sensors that $(k-\omega)$ covers point P

Input: A point P and a set G consisting of n sensors that cover P.
Output: k sensors that $(k-\omega)$ covers P. There is possibility that no output is found.

Let L store the output
Sort G in counter-clockwise order around P
$L \leftarrow \emptyset$
for $i = 1$ **to** n **do**
    $temp \leftarrow$ `RecurFinding(Gi)`
    $L \leftarrow L \cup temp$
end for

Algorithm 2: Find index + 1 element in L

Input: A list of sensors that is currently chosen, contains index sensors.
Output: All lists of k sensors that $(k-\omega)$ cover the point P beginning with the input list.

`RecurFinding(cur)`
    $L \leftarrow \emptyset$
    index $\leftarrow$ cardinality of cur
    if $m == k$ **then**
        return {cur}
    end if
    $last \leftarrow$ index of last element in cur
    $first \leftarrow$ index of first element in cur
    for $i = last + 1$ **to** n **do**
        if $(\overrightarrow{\mathsf{PG}[last]}, \overrightarrow{\mathsf{PG}[i]}) > \omega$ && $(\overrightarrow{\mathsf{PG}[i]}, \overrightarrow{\mathsf{PG}[first]}) > (\mathsf{k} - \mathsf{index} + 1)\omega$ **then**
            $temp \leftarrow$ `RecurFinding(cur + (G[i]))`
            $L \leftarrow L \cup temp$
        end if
    end for
    return L
end

The first element of L can be any sensor in G since it doesn't require any condition. For convenient, we choose $L[1] = G[1]$. Thus, `RecurFinding({G[1]})` is called to start the finding process. After function call `RecurFinding({G[1]})`, the function will return all the satisfied lists containing G[1]. The finding process stops when we have called `RecurFinding({G[i]})` with every i from 1 to n. And the algorithm will output a set containing all the lists of sensors that $(k-\omega)$ cover the considered point P.

4.1.2. Finding a barrier in a monitoring region

a, Partitioning the monitoring region by Dynamic Partition method

To find a barrier in the monitoring region, we first determine the areas that are $(k-\omega)$ covered inside the monitoring region. To solve this problem, we partition the monitoring region into multiple small rectangles and check whether these rectangles are $(k-\omega)$ covered or not. However, uniform partitioning often requires a high computation time especially when the monitoring region is large. To overcome this challenge, we propose a new partition method called Dynamic Partition. The idea is: only the rectangle regions which are not $(k-\omega)$ covered will be partitioned into smaller rectangles, otherwise, they are kept untouched. The first rectangle to be checked is the monitoring region. Using the algorithm in subsection 1, if a rectangle is $(k-\omega)$ covered, mark it as true, otherwise, split it into four equal sub-rectangles. After a rectangle is split, smaller rectangles are generated and the process of checking and splitting is applied to these new rectangles. A rectangle will not be split if it is $(k-\omega)$ covered or its size reaches a predefined limited value. The smaller the limited size is, the more precise the result of our algorithm gets. This condition guarantees our algorithm not to go into an infinite loop. The process is illustrated in Figure 6.

The pseudo code of the method is described in Algorithm 3

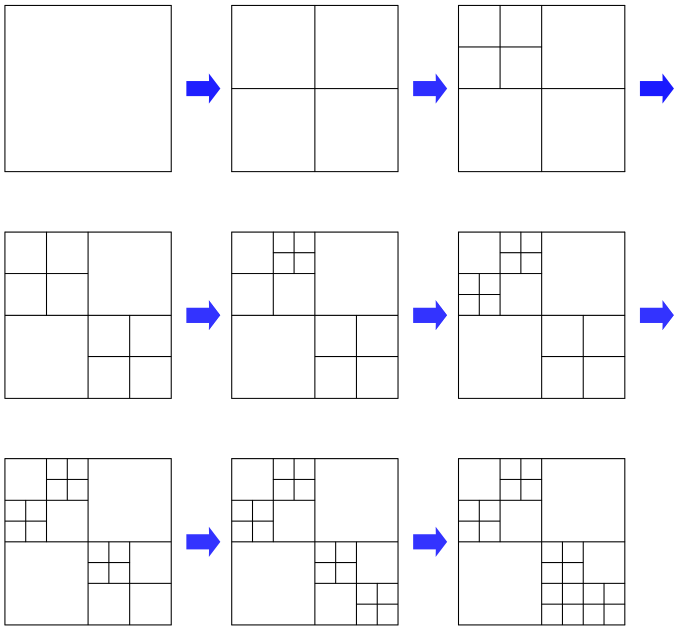


Figure 6: Illustration of Dynamic Partitioning

Algorithm 3: Dynamic Partition

Input:

