

# Barrier Coverage in Camera Sensor Networks

Yi Wang and Guohong Cao  
Department of Computer Science and Engineering  
The Pennsylvania State University  
University Park, PA, 16802, USA  
{yuw124, gcdo}@cse.psu.edu

## ABSTRACT

Barrier coverage has attracted much attention in the past few years. However, most of the previous works focused on traditional scalar sensors. We propose to study barrier coverage in camera sensor networks. One fundamental difference between camera and scalar sensor is that cameras from different positions can form quite different views of the object. As a result, simply combining the sensing range of the cameras across the field does not necessarily form an effective camera barrier since the face image (or the interested aspect) of the object may be missed. To address this problem, we use the angle between the object's facing direction and the camera's viewing direction to measure the quality of sensing. An object is full-view covered if there is always a camera to cover it no matter which direction it faces and the camera's viewing direction is sufficiently close to the object's facing direction. We study the problem of constructing a camera barrier, which is essentially a connected zone across the monitored field such that every point within this zone is full-view covered. We propose a novel method to select camera sensors from an arbitrary deployment to form a camera barrier, and present redundancy reduction techniques to effectively reduce the number of cameras used. We also present techniques to deploy cameras for barrier coverage in a deterministic environment, and analyze and optimize the number of cameras required for this specific deployment under various parameters.

## Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design – *network topology*

## General Terms

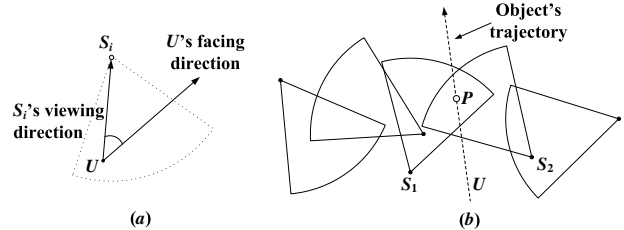
Algorithms, Theory

## Keywords

Camera Sensors, Barrier Coverage, Full-view Coverage

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiHoc'11, May 16–19, 2011, Paris, France  
Copyright 2011 ACM 978-1-4503-0722-2 ...\$10.00.



**Figure 1:** (a)  $U$  is an object; dotted line is the sensing range of  $S_i$  and  $US_i$  is the viewing direction of  $S_i$ ; (b) if  $U$  faces the direction along the trajectory,  $S_1$  and  $S_2$  are not able to view its face, although  $U$  is within their coverage.

## 1. INTRODUCTION

Barrier coverage is one of the most important issue for various sensor network applications, e.g., national border control, critical resource protection, security surveillance and intruder detection, etc. In a wireless sensor network, a barrier is formed by a set of sensors whose sensing ranges are contiguous and span (usually a strip area) across the monitored field. Every object traversing the field from one side to another is detected by the sensors on the barrier. Compared to achieving full coverage, the number of sensors required for barrier coverage is much less. Hence barrier coverage is considered more scalable and attractive for large-scale deployment in practice.

Previous studies in barrier coverage mainly focused on traditional scalar sensor networks, in which the sensing range of a sensor is often modeled as a disk and an object is said to be covered or detected by a sensor if it is within the sensing range of the sensor [14]. Recently, there has been an increasing interest in camera sensor networks [18, 10, 8, 2, 1, 22]. Compared with traditional scalar sensors, camera sensors can provide much richer information of the environment in the forms of images or videos and hence promise a huge potential in applications. However, the cost of camera sensor is considered to be higher than scalar sensor. Deploying camera sensors for achieving full coverage in a large scale is very hard if not impossible. Thus for applications like security surveillance, it is desirable to build up a cost-efficient “camera barrier” such that every intruder’s image can be detected effectively.

However, the barrier coverage of camera sensors is much different and more complicated than the traditional barrier coverage problem. Simply combining the sensing range of a series of cameras across the monitored field does not pro-

vide effective barrier coverage. This is because an intruder may cross the barrier without being identified, i.e., its *face* image could be missed (Figure 1). In fact, one fundamental difference between camera and traditional scalar sensors in coverage is that camera sensors may generate very different views of the same object if they are from different viewpoints. For example, if a camera is placed behind the intruder, no face image can be identified. Studies in computer vision 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 [4], i.e., if the object is facing straight to the camera. As the angle between the object’s facing direction and the camera’s viewing direction (defined by the vector from the object to the camera, Figure 1(a)) increases, the detection rate drops dramatically [20, 17]. Therefore, to maintain a high level surveillance quality, a good camera barrier should guarantee that no matter where the traversing object faces, there is always some camera to effectively capture its face image.

In this paper, we study the barrier coverage in camera sensor networks by leveraging the concept called full-view coverage. An object is said to be full-view covered if there is always a camera to cover it no matter which direction it faces and the camera’s viewing direction is sufficiently close to the object’s facing direction (rigorous definition is given in Section 2). An effective camera barrier is essentially a connected zone across the monitored field such that every point within the zone is full-view covered.

Based on this model, we study the problem of constructing camera barrier in both random and deterministic deployment. In practice, sensors can be either deployed randomly, e.g., being dropped from aircraft to an inaccessible zone, or deployed deterministically, e.g., being placed manually in a controlled environment. In the first scenario, we have no precise control on the positions of the camera sensors, and a post-deployment procedure is needed to select cameras to form a barrier. As mentioned above, the challenge here is that selecting cameras with overlapped sensing range does not guarantee a full-view covered barrier. For each point on the barrier, there must be a set of active cameras spreading around it and covering it from multiple viewpoints. What makes the problem even more difficult is that the camera’s viewing direction depends on the geometric relationship between the camera and the moving object, and the object’s facing direction could point to anywhere. Hence bounding the difference of the two vectors for good camera coverage on the barrier is a big challenge. The same difficulties exist in the second scenario where a deployment pattern should be devised such that a line across the field is full-view covered by the deployed cameras. Note that this is straightforward in traditional disk sensing model since we can place sensors one by one along the line with each one’s range overlapping with its neighbors’. However, these solutions can not be applied for camera barrier due to these new challenges.

The main contributions of this paper are as follows. First, we define the camera barrier coverage problem based on the full-view coverage model. It characterizes the unique requirement of a good barrier coverage in camera sensor networks in practice. Second, we provide a series of procedures to select camera sensors in an arbitrary deployment to form a camera barrier that can effectively detect the intruder’s face image, and introduce a method to reduce the camera redundancy. Third, we devise a deployment pattern that

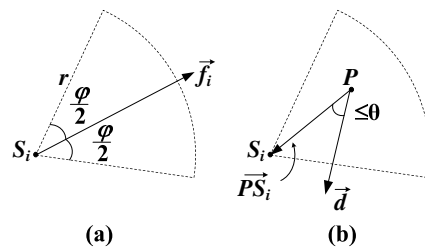


Figure 2: The full-view coverage model.

can achieve camera barrier coverage in a deterministic deployment. We analyze and optimize the number of cameras used for this specific deployment under various camera parameters. Finally, we validate our results by extensive simulations.

The remainder of this paper is organized as follows. Section 2 introduces the full-view coverage model and the concept of camera barrier coverage. Section 3 gives the detailed description on how to select cameras to form a camera barrier in an arbitrary deployed camera sensor network. Section 4 shows the details of a deployment pattern for camera barrier coverage in a deterministic deployment and analyzes the number of cameras needed in this deployment. Section 5 presents the evaluation results. The related work is reviewed in Section 6 and the paper is concluded in Section 7.

