Connected Wireless Camera Network Deployment with Visibility Coverage

Hua Huang^{*} Chien-Chun Ni[†] Xiaomeng Ban[†] Jie Gao[†] Andrew T. Schneider^{*} Shan Lin^{*}

* Department of Computer and Information Science, Temple University. {hua.huang, tue82238, shan.lin}@temple.edu
 [†] Department of Computer Science, Stony Brook University. {chni, jgao, xban}@cs.stonybrook.edu

Abstract—We consider the problem of deployment of cameras inside a complex indoor setting for surveillance applications. We formulate the problem of minimum guarding network that places a minimum number of cameras satisfying both visual coverage of the domain and network connectivity. We prove that finding the minimum guarding network in both the geometric setting and discrete setting is NP-hard. We also give a 2approximation algorithm to the geometric minimum guarding network. Motivated by the connection of this problem with watchman tour problem and art gallery problem, we developed two algorithms that generate satisfactory results on a prototyped testbed.

Keywords—Visibility Coverage, Wireless Connectivity, Camera Network

I. INTRODUCTION

With the recent advancement of pervasive computing, wireless networks, and optical sensing, wireless camera networks are deployed for a wide range of applications. In many existing systems for infrastructure monitoring and home health-care, camera network deployments heavily rely on existing infrastructure. Typically, a node is plugged into the wall and directly connects to a local network access point through wire or wireless. However, such infrastructure support is not available in many scenarios like first responder and military applications, where the network needs to be deployed quickly with little infrastructure support. For example, soldiers deploy a camera network to monitor suspicious activities in a building during military operations. These applications impose a unique set of requirements on camera network deployment: 1) full visibility coverage. The whole building needs to be monitored completely; 2) reliable wireless connectivity. All the camera nodes need to be self-organized to form a camera network, to transfer pictures or videos to a base station over wireless; 3) low deployment cost. Minimal number of devices is desirable for the low deployment effort and short deployment time. These constraints, together with the complex building structure floor plan, make the camera network deployment very challenging.

To address these challenges, it is essential to employ optical sensing and wireless communication models together. Therefore, we define the *Minimum Connected Guarding Network Problem*, which combines the visibility sensing model and wireless communication model into an optimization framework. In the visibility model, each camera's sensing range is only restricted by line of sight. This is a generalization of the "cone" model [1] of a camera. While in the wireless communication model, wireless relay nodes are introduced to connect adjacent camera nodes when the distances of their wireless links are too large. Based on these two models, the Minimum Connected Guarding Network Problem looks for the minimum number or camera and relay nodes that can guard a floor, while ensuring the wireless connectivity.

The integration of isotropic sensing and wireless networking makes this problem unique and interesting. Previous researchers have proposed many camera deployment algorithms [2][3][4] to maximize the visual coverage of the camera network, but little attention is given to provide a connected network while minimizing the wireless communication cost. Whereas related sensor network coverage research [5][6] provides valuable insights to deployment problems, but they usually assume short range circular sensing models, which do not apply for optical camera. To fill this missing gap, our goal is to achieve full visibility coverage with the minimal number of necessary cameras that also optimize the overall communication cost.

The solution of the minimum connected guarding network problem clearly depends on the scale of the communication link connecting these nodes. Take one extreme, say the communication range of the sensor nodes is large enough such that any two nodes inside the building can directly communicate with each other. The minimum connected guarding network problem boils down to the classical Art Gallery Problem (AGP), which aims at finding a minimum number of point guards such that any point in the building is within direct line of sight of at least one guard. The art gallery problem is a well known NP-hard problem and it has been extensively studied for approximation solutions [7]. Take the other extreme, say that the communication range is very small compared to the scale of building, we basically need to place the sensors along a path to keep them connected. Thus the problem converges to one of finding a connected geometric network such that any point is visible to at least one point of the network. It is not hard to show that such a network of minimum length must be a tree. The problem of finding a minimum guarding tree for a polygon has not been given much attention in the past. A problem similar to ours is the watchman route problem [8], i.e., finding a route of minimum length that guards an entire polygon. It is known that the watchman route problem is NPhard for the general polygon with holes [9]. But nothing is known about the problem if nodes are not connected as a path but a tree.

In this paper, we initiate the study of the deployment of cameras that satisfy both visual coverage and wireless connectivity. We first show that the minimum guarding network is NP-hard to find, when the wireless communication range is a constant r. We also show that one extreme of this setting, when the communication range $r \rightarrow 0$, is still NP-hard to solve. This is termed the geometric minimum guarding network. We then give a 2-approximation algorithm to the geometric minimum guarding network, by using the watchman tour in a simple polygon.

In terms of algorithm development for practical implementations, we consider two possible deployment settings. In the first setting, termed the Connected Visibility region Planning (CVP), we assume that the building floor plan is known and the network designer has plenty of time to deploy and optimize the network. It first identifies a set of camera locations that guarantee the visibility based coverage of the domain, and then find the locations for the relay nodes to connect camera nodes into a minimal spanning tree. This algorithm is computational intensive but produce near optimal deployment result in practice. In the second setting, termed the Connected Visibility region Tracking (CVT), we assume first responder applications in which in-field dynamic deployment must be done quickly. We employ the watchman tour based solution, allowing the first responders to drop sensors along the (shortest) watchman route.