- Length and width of the monitering region Ω: L, W respectively
- A set of sensors S
- Maximum depth of quad-tree: DMAX

Output: A set of covered rectangles

Let coveredRectangles store the output
Let rootRec denote the monitoring region
Q $\leftarrow \emptyset$
add rootRec to Q
while Q $\neq \emptyset$ **do**
  tempRec $\leftarrow$ dequeue(Q)
  Check if tempRec is $(k - \omega)$ covered
  if tempRec *is* $(k - \omega)$ *covered* **then**
    add tempRec to coveredRectangles
  else if tempRec.rank $<$ DMAX **then**
    split tempRec in into 4 sub-rectangles
    add 4 sub-rectangles of tempRec to Q
  end if
end while

b, Finding a $(k - \omega)$ coverage barrier

After procedure in 4.1.2, we now have a set R_{cover} of rectangles that are $(k-\omega)$ covered. To find a $(k-\omega)$ coverage barrier, we need to find a continuous area formed from rectangles in R_{cover} that connects the left side to the right side of Ω. The method is to transform the rectangles set into a graph. Each vertex in the graph corresponds to a rectangle in R_{cover}. Two vertices are considered adjacent if the corresponding rectangles share at least one point. Two virtual vertices are added to the graph, source vertex s and sink vertex t. All vertices corresponding to the rectangles lying on the left side of Ω are adjacent to s and all vertices corresponding to the rectangles lying on the right side of Ω are adjacent to t. After the graph is constructed, we use Breath First Search algorithm to find a path from s to t. If a path is found, we conclude that there exists a $(k-\omega)$ barrier in the monitoring region. Otherwise, the barrier does not exist.

4.2. Evaluate the quality of a $(k-\omega)$ barrier

The algorithm takes the nodes forming a barrier and the k-list of sensors associating with each node as the input and compute the coverage on the input barrier.

The coverage of the barrier is calculated as the average of every node which forms that barrier with the weight assigned as the area of each node. With B_i as the nodes forming the barrier B, we have

$$E(B) = \iint_B E(P)dxdy.\frac{1}{S_B} =$$

$$(\sum_i(\iint_{B_i} E(P)dxdy)).\frac{1}{S_B} =$$

$$(\sum_i E(B_i).S_{B_i}).\frac{1}{S_B} =$$

$$(\sum_i E(B_i).S_{B_i}).\frac{1}{\sum_i S_{B_i}}$$

As a result, this calculation method is consistent with our definition of coverage on the barrier in the 3.1.3 section, hence may provide preferable assessment on each setting of parameters.

Since it is impossible and unnecessary to compute the exact coverage value of each note, it is sufficient to publish a method to estimate an approximation of the coverage on the considered node. Take into account the fact that in each node is $(k-\omega)$ covered by an unique k-list of sensors, hence the coverage value inside the node is a continuous function. As a result, we can create a dense grid in each node, and estimate the node coverage with the average of the vertices on the discrete grid. For convenience, the size of the grid is fixed to be the size of the node which will not be split further in 4.1.2.

5. Experimental results

5.1. Simulation method

This part will analyse the effect of several parameters to 3 aspects of the result, which is the probability of creating barrier, the average exposure along the barrier and the overall computational time. The algorithm is performed on every instance and keep recording 3 data, the creation of barrier, the computation time and the exposure on a barrier if there is one. Then, the result are combined for all instances of the same parameter settings to achieve the probability of barrier creation, the average computation time and the average exposure on the found barriers.

5.2. System settings

All the experiments are performed on a personal computer with core Intel Core i7-7700HQ, 8GB of RAM running Windows 10 Home, the programming language used to simulate the algorithm is Java 11.