## 2. NOTATIONS AND MODEL

Camera sensors<sup>1</sup> are deployed to monitor a bounded region  $A$  (target field). Each sensor has a sensing range  $r$ , a field-of-view (FoV) angle  $\varphi$  and an orientation vector  $\vec{f}_i$ , which together define the sensing sector (Figure 2(a)). We use  $S_i$  to denote the  $i$ -th sensor. Without ambiguity,  $S_i$  also denotes the sensor’s position. For any two points  $U, V$ , let  $\|UV\|$  denote the (Euclidean) distance between them. For any two vectors  $\vec{v}_1$  and  $\vec{v}_2$ , let  $\alpha(\vec{v}_1, \vec{v}_2)$  denote the angle between them, which ranges from 0 to  $\pi$ . A point  $P$  is covered by a sensor  $S_i$  if  $P$  is in the sensing sector of  $S_i$ , i.e.,  $\|PS_i\| \leq r$  and  $\alpha(\vec{f}_i, \overrightarrow{S_iP}) \leq \varphi/2$ , where  $\overrightarrow{S_iP}$  denotes the vector from  $S_i$  to  $P$  (Figure 2(b)).

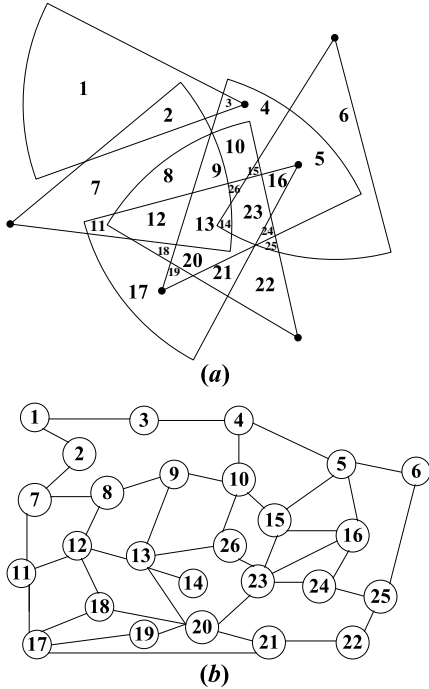
**Definition 2.1** (Full-View Coverage) A point  $P$  is full-view covered if for any facing direction (i.e., any vector  $\vec{d}$ ), there is a sensor  $S_i$ , such that  $P$  is covered by  $S_i$  and  $\alpha(\vec{d}, \overrightarrow{S_iP}) \leq \theta$ , where  $\theta$  ( $\in [0, \pi/2)$ ) is a predefined parameter which is called the *effective angle*. A region is full-view covered if every point in it is full-view covered.

Based on this concept, we introduce the following definition of what a good barrier coverage in a camera sensor network should be.

**Definition 2.2** (Camera Barrier) Given a rectangular field  $A$  with one side being the entrance and the opposite side being the destination, a camera barrier  $B$  is a connected region inside  $A$  such that  $B$  is full-view covered and every path from one point on the entrance side to another point on the destination side intersects with  $B$ .

For convenience, we assume the entrance side is at the bottom and the destination side is on the top.

<sup>1</sup>We may interchangeably use cameras or sensors for short throughout the paper.



**Figure 3:** (a) The plane is partitioned into sub-regions; each sub-region is identified by a number; (b) the graph is constructed based on the relationship among the sub-regions; the number on the node indicates the corresponding sub-region in (a).

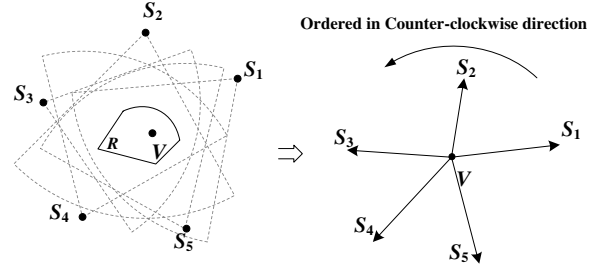
### 3. CAMERA SELECTION FOR BARRIER COVERAGE IN RANDOM DEPLOYMENT

In practice, camera sensors can be deployed randomly in a target area and hence we do not have precise control over the position of each camera. Even if the initial deployment is controlled, sensors' failure due to energy or other factors can still make the network's topology unpredictable. In this section, we propose a method to select camera sensors from an existing and arbitrary deployment to form a camera barrier.

As mentioned in the introduction, simply selecting cameras across the field with connected sensing range does not necessarily form a camera barrier. We need to guarantee that each point of the barrier is full-view covered. This is the key challenge here. We approach this problem by first converting the monitored field into a graph (discretization) in which each node represents a small sub-region and two nodes are connected if they are adjacent in the original field. By doing this, we can verify the coverage quality of each sub-region and determine a subset of nodes (sub-regions) that are full-view covered. Then we find a path from the left boundary to the right boundary, consisting of nodes that are full-view covered. This path represents a set of contiguous sub-regions across the field, which is essentially the camera barrier we are looking for. We also utilize some redundancy reduction techniques to effectively reduce the number of cameras in use. We show the details in the subsequent sections.

#### 3.1 Discretization

Given a set of deployed sensors, field  $A$  can be partitioned



**Figure 4:** The circular list of  $V$  is defined based on the positions of the cameras covering  $V$ .

into sub-regions, where each sub-region is defined to be a set of points covered by the same set of sensors. Two sub-regions are adjacent if they share at least one common boundary, which can be a line or arc segment from the boundary of the sensing range of some sensors. We model all the sub-regions and their relationship to each other by a graph  $G = (V, E)$ . Each node in  $V$  represents a sub-region. There is an edge  $(i, j)$  between node  $i$  and  $j$  if and only if they are adjacent sub-regions. An example of this graph is shown in Figure 3.

Two virtual nodes  $s$  and  $t$  are then added into this graph. They represent the left and right boundaries of field  $A$  respectively. There is an edge  $(s, i)$  between node  $s$  and  $i$  if sub-region  $i$  intersects with the left boundary of  $A$ . Similarly there is an edge  $(j, t)$  if sub-region  $j$  intersects with the right boundary of  $A$ .

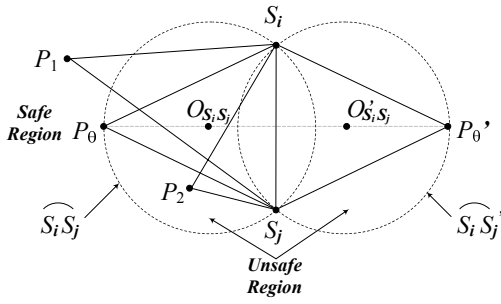
The number of sub-regions in  $G$  is  $O(n^4)$ , where  $n$  is the total number of cameras. The reason is as follows. We can consider the field  $A$  as a planar graph, where the vertices are the crossing points of sensing sectors and edges are the line or arc segments between any two crossing points. Since any two sensing sectors can have  $O(1)$  crossing points on their perimeters, the number of vertices is  $O(n^2)$ . This further implies the total number of edges is  $O(n^4)$ . From Euler's formula [3], the number of faces, i.e., sub-regions, is thus equal to  $2 - O(n^2) + O(n^4)$ , which is  $O(n^4)$ .

#### 3.2 Full-view Coverage Verification on Sub-regions

For a given sub-region  $R$ , we need to verify if the condition in Definition 2.1 holds for every point in it. Note that all points in  $R$  are covered by the same set of sensors. In this section, we only consider this set of sensors. Since  $R$  is always within their FoV, we can ignore their orientation vectors (i.e.,  $\vec{f}_i$ ). What really matters here is the position of each camera and the geometrical relationship between them.