5.3. Parameter settings

The sensing fields in all experiments are presented as rectangles with the width of 50m and the length of 200m. Sensor nodes are deployed uniformly randomly in a rectangle with each side extended compared to the sides of the sensing field a distance equal to the sensing radius of each sensor to guarantee the uniform distribution regarding sensing area inside the sensing field. Each set of parameter contains several independent random networks to conduct the algorithm on and measure the probability of achieving barrier for each circumstance. Altogether there are 150 instances of parameters with 100 experiment on each instance which were analysed with our algorithm. The details are given below:

Length	200
Width	50
Sensing Radius	30
Minimum sensing radius	5
Sensing angle	90

Table 1: General parameters

k	3
ω	85 - 110
Sensor number	250 - 650

Table 2: Problem parameter with $k = 3$

k	3
ω	85 - 110
Sensor number	250 - 650
Experimental field	100

Table 3: Problem parameter with $k = 3$

k	4
ω	50 - 80
Sensor number	250 - 600
Experimental field	100

Table 4: Problem parameter with $k = 4$

k	5
ω	45 - 65
Sensor number	250 - 750
Experimental field	100

Table 5: Problem parameter with $k = 3$

k	6
ω	40 - 55
Sensor number	250 - 850
Experimental field	100

Table 6: Problem parameter with $k = 3$

5.4. *Computation results*

5.4.1. *Effect of ω on algorithm performance*

ω is an important parameter in the $k - \omega$ coverage model. As a result, this parameter has a considerable impact on the output of the algorithm. Because ω is a lower bound for the angle between two consecutive sensors in the perspective of the considered point, every sensor set that satisfies the condition with large ω would also successfully make a $k - \omega$ cover with lower ω. In short, a decrease in parameter ω may result in an expansion in the result space of the algorithm. This leads to two different consequences. On the one hand, there would be more sets of sensor $k - \omega$ cover a single node, which means that the exposure of that node is likely to be lifted. However, on the other hand, the lower value of ω could reduce the average rank of the covered nodes, as the nodes are more easily covered, which leads to a lower exposure, since the sets that cover the bigger node tend to position further than the sets covering the smaller ones.

As a consequence, firstly, with a lower value of ω, the algorithm would offer a greater chance of $k - \omega$ barrier existence, and probably also a greater exposure on the obtained barriers. However, the probability of forming barriers can never exceed 100%, the curve that represents the barrier probability would approach 100% and does not rise higher with lower ω. Furthermore, a high exposure usually occurs when the node is exposed at every direction, which may satisfy the condition with high ω. Consequently, the exposure value would eventually approach a bound when the value of ω decrease.

On the other hand, a lower value of ω results in a larger searching space, which leads to a drastic rise in computation time. As a result, it is suitable to choose a sufficient ω so that the barrier probability and the barrier exposure approach its upper bound while the computation time is still acceptable.

5.4.2. *Effect of sensor number on algorithm performance*

Like the effect of ω, a larger value of sensor number would lead to a larger searching space. However, in this occasion, the negative effect on exposure seems to be more important. As a result, the barrier exposure tends to fall slowly as the sensor number rises. Furthermore, the large number of sensors leads to a huge computational work. This results in the computation time surge dramatically as the sensor number grows. As a result, it is suitable to choose a sufficient sensor number so that the barrier probability approaches 100%, while the barrier exposure has not fallen too considerable and the computation time is still acceptable.

5.4.3. *Effect of k on algorithm performance*

k is the parameter affecting the achieved results the most regarding all 3 aspects. This is because the change in k would manipulate the problem entirely, an answer with a value of k would not be an answer with another value of k. As a consequence, the achieved results are drastically different among every value of k.

As mentioned in previous parts, generally, the exposure of the barriers would not be much different from the others. As a result, we may reach a conclusion that for every value of k, it is possible to define a critical value of barrier exposure which denotes the largest achieved value of exposure for a certain value of k. And this critical exposure value could be use to compare the performance of the problem with different values of k.

Regarding this metric, in general, as there are more sensors that cover a certain point, an increase in the value of k may lead to a larger critical exposure. However, since the function of $cos(x)$ has a derivative getting lower as the value of x comes close to 0, and the effect of increasing k on decreasing the sight angle of sensors to the parts of the intruder $(\phi - \phi_i)$ may reduce with larger k. As a result, the critical exposure value will eventually reach a bound when the value of k keep climbing.

Finally, the effect of k on computation time is. This is because that the large value of k would leads to a larger nest in traversing for all the $k - \omega$ sets and larger loop when checking the exposure of nodes, hence the computation time for finding all the sets that $k - \omega$ cover each node and determining the sets of sensor with largest exposure is increased considerably.

In conclusion, there may exist a value of k such that its critical exposure approaches the upper bound while the computation time has not been exceptional.

5.4.4. Compare the devised metric with the traditional coverage models

The figure below illustrates the result of barrier coverage regarding the density computation, which means the coverage of a barrier is calculated as the average of the coverage of every point on the barrier. Practically, the method to evaluate the coverage of a barrier has been proposed in section 4.2. The assessment takes into consideration 2 attenuated models, which are the Closest-Sensor and the All-Sensor Intensity.

From the above figure, it is obvious that the differential coverage model outperform the tradition ones

6. Conclusion

This paper addressed the minimal exposure path problem for attenuated sensing model with all mobile sensors. We first have considered and formulated a model of the minimal exposure path problem in all mobile sensor networks; we have proposed the genetic algorithm to solve this issue. In addition, extensive experimental simulations were conducted to validate and to evaluate the proposed model as well as algorithm. The results showed that: the proposed algorithm could be effectively applied to both static and mobile models of wireless sensor networks; the coverage of mobile sensor networks is almost better than the coverage of static sensor networks in case having the same number of sensors.

References

[1] V. Blanz, P. Grother, P. J. Phillips, T. Vetter, Face recognition based on frontal views generated from non-frontal images, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 2, IEEE, 2005, pp. 454–461.

[2] Y. Wu, X. Wang, Achieving full view coverage with randomly-deployed heterogeneous camera sensors, in: 2012 IEEE 32nd International Conference on Distributed Computing Systems, IEEE, 2012, pp. 556–565.

[3] H. Ma, M. Yang, D. Li, Y. Hong, W. Chen, Minimum camera barrier coverage in wireless camera sensor networks, in: 2012 Proceedings IEEE INFOCOM, IEEE, 2012, pp. 217–225.

[4] B. Xu, Y. Zhu, D. Li, D. Kim, W. Wu, Minimum (k, ω)-angle barrier coverage in wireless camera sensor networks., IJSNet 21 (3) (2016) 179–188.

[5] Y.-C. Tseng, P.-Y. Chen, W.-T. Chen, k-angle object coverage problem in a wireless sensor network, IEEE Sensors Journal 12 (12) (2012) 3408–3416.

[6] D. W. Gage, Command control for many-robot systems, Tech. rep., Naval Command Control and Ocean Surveillance Center Rdt And E Div San Diego CA (1992).

[7] Y. Wang, G. Cao, Barrier coverage in camera sensor networks, in: Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing, ACM, 2011, p. 12.

[8] A. Chen, T. H. Lai, D. Xuan, Measuring and guaranteeing quality of barrier-coverage in wireless sensor networks, in: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing, ACM, 2008, pp. 421–430.

References

[1] V. Blanz, P. Grother, P. J. Phillips, T. Vetter, Face recognition based on frontal views generated from non-frontal images, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 2, IEEE, 2005, pp. 454–461.

[2] Y. Wu, X. Wang, Achieving full view coverage with randomly-deployed heterogeneous camera sensors, in: 2012 IEEE 32nd International Conference on Distributed Computing Systems, IEEE, 2012, pp. 556–565.

[3] H. Ma, M. Yang, D. Li, Y. Hong, W. Chen, Minimum camera barrier coverage in wireless camera sensor networks, in: 2012 Proceedings IEEE INFOCOM, IEEE, 2012, pp. 217–225.

[4] B. Xu, Y. Zhu, D. Li, D. Kim, W. Wu, Minimum (k, ω)-angle barrier coverage in wireless camera sensor networks., IJSNet 21 (3) (2016) 179–188.

[5] Y.-C. Tseng, P.-Y. Chen, W.-T. Chen, k-angle object coverage problem in a wireless sensor network, IEEE Sensors Journal 12 (12) (2012) 3408–3416.

[6] D. W. Gage, Command control for many-robot systems, Tech. rep., Naval Command Control and Ocean Surveillance Center Rdt And E Div San Diego CA (1992).

[7] Y. Wang, G. Cao, Barrier coverage in camera sensor networks, in: Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing, ACM, 2011, p. 12.

[8] A. Chen, T. H. Lai, D. Xuan, Measuring and guaranteeing quality of barrier-coverage in wireless sensor networks, in: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing, ACM, 2008, pp. 421–430.