For any given point  $V \in R$ , we define a *circular list* of these sensors regarding their viewing direction on  $V$  as follows (Figure 4). Initially the list is empty. We begin with any vector  $\vec{VS}_i$  and place it into the list first. Then we rotate  $\vec{VS}_i$  around  $V$  in the counterclockwise direction until it becomes parallel to the next vector  $\vec{VS}_j$ . Then we place  $\vec{VS}_j$  into the list, right after  $\vec{VS}_i$ . We continue rotating and place the vectors sequentially into the list until we meet the beginning vector again. Then the list is completed. The list is denoted by  $CL_V = \{\vec{VS}_{V_1}, \dots, \vec{VS}_{V_k}\}$ , where  $k$  is the number of sensors covering  $R$ .

In such a list, each element has a "next" pointer pointing to the element right after it. The "next" pointer of the last



**Figure 5: The safe and unsafe region of  $S_i, S_j$ ; here  $\angle S_i P_1 S_j > \angle S_i P_0 S_j = \angle S_i P_0' S_j > \angle S_i P_2 S_j$ .**

element, which is  $\overrightarrow{VS_{V_k}}$ , points to the first element  $\overrightarrow{VS_{V_1}}$ . Two lists are considered the same if they have the same set of elements and the “next” pointer of each element points to the same element in both sets. By using the concept of circular list, the condition in Definition 2.1 is equivalent to the following.

**Lemma 3.1** A given point  $V$  is full-view covered if and only if for  $CL_V$  constructed as above, the rotation angle from  $\overrightarrow{VS_{V_i}}$  to  $\overrightarrow{VS_{V_{i+1}}}$  is less than or equal to  $2\theta$  for any  $1 \leq i \leq k$ , where  $V_{k+1} = V_1$ .

PROOF. Suppose the condition holds. Then for any  $\vec{d}$ , there are two sensor  $S_{V_i}$  and  $S_{V_{i+1}}$  such that either the rotation angle from  $\overrightarrow{VS_{V_i}}$  to  $\vec{d}$  or the angle from  $\vec{d}$  to  $\overrightarrow{VS_{V_{i+1}}}$  is less than or equal to  $\theta$ . Thus  $V$  is full-view covered.

Suppose  $V$  is full-view covered but the rotation angle from  $\overrightarrow{VS_{V_i}}$  to  $\overrightarrow{VS_{V_{i+1}}}$  is larger than  $2\theta$  for some  $i$ . Then, consider vector  $\vec{d}$  along the bisector of the angle, the angle between either  $\overrightarrow{VS_{V_i}}$  or  $\overrightarrow{VS_{V_{i+1}}}$  and  $\vec{d}$  is larger than  $\theta$ . Therefore the condition is true.  $\square$

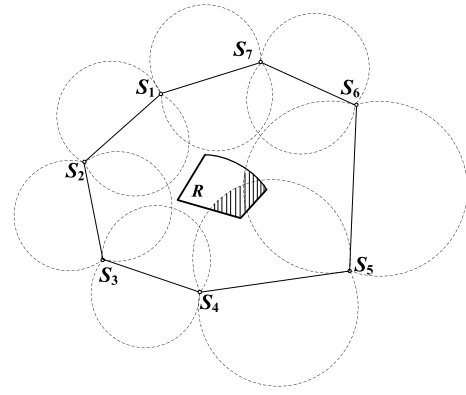
We need to determine if the above condition holds for every  $V \in R$ . To this end, we introduce the concepts of safe and unsafe regions. For any two sensors  $S_i$  and  $S_j$ , we define the *safe region* to be the area in which for any point  $V$ ,  $\alpha(\overrightarrow{VS_i}, \overrightarrow{VS_j}) \leq 2\theta$ ; and define the *unsafe region* to be the area in which for any point  $V$ ,  $\alpha(\overrightarrow{VS_i}, \overrightarrow{VS_j}) > 2\theta$ . The following lemma shows how to find the two regions (Figure 5).

**Lemma 3.2** Given  $S_i$  and  $S_j$ , there are two arcs  $\widehat{S_i S_j}$  and  $\widehat{S_i' S_j'}$  which connect  $S_i$  and  $S_j$  and are symmetrical with respect to the line  $S_i S_j$ , such that the unsafe region is the enclosed region bounded by the arcs and the safe region is the open region outside the unsafe region.

PROOF. We show how to find the two arcs. First we can find two different points  $P_\theta$  and  $P_\theta'$  on the perpendicular bisector of segment  $\widehat{S_i S_j}$ , such that  $\angle S_i P_\theta S_j = \angle S_i P_\theta' S_j = 2\theta$  and they are on different sides of  $S_i S_j$ . Without loss of generality, suppose  $P_\theta$  is on the left side and  $P_\theta'$  is on the right side (Figure 5).

We draw the circumscribed circles of triangle  $\triangle S_i P_\theta S_j$  and  $\triangle S_i P_\theta' S_j$ . Denote the centers of the circles by  $O_{S_i S_j}$  and  $O'_{S_i S_j}$ , and the radius (which is the same for both) by  $r_{safe}$ . Then arc  $\widehat{S_i S_j}$  is the portion of the perimeter of  $\odot O_{S_i S_j}$  on the left side and  $\widehat{S_i' S_j'}$  is the portion of  $\odot O'_{S_i S_j}$  on the right.

In fact, for any circle and a fixed chord (defined here by  $\widehat{S_i S_j}$ ) of the circle, all inscribed angles with two endpoints



**Figure 6: Dotted circles are the boundaries between safe and unsafe regions of two neighboring cameras; the shaded area of  $R$  is not full-view covered since it is within the unsafe regions of  $S_4, S_5$  and  $S_5, S_6$ .**

at the ends of the chord are either equal or supplementary to each other. Specifically, they are equal if the third points of the angles are on the same side of the chord. Furthermore, for a given point  $P_\theta$  on the perimeter of the circle and another point  $P$  on the same side of line  $S_i S_j$  as  $P_\theta$ , if  $P$  is outside the circle ( $\|PO_{S_i S_j}\| > r_{safe}$ ), then  $\angle S_i P S_j < \angle S_i P_\theta S_j$ ; if  $P$  is inside the circle ( $\|PO_{S_i S_j}\| < r_{safe}$ ), then  $\angle S_i P S_j > \angle S_i P_\theta S_j$ . The proof of this property can be found in any textbook on Euclidean Geometry and hence omitted here.  $\square$

Now we can give a necessary and sufficient condition for  $R$  to be full-view covered under some constraint.

**Theorem 3.3** Suppose for every point  $V \in R$ , the circular list  $CL_V = \{S_{V_1}, \dots, S_{V_k}\}$  is the same (in a circular way). Then  $R$  is full-view covered if and only if it is within the polygon bounded by  $\{\overrightarrow{S_{V_i} S_{V_{i+1}}}, 1 \leq i \leq k\}$  and for any  $1 \leq i \leq k$ , the unsafe region of  $S_{V_i}$  and  $S_{V_{i+1}}$  does not intersect with  $R$ , where  $V_{k+1}$  denotes  $V_1$ .

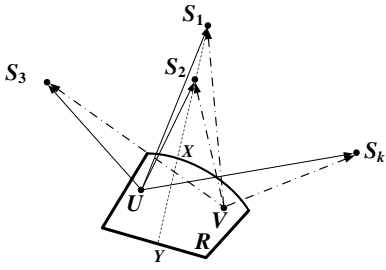
PROOF. This is a corollary from Lemma 3.1 and 3.2.  $\square$

The example in Figure 6 is illustration of our idea. In this example, there are seven cameras covering sub-region  $R$ . We draw the boundaries of the unsafe regions for the seven pairs of neighboring sensors (indicated by dotted arcs) as in Lemma 3.2, and check if they intersect with  $R$ . Note that in computation this can be done by comparing the distance between the circle’s center to each boundary segment of  $R$  against the circle’s radius. As can be seen in the figure, the unsafe regions of  $S_4, S_5$  and  $S_5, S_6$  intersect with  $R$ , and hence the intersection area (shaded area of  $R$ ) is not full-view covered. All other areas in  $R$  are full-view covered.

There is one more issue: the circular list  $CL_V$  may not be the same for every point  $V \in R$ . For example in Figure 7,  $S_1$  is prior to  $S_2$  in  $V$ ’s list, but  $S_2$  is prior to  $S_1$  in  $U$ ’s list. This happens if two cameras covering  $R$  are on a line which intersects with  $R$  (e.g., the line  $\widehat{S_1 S_2}$  intersects with  $R$  at  $X, Y$ ). To solve this problem, we need the following concept.

**Definition 3.4 (Partition)** A partition is a maximal subset of points in any sub-region  $R$  such that the circular list of any point in the subset is the same.

Recall that two circular lists are considered the same if they contain the same set of cameras and the relative orders



**Figure 7: The circular lists for  $U$  and  $V$  are different:**  $CL_V = \{S_1, S_2, S_3, \dots, S_k\}$ ,  $CL_U = \{S_2, S_1, S_3, \dots, S_k\}$ .

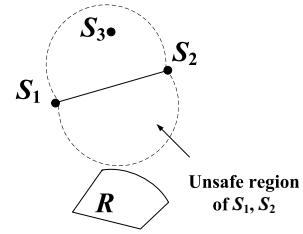
of the cameras are the same. Thus we can use Theorem 3.3 to evaluate the coverage on each partition. To find all the partitions of a sub-region, we divide the sub-region  $R$  by the lines connecting any two cameras covering  $R$ . For example, in the above example,  $R$  can be divided into two partitions by  $\overline{S_1 S_2 X Y}$ . If there were another pairs of cameras like this, then  $R$  would be further divided into more partitions.

### 3.3 Camera Selection for Barrier Coverage

Once the full-view verifications on all sub-regions have been finished, the graph  $G$  can be simplified by removing all nodes that are not full-view covered. Also, all nodes that have only one neighbor can be removed since they can not be intermediate nodes of any  $s - t$  path. In the final graph, every  $s - t$  path (if there is) is actually corresponding to a series of sub-regions that are all full-view covered and connected together, and hence forming a camera barrier across  $A$ . Among these paths (barriers), we want to select the barrier which requires the minimum number of active cameras.

However, the number of  $s - t$  paths can be an exponential function of the number of nodes. Hence it is not efficient to search all the candidate paths to find the one with the minimum cameras used. In fact, even if we were able to find the path with the minimum cameras used, the path is still not guaranteed to be the optimal as some redundancy may exist on the path (see later discussion). Thus as a heuristic, we use a shortest path between  $s$  and  $t$  which can be found by Dijkstra's algorithm [7]. Recall that in Dijkstra's algorithm, we start from the node  $s$  and gradually find the shortest distance to every other nodes. A variable  $dist_i$  is initialized to infinity and gradually updated for each node  $i$  during the execution. It records the up-to-date shortest distance from  $s$  to the node. A subset  $B$  of nodes whose  $dist_i$  have been determined (i.e., not changing in the future) is maintained. Each time a new node  $j$  which is not in the subset and has the smallest  $dist_j$  is selected and added into the subset. Meanwhile, for any of its neighbor  $k$ , if  $dist_k$  is greater than  $dist_j + cost(j, k)$ ,  $dist_k$  is changed to the latter, where  $cost(j, k)$  is the edge cost from  $j$  to  $k$ . In our case, the cost of every edge is 1, although some optimization can be done as we will see later. The algorithm stops if  $t$  is selected and added into the set, which means the shortest path from  $s$  to  $t$  is found.

One optimization can be made to the algorithm. We observe that two nodes are adjacent if the two sub-regions share a common boundary. That means the two camera sets covering these two sub-regions differ by only one element, which further implies that one of the two sets includes the other. Thus, if the sub-region covered by the larger set is chosen to be working, the other sub-region can be covered



**Figure 8:  $S_1, S_2, S_3$  all cover  $R$  and  $S_3$  is redundant because the unsafe region of  $S_1$  and  $S_2$  does not intersect with  $R$ .**

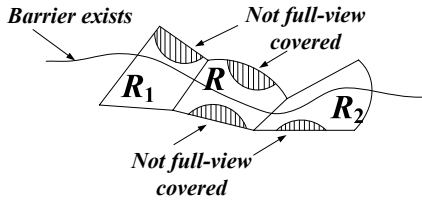
at no additional cost. During the execution of Dijkstra's algorithm, we take advantage of this property by setting the cost of the edge from the node with a larger camera covering set to the node with a smaller subset to be 0, and all other edges to be 1. This encourages the algorithm to select the node which is covered by cameras that are already used.

### 3.4 Redundancy Reduction

The method described above has one drawback: there could be many redundant cameras which can be turned off without affecting the validation of the barrier. In our method, for each sub-region on the barrier, all the cameras covering this sub-region were turned on although some of them may not be necessary. For example, in Figure 8 camera  $S_1, S_2, S_3$  all cover sub-region  $R$  where  $S_3$  can be turned off if  $S_1$  and  $S_2$  are both on. This issue becomes serious if the camera deployment is dense. Therefore we propose the following method to reduce the degree of redundancy. Note that this is not an easy problem since a camera that is redundant for one sub-region may be necessary for another sub-region.

First we identify redundant cameras for each sub-region. A camera  $S_i$  is redundant with respect to a sub-region  $R$  if  $R$  is still full-view covered after  $S_i$  is turned off. Based on Theorem 3.3, there is a simple way to test this. If the unsafe region of the two neighboring cameras of  $S_i$  (i.e., the two adjacent to  $S_i$  in the circular list) does not intersect with  $R$ , then  $S_i$  is redundant. Here we call these two cameras *dependent cameras* of  $S_i$ . Note that for the barrier to be valid, the dependent cameras of  $S_i$  should be working when  $S_i$  is turned off. For example, consider a camera  $S_3$  in Figure 8. If the unsafe region of  $S_1, S_2$  does not intersect  $R$ ,  $S_3$  can be turned off and  $S_1$  and  $S_2$  are the dependent cameras. This judgement can be made during the verification procedure in Section 3.2 and we need to consider each partition separately as the neighbors of the cameras may change from one partition to another. Finally, a camera  $S_i$  is considered a candidate to be turned off if it is redundant with respect to all the sub-regions that are in its FoV and selected to be part of the barrier (the  $s - t$  path). All the dependent cameras of  $S_i$  should be recorded during the above process.

Then we consider how to safely remove the redundant cameras along the entire barrier. Note that we can not simultaneously remove all the candidates since some of them could be dependent cameras for others. We want to turn off as many cameras as possible without affecting the barrier. For this purpose, a graph called the dependent graph  $DG = (V, E)$  is constructed, where a node  $S_i \in V$  if  $S_i$  is a candidate, and an edge  $(S_i, S_j) \in E$  if  $S_i$  is a dependent camera of  $S_j$  or vice versa.



**Figure 9:** Although none of the three sub-regions  $R$ ,  $R_1$  and  $R_2$  is full-view covered, a valid camera barrier exists across the full-view covered portion within the three sub-regions.

**Theorem 3.5** The cameras in an independent vertex set of the dependent graph  $DG$  can be turned off simultaneously without affecting the validation of the barrier. In particular, the maximum set of cameras that can be turned off simultaneously corresponds to a maximum independent vertex set of  $DG$ .

**PROOF.** Recall that an independent vertex set of a graph is a set of vertices in which any two of them have no common edges. Any two cameras from an independent vertex set of the above graph can not be the dependent cameras of each other, which further means their eligibility to be turned off do not assume the existence of the other. Thus the theorem is proved.  $\square$

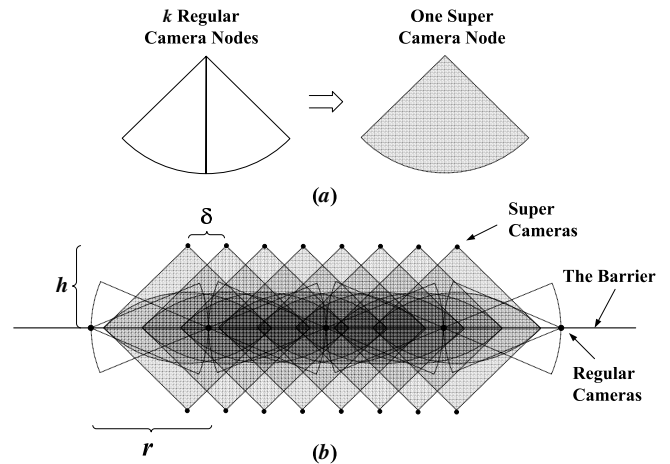
Finding the maximum independent vertex set is proved to be NP-complete. Both centralized and distributed algorithms (heuristics) have been studied extensively. One possible solution can be found in [16].

The simulation results in Section 5 indicate that the algorithm can substantially reduce the number of cameras in use.

### 3.5 Discussion

Our algorithm of full-view coverage verification not only answers the question if a sub-region is full-view covered, but also precisely identifies which part of the sub-region is full-view covered and which part is not. This is very important because in some cases, even if a sub-region is not full-view covered, which only means some portion may not be covered as required, its full-view covered portion can still contribute to the construction of the barrier. An illustration is shown in Figure 9. In this example, none of the three sub-regions are considered full-view covered. However, since the full-view covered portion of the three is connected, it is still possible to construct a barrier across them.

One interesting application of the above method is to identify the “weak spot” of the given deployment. For example, the system planner may want to know if there are some traversing paths from the entry side to the exit side such that the object can follow without being identified by the cameras. To answer this question, we just need to find a connected zone from the entry to the exit such that every point within the zone is NOT full-view covered. If such (breaking) paths exist, a further question would be how to improve the surveillance quality if more cameras can be deployed or the original deployment can be adjusted (e.g., steering the deployed cameras although some delay would be incurred [23]). These are some extensions (dual problems) to the original barrier construction problem. They are very practical since sometimes the barrier may not exist but a metric to measure the quality of the surveillance system in terms of coverage



**Figure 10:** (a) Multiple cameras are bundled together to form one super camera with larger FoV; (b) A deployment pattern for camera barrier coverage: the black dots on the barrier denote regular cameras; the grey area denotes the FoV of the super cameras.

and a method to improve it are still needed. We leave these issues as future work.

## 4. CAMERA BARRIER CONSTRUCTION IN DETERMINISTIC DEPLOYMENT

In this section, we introduce a deterministic deployment pattern for camera barrier coverage. Our goal is to deploy a set of cameras to the monitored field such that each point of a *given barrier (line)* is full-view covered and the number of cameras used is as few as possible. We first describe the deployment pattern and then analyze the number of cameras used under various deployment parameters. Note that although our focus here is on the case where the barrier is a straight line, the technique can be extended to other scenarios where the barrier consists of line segments, or the barrier is a curve, although some approximation may be needed to deal with curves.

### 4.1 Description of the Deployment Pattern

We first place cameras one by one along the barrier with each one facing to the right, i.e., the camera’s orientation vector  $f_i$  is parallel to the barrier and points to the right. The distance between every two adjacent cameras is  $r$ . Similarly, a second set of cameras are placed one by one on the barrier with each one facing to the left (Figure 10(b)).

Then we place another set of cameras along a line above the barrier, with distance  $h$  to it, where  $h$  is a parameter to be defined later. On this line, any two adjacent deployment spots are separated by distance  $\delta$  (to be defined later). At each deployment spot, we place  $k$  (to be defined later) cameras together and merge their FoV to form a “super camera” node with FoV equals to  $k\varphi$  (Figure 10(a)). Note that each super camera can be considered as one camera with larger FoV and all other parameters the same as a regular camera. The orientation vector of each super camera points down to the barrier. Symmetrically, we deploy another set of super cameras along the line with distance  $h$  below the

barrier. Each super camera on this line points up to the barrier (Figure 10(b)).

Now we elaborate how to derive the above three parameters:  $h$ ,  $k$  and  $\delta$ . Given the camera's parameters  $(r, \varphi, \theta)$ , we have some flexibility in choosing one parameter from  $(h, k, \delta)$ , and the choice of the other two depends on the chosen one. We give the relationship among them in the following theorem.

**Theorem 4.1** Given  $0 \leq h \leq r$ , in order to guarantee that every point of the barrier is full-view covered, the minimum value for  $k$  is  $k \geq 2 \arccos(\frac{h}{r})$  and the maximum value for  $\delta$  is  $\delta \leq \min\{\delta_1, \delta_2\}$ , where  $\delta_1 = \sqrt{r^2 - h^2} - \frac{h}{\tan 2\theta}$  and  $\delta_2 = 2h \tan \theta$ ; and furthermore,  $h$  should be smaller than  $h_0 = \frac{r}{\sqrt{1 + 1/(\tan 2\theta)^2}}$ .

**PROOF.** Consider an arbitrary point  $P$  on the barrier with facing direction  $\vec{d}$ . We prove that if  $k$  and  $\delta$  are as above, there is always a sensor  $S_i$  such that  $P$  is covered by  $S_i$  and  $\alpha(\vec{d}, \overrightarrow{PS_i}) \leq \theta$ .

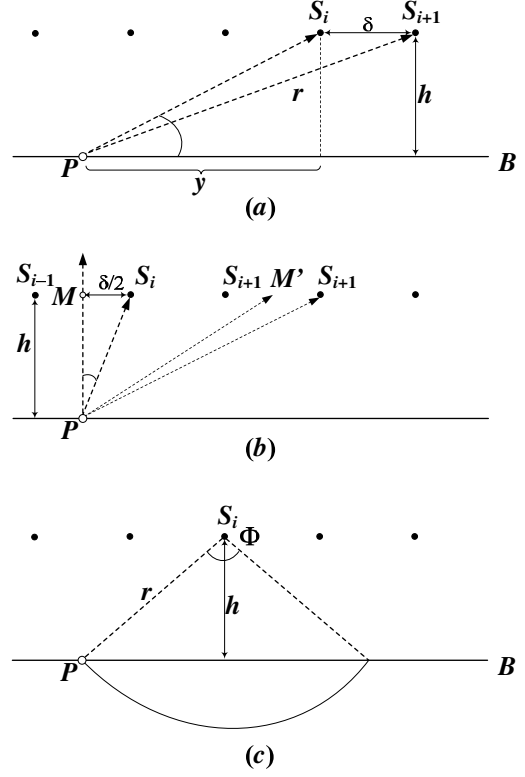
We use the angle between  $\vec{d}$  and the barrier, denoted as  $\alpha_d$ , to represent  $\vec{d}$  (i.e., let  $\vec{d}$  rotate until it is parallel to the barrier and pointing to the right; then the angle range crossed is used to indicate  $\vec{d}$ ). Since the deployment is symmetrical with respect to the barrier, we only consider the case when  $0 \leq \alpha_d < \pi$ . First, notice that we have placed two group of cameras on the barriers, with one group pointing to the right and other pointing to the left. Thus if  $0 \leq \alpha_d \leq \theta$  or  $\pi - \theta \leq \alpha_d < \pi$ , there is always a sensor from the two group such that the above condition is satisfied for  $\vec{d}$ .

When  $\theta \leq \alpha_d \leq \pi - \theta$ , there are two extreme cases to consider, which we show in Figure 11. The first case is that there is a super camera  $S_{i+1}$  such that  $\|PS_{i+1}\| = r + \epsilon$ , where  $\epsilon$  is a positive number that can be arbitrarily small. Hence  $P$  is just outside the coverage range of  $S_{i+1}$ . (Figure 11(a)). In this case, if  $\vec{d}$  is along the direction  $\overrightarrow{PS_{i+1}}$ , either there is a super camera  $S_i$  with distance less than  $r$  (as shown in the figure,  $S_i$  is to the left of  $S_{i+1}$  such that  $\alpha(\vec{d}, \overrightarrow{PS_i}) \leq \theta$ ), or the angle between  $\vec{d}$  (i.e.,  $\overrightarrow{PS_{i+1}}$ ) and the barrier is no more than  $\theta$ . The second case is that the projection point of  $P$  on the deploying line of the super cameras is at the mid-point (denoted as  $M$ ) of two adjacent super cameras  $S_{i-1}$  and  $S_i$ , and  $\vec{d}$  is along the direction  $\overrightarrow{PM}$  (Figure 11(b)). If the above condition holds for  $\vec{d}$  in these two cases, it also holds for other  $\vec{d}$ .

In the first case, let  $B$  be a point at the right end of the barrier. We only need to show that  $\angle S_i P B \leq 2\theta$ . Notice that  $\angle S_i P B \leq \arctan \frac{h}{y}$ , where  $y = \sqrt{r^2 - d^2} - \delta$ . Since  $\delta \leq \delta_1 = \sqrt{r^2 - h^2} - \frac{h}{\tan 2\theta}$ , as indicated in the assumption,  $y \geq \frac{h}{\tan 2\theta}$ . Hence  $\angle S_i P B \leq 2\theta$ .

In the second case, we only need to show that  $\angle MPS_i \leq \theta$ . Notice that  $\angle MPS_i = \arctan \frac{\delta}{2h}$  and  $\delta \leq \delta_2 = 2h \tan \theta$ , as indicated in the assumption. Hence  $\angle MPS_i \leq \theta$ .

For the minimum value of  $k$ , the above argument holds only if for each super camera, a point  $P$  on the barrier with distance  $r - \epsilon$  (for any  $\epsilon > 0$ ) should be in the range of the super camera (Figure 11(c)). Based on this and the assumption that the super camera is with distance  $h$  to the barrier, we can obtain the minimum FoV (denoted as  $\Phi$ ) needed for the super camera as  $\Phi = 2 \arccos \frac{h}{r}$ . Hence the number of cameras needed for each super camera node is  $k = \lceil \frac{\Phi}{\varphi} \rceil = \lceil \frac{2 \arccos(h/r)}{\varphi} \rceil$ .



**Figure 11: Analysis on the relationship among the deployment parameters.**

Finally, the upper bound  $h_0$  is given by the constraint  $\delta_1 \geq 0$ . Therefore, the theorem is proved.  $\square$

## 4.2 Analysis on Number of Cameras

Given the above dependency relationship of  $k$ ,  $\delta$  on  $h$ , we optimize the choice of  $h$  such that the total number of cameras used is minimized.

There are two groups of cameras: the first group consists of the regular cameras deployed on the barrier and the second group consists of two sets of super cameras deployed on the two lines with distance  $h$  to the barrier. Without loss of generality, let us consider a unit length of the barrier and assume all the other parameters are unified. The number of cameras used in the first group is  $2 \cdot \frac{1}{r}$ , where  $r$  is normalized to unit length. This is independent of the choice of  $h$ . The number of cameras needed for the second group is  $N(h) = 2 \cdot k \frac{1}{\delta}$ , which depends on the choice of  $h$  as indicated in Theorem 4.1. In order to minimize the total number of cameras, we need to minimize  $N(h)$ .

**Theorem 4.2** Given  $0 \leq h \leq h_0$ , where  $h_0$  is given in Theorem 4.1, the density of cameras needed (i.e., number per unit of length) in the above deployment is

$$\rho(r, \theta, \varphi) = \frac{2}{r} + \frac{\lceil \frac{2 \arccos(h_1/r)}{\varphi} \rceil}{h_1 \tan \theta}$$

$$\text{where } h_1 = \frac{r}{\sqrt{1 + (1/\tan 2\theta + 2 \tan \theta)^2}}$$

**PROOF.** The first part of the expression is the number of cameras deployed on the barrier, which does not depend on

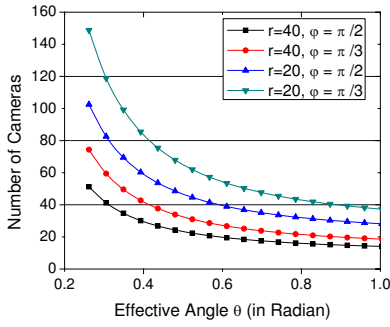


Figure 12: Cameras needed for constructing a 100m barrier.

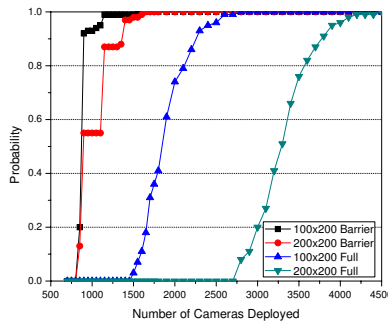


Figure 13: Coverage probability vs. Number of deployed cameras.

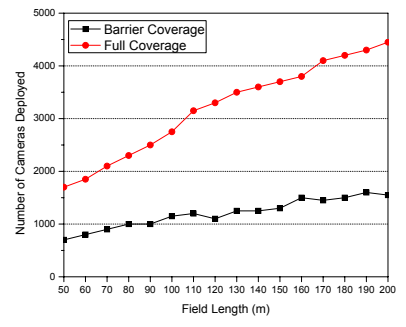


Figure 14: Number of deployed cameras vs. Field length.

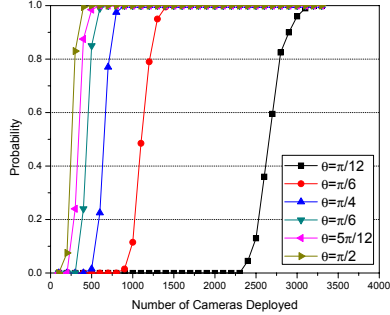


Figure 15: Coverage probability vs. Number of deployed cameras.

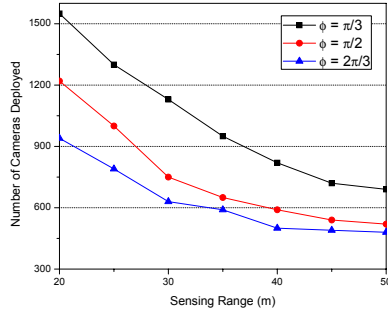


Figure 16: Number of cameras vs. Sensing range.

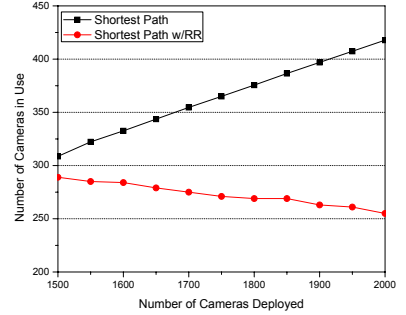


Figure 17: Cameras in use vs. Cameras deployed.

the choice of  $h$ . To see the second part, we minimize  $N(h)$  with respect to  $0 \leq h \leq h_0$ . From Theorem 4.1, we have

$$N(h) = \frac{\arccos(h/r)}{\min\{\delta_1(h), \delta_2(h)\}},$$

where  $\delta_1(h) = \sqrt{r^2 - h^2} - \frac{h}{\tan(2\theta)}$  and  $\delta_2(h) = 2h \tan(2\theta)$ . As  $h$  increases,  $\delta_1(h)$  decreases and  $\delta_2(h)$  increases. When  $h = h_1$  as indicated in the theorem,  $\delta_1(h) = \delta_2(h)$ .

When  $h \in (0, h_1)$ ,  $N(h) = \frac{\arccos(h/r)}{\delta_2(h)}$ . Since  $\arccos(h/r)$  is a decreasing function of  $h$ , the minimum value of  $N(h)$  is obtained when  $h \rightarrow h_1$ . When  $h \in (h_1, h_0)$ ,  $N(h) = \frac{\arccos(h/r)}{\delta_1(h)}$ . By calculating the derivative of  $N(h)$ , we know that  $N(h)$  is an increasing function of  $h$  within this interval. Thus the minimum value is obtained when  $h \rightarrow h_1$ .  $\square$

Figure 12 is an illustration of the above result. It shows how many cameras are needed to construct a camera barrier with length of 100m when  $\theta$  is from  $\frac{\pi}{12}$  to  $\frac{\pi}{3}$  and  $r = 20m, 40m$  and  $\varphi = \frac{\pi}{3}, \frac{\pi}{2}$  respectively.

## 5. EVALUATIONS

In this section, we present the evaluation results. We first compare the number of cameras needed for barrier coverage and full coverage, i.e., every point of the monitored area is full-view covered. Then we show how our algorithm reduces the number of cameras used for the barrier coverage.

### 5.1 Comparison with Full coverage

One major advantage of barrier coverage over full coverage is the cost-effectiveness. In camera sensor networks, full coverage means that every point of the monitored field

is full-view covered. Since full-view coverage essentially requires each point to be covered by multiple camera sensors (at least  $\lceil \pi/\theta \rceil$ ), the full coverage in camera sensor networks is imaginably more demanding than in traditional scalar sensor networks, and hence the saving by barrier coverage could be more significant.

We have two scenarios here. In the first scenario, the monitored field is 200m in width (along  $x$ -axis), 100m and 200m in length ( $y$ -axis) separately. The camera's parameters are  $r = 30m, \theta = \pi/3, \varphi = 2\pi/3$ . Cameras are deployed randomly and uniformly in the deployed field. To avoid the boundary effect<sup>2</sup>, the deployed field is a larger area with both the length and the width  $2r$  longer than the monitored field. Figure 13 shows how the coverage probability varies as the number of deployed sensor increases. To estimate the probability, we run each experiment 500 times and the probability is obtained by dividing the number of times when the desired coverage is achieved by 500. As the result in Figure 13 shows, the probability of the existence of a camera barrier (denoted as "barrier") is almost 1 when the number of cameras deployed is beyond 1000 if the field length is 100. On the other hand, at least 2500 cameras are needed for full coverage (denoted as "full"). The difference is even bigger if the field length is 200, where barrier coverage demands no more than 1500 cameras but full coverage demands more than 4000 cameras.

In the second scenario, the camera's parameters and the width of the monitored field are fixed as in the above. We

<sup>2</sup>If the deployment field is the same as the monitored field and the deployment is random and uniform, then the point close to the boundary is less likely to be covered than the point in the center area.



change the length of the field from  $50m$  to  $200m$  and observe how many cameras are needed to achieve the desired coverage (barrier and full) with at least 0.99 probability. Note that in a random deployment given the same number of deployed cameras, as the field length increases the camera density will drop. As a result, to achieve the same high probability of coverage (in both full and barrier coverage), more cameras should be deployed. As shown in Figure 14, the number of cameras required for barrier coverage is much less than that in full coverage. As the field length increases, the number of cameras required for full coverage increases much faster than that for barrier coverage. This result is consistent with our expectation: given the field width unchanged, in full coverage the area to be full-view covered increases linearly as the field length increases, and so does the number of cameras needed; however since the barrier is across the width of the field, which is unchanged during the test, the number of cameras needed does not increase that fast, and the advantage of cost-effectiveness of barrier coverage is more obvious.

## 5.2 Impact of Camera Parameters

We study the impact of the three camera parameters: the sensing range (radius)  $r$ , the FoV  $\varphi$  and the effective angle  $\theta$  on the probability of camera barrier coverage. In Figure 15, it is shown that with different effective angle ( $\theta$ ) how the probability of camera barrier coverage varies as the number of deployed cameras increases. The monitored field is  $100m \times 200m$ . The sensing range  $r = 30m$  and the FoV  $\varphi = 2\pi/3$ . The probability is estimated in the same way as in the above experiment. Intuitively, smaller  $\theta$  implies more cameras required for an object to be full-view covered, and hence more cameras needed for a camera barrier. This is exactly what the figure shows. As  $\theta = \pi/12$ , about 3000 cameras are needed for the existence of a camera barrier with probability approaching 1. On the other hand, no more than 500 cameras are needed if  $\theta = \pi/2$ . Note that  $\theta = \pi/12$  means a qualified image must be taken when the object's facing direction is no more than 15 degree away from the frontal view, which is a very high quality requirement in practice. Even reducing the level a little to  $\theta = \pi/6$  (30 degree) can dramatically reduce the required scale of deployment.

Figure 16 shows the number of cameras needed ( $y$ -axis) in deployment to achieve a 0.99 probability of a camera barrier. As in the above experiment, the field size is  $100m \times 200m$ . Here the effective angle  $\theta$  is fixed to be  $\pi/3$ . There are three curves, for  $\varphi = \pi/3$ ,  $\pi/2$  and  $2\pi/3$  separately. Each curve shows how the required number of deployed cameras decreases as the sensing range increases given a fixed FoV. By comparing the three curves, it can be seen that as the FoV becomes wider, the number of cameras needed is reduced.

## 5.3 Number of Cameras on Barrier

The effectiveness of redundancy reduction is shown in Figure 17. In this test, we vary the total number of deployed cameras and observe the number of cameras in use. The camera's parameters are  $r = 30m$ ,  $\theta = \pi/3$ ,  $\varphi = 2\pi/3$ , and the monitored field is  $100m \times 200m$ . As can be seen, if no redundancy reduction is used (indicated as "Shortest Path"), the number of cameras in use increases (almost linearly) as the total number of deployed cameras increases. This is because in the original algorithm, all cameras covering

a sub-region is selected if this sub-region is chosen to be on the barrier. As the redundancy reduction procedure is carried out (indicated as "Shortest Path w/RR"), the redundant cameras are turned off as many as possible. Moreover, as more cameras are deployed, there is more flexibility on camera selection, and the number of cameras in use drops.

## 6. RELATED WORK

Barrier coverage has been first studied in [9]. In wireless sensor networks, one related problem is the maximum breach and minimum exposure path problem [15, 12]. In this problem, the coverage quality of a sensor (or exposure) is modeled as a decreasing function of the distance between the sensor and the object. The goal is to find a traversing path in a deployed sensor network such that the maximum exposure is minimized. After the introduction of the problem, some distributed algorithms have been proposed, in which sensor collaboration is exploited to detect the intruder [6, 21].

The concepts of weak and strong barrier coverage in wireless sensor networks are introduced in [11]. A wireless sensor network provides weak barrier coverage if the intruder is guaranteed to be detected when it takes the shortest path (i.e., an orthogonal line) to cross the field. Strong barrier coverage guarantees the detection of the intruder no matter what kind of path it takes. They obtain the critical condition of weak barrier coverage in a random deployment. The critical condition for strong barrier coverage is obtained in [13] by using percolation theory. They also give a distributed algorithm to construct the sensor barrier. An effective way of measuring the quality of barrier coverage is proposed in [5]. The idea is that if the intruder is guaranteed to be detected when its path is confined in a sliced area with a given width (bounded), then the bound of this width can be used to measure the quality of the barrier. Under this model, the strong barrier coverage and the weak barrier coverage are two extreme cases. They also provide an efficient way to find the weak point of the barrier based on the measurement results. The concept of barrier information coverage is introduced in [25]. The basic idea is to exploit the collaboration between sensors on target detection to reduce the number of sensors in use and hence prolong the network lifetime. Finally, the problem of constructing sensor barrier with mobile sensor is studied in [19]. An optimization algorithm is given to schedule the movement of the mobile sensors for barrier coverage under the constraint that the moving distance of each mobile sensor is limited.

The full-view coverage model is first introduced by us in [24]. A full-view coverage verification method is proposed and an estimate of deployment density to achieve full-view coverage for the whole monitored area is given. We adopt a similar technique for full-view coverage verification in the second step of barrier construction for arbitrary deployment. The difference is as follows. In [24], coverage verification only needs to be applied to the boundary of the sub-regions as it is a "yes" or "no" problem and the whole area is full-view covered if and only if each boundary segment is full-view covered. However, in this paper, since the barrier coverage is a construction problem which is much complex, we need a finer analysis on each sub-region to show exactly which part of each sub-region is full-view covered and which part is not. As mentioned in the end of Section 3, this can help precisely identify and construct the barrier.

## 7. CONCLUSION

Barrier coverage is attractive for many practical applications of wireless sensor networks. There is no exception for camera sensor networks, which combine the technology advancement in both computer vision and wireless sensor networks, and are believed to be the key component for many appealing applications. However, because of the unique feature of camera sensors, the barrier coverage problem is more challenging and hence deserves more studies.

A good camera barrier should be able to detect any desired aspect (face) of the intruder traversing the monitored field. In this paper, we exploited the concept of full-view coverage and proposed a systematic way to build up a camera barrier in both random and deterministic deployment. A camera barrier with the full-view coverage property guarantees that no matter where the intruder faces, it will always be detected by an active camera whose viewing direction is close enough to the intruder's facing direction and hence the face image can be identified effectively. We designed a novel method to construct a camera barrier in any given sensor deployment. As some redundant cameras may exist on the barrier which can increase the cost, we also presented a method to reduce the degree of redundancy without affecting the validation of the barrier. A deterministic deployment pattern for camera barrier construction was also proposed. The number of cameras used has been analyzed and optimized under various camera parameters for this specific deployment pattern.

## Acknowledgment

We would like to thank Sudipta Sengupta and the anonymous reviewers for their insightful comments and helpful suggestions. This work was supported in part by the US National Science Foundation (NSF) under grant number CNS-0916171.

## 8. REFERENCES

- [1] Distributed smart cameras. *Proceedings of the IEEE, Special Issue*, 96(10), 2008.
- [2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury. A survey on wireless multimedia sensor networks. *Comput. Netw.*, 51(4):921–960, 2007.
- [3] P. S. Alexandroff. *Combinatorial Topology*. New York: Dover, 1998.
- [4] V. Blanz, P. Grother, P. J. Phillips, and T. Vetter. Face recognition based on frontal views generated from non-frontal images. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 454–461, 2005.
- [5] A. Chen, T. H. Lai, and D. Xuan. Measuring and guaranteeing quality of barrier-coverage in wireless sensor networks. In *ACM MobiHoc 2008*.
- [6] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja. Sensor deployment strategy for detection of targets traversing a region. *Mob. Netw. Appl.*, 8:453–461, August 2003.
- [7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, 2nd Ed.* The MIT Press, 2001.
- [8] A. O. Ercan, A. El Gamal, and L. J. Guibas. Object tracking in the presence of occlusions via a camera network. In *ACM IPSN 2007*.
- [9] D. Gage. Command control for many-robot systems. In *Proc. Nineteenth Annual AUVS Technical Symposium (AUVS-92)*, pages 22–24, 1992.
- [10] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu. SensEye: a multi-tier camera sensor network. In *ACM MULTIMEDIA 2005*, pages 229–238.
- [11] S. Kumar, T. H. Lai, and A. Arora. Barrier coverage with wireless sensors. In *ACM MobiCom 2005*, pages 284–298.
- [12] X.-Y. Li, P.-J. Wan, and O. Frieder. Coverage in wireless ad hoc sensor networks. *IEEE Transactions on Computers*, 52:753–763, 2003.
- [13] B. Liu, O. Dousse, J. Wang, and A. Saipulla. Strong barrier coverage of wireless sensor networks. In *ACM MobiHoc 2008*, pages 411–420.
- [14] C. Liu and G. Cao. Spatial-temporal coverage optimization in wireless sensor networks. *IEEE Transaction on Mobile Computing*, Apr. 2011.
- [15] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *IEEE INFOCOM 2001*, pages 1380 – 1387.
- [16] V. T. Paschos. A  $\delta/2$ -approximation algorithm for the maximum independent set problem. *Information Processing Letter*, 44(1):11–13, 1992.
- [17] P. J. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott, and M. Sharpe. FRVT 2006 and ICE 2006 large-scale results. *National Institute of Standards and Technology, Tech. Rep. NISTIR 7408*, 2007.
- [18] M. Rahimi, S. Ahmadian, D. Zats, R. Baer, D. Estrin, and M. Srivastava. Cyclops: In situ image sensing and interpretation in wireless sensor network. In *ACM SenSys 2005*.
- [19] A. Saipulla, B. Liu, G. Xing, X. Fu, and J. Wang. Barrier coverage with sensors of limited mobility. In *ACM MobiHoc 2010*.
- [20] C. Sanderson, T. Shang, and B. C. Lovell. Towards pose-invariant 2d face classification for surveillance. In *AMFG'07: Proceedings of the 3rd international conference on Analysis and modeling of faces and gestures*, pages 276–289. Springer-Verlag, 2007.
- [21] G. Veltri, Q. Huang, G. Qu, and M. Potkonjak. Minimal and maximal exposure path algorithms for wireless embedded sensor networks. In *ACM SenSys 2003*, pages 40–50.
- [22] P. Wang, R. Dai, and I. F. Akyildiz. Collaborative data compression using clustered source coding for wireless multimedia sensor networks. In *IEEE INFOCOM 2010*.
- [23] Y. Wang and G. Cao. Minimizing service delay in directional sensor networks. In *IEEE INFOCOM 2011*.
- [24] Y. Wang and G. Cao. On full-view coverage in camera sensor networks. In *IEEE INFOCOM 2011*.
- [25] G. Yang and D. Qiao. Barrier information coverage with wireless sensors. In *IEEE INFOCOM 2009*.