The proposed deployment algorithms are evaluated in real scenarios. We built a wireless camera network testbed to validate the effectiveness and accuracy of our algorithms. Each node of this testbed consists of four off-the-shelf components: a beagle board computer, a WiFi radio, a webcam, and a battery cape. We deployed these camera nodes according to locations generated from CVP and CVT in our campus buildings. Real deployments over time can achieve above 99% wireless connectivity and satisfactory visibility coverage (Currently our camera node doesn't have panoramic cameras). Compared with existing solutions, our deployments have significantly better wireless communication connectivity. Moreover, the simulation results suggest incorporating variable communication range in our deployment algorithm will yield more efficient results.

The rest of the paper is organized as follows: The problem is formally defined in section II. The CVP and CVT algorithms are described in section III. In section V, a realistic camera network implementation is described. The evaluation results are reported in section VI and the related works are reviewed in section VII.

II. MODEL AND HARDNESS

In this section, we will formulate the wireless camera deployment problem rigorously and prove the hardness of the problem.

We consider an indoor scenario and the domain of deployment is modeled by a polygon P. A camera node is a wireless node with visual sensing range defined by line of sight, and wireless communication range defined as a disk of radius r. We would like to place a minimum number of cameras inside P such that two conditions are met:

- All nodes collectively 'guard' the entire polygon P in the sense that any point of P has a direct line of sight path to at least one camera;
- The nodes form a connected network using wireless communication.



Fig. 1: Given an instance of the minimum geometric rectilinear Steiner tree (left), we turn it to an instance in which each point is replaced by a T-junction obstacle (shown to the right).

This problem is denoted as the *Minimum Connected Guarding Network* problem.

If the input polygon is convex, then the camera deployment problem is trivial — placing one camera in any location inside the polygon can ensure full coverage (and it is trivially a connected network). Therefore, we will focus on the setting when P is non-convex.

Theorem 1. Finding the minimum connected guarding network is NP-hard, even in a simple polygon P.

Proof: We use an induction from the standard art gallery problem. Given an arbitrary art gallery instance with an input polygon P, we scale the polygon P down such that it is within a unit disk. The optimal solution for the art gallery problem does not change. But the cameras in any guarding solution form a complete graph. Thus if we have a solution for the minimum connected guarding network, it is the optimal solution for the art gallery problem, which is known to be NP-hard to find.

Notice that the proof above depends on the communication range being a fixed constant. When the size of the deployment domain is much greater than the communication range, i.e., $r \rightarrow 0$, the minimum connected network becomes a geometric graph that 'guards' the polygon P. We would like to find such a geometric network with minimum total length. We call this problem the *minimum geometric guarding network problem*. This problem is also hard for a general polygon P.

Theorem 2. Finding the minimum geometric guarding network in a general polygon with holes is NP-hard.

Proof: We use reduction from the minimum geometric rectilinear Steiner tree problem in the plane. Given n points on a unit lattice called sites, we would like to find a tree T connecting the n sites with minimum total length. The tree may use other non-site lattice points as vertices and all edges of the tree must be either horizontal or vertical. See the left figure in Figure 1. Given such an instance, we construct an instance for the guarding problem. We first enlarge lattice edges to narrow corridors. Each lattice grid becomes a 'hole' of the polygon. In particular, a site vertex will map to a small 'T-junction' gadget hole such that one must visit the junction point in order to guard it. The T-junction hole is small enough to fit inside the corridor. See Figure 1 for the sizes of the corridor and the T-junction hole.

Now we can verify that for a positive integer m, there exists a rectilinear Steiner tree of length at most m if and only if there exists a minimum geometric guarding network of length at most $m + 3\epsilon$. Take a very small ϵ say 0.1. This shows that the minimum geometric guarding network problem is NP-hard for a polygon with holes.

Given the hardness results, we then move on to find approximation algorithms and practical solutions with good performance in reality.

III. MINIMUM GEOMETRIC GUARDING NETWORK

In this section we first show some useful properties of the minimum geometric guarding network. Then we present a 2-approximate solution for the minimum geometric guarding network problem in a simple polygon. This algorithm is also a building block for the general case when $r \neq 0$, to be discussed afterwards.

We represent the input polygon P by a sequence of vertices v_1, v_2, \dots, v_n , with $n \ge 4$. For $i = 1, 2, \dots, n-1$, $e_i = (v_i, v_{i+1})$ represents an edge of the polygon connecting node v_i and v_{i+1} . For ease of presentation, we also impose a direction upon each edge such that the interior of the polygon lies to the left of the edge, or equivalently, the boundary of P is directed counterclockwise. We also assume that without loss of generality that the vertices of P are in general position, i.e., no three vertices are collinear.

A vertex v is a *reflex vertex* if the interior angle at v is greater than π ; a vertex is called *convex* otherwise. A chain of vertices between v_i and v_j is defined as all the vertices that will be encountered if one scans from v_i counterclockwise to v_j . The *visibility polygon* of a point x inside P, denoted by V(x), is defined as the set of points in P with direct line of sight from x.

We say a set of points M inside P are guards or a guard cover, if for any point $p \in P$, there is a point $q \in M$ such that q sees p. We also say that a guard cover is able to guard P.

Theorem 3. Given a polygon P, the minimum geometric network is a tree of polygonal curves.

Proof: Take any geometric guarding network G within P, we can find a finite size 'guard cover' M on G. In particular, we take each reflex vertex v_i and extend its two adjacent edges, $v_{i-1}v_i$ and v_iv_{i+1} . This gives us at most n lines forming an arrangement cutting the polygon P into convex pieces. We take all vertices of this arrangement within P. For each such vertex, we take a point on G visible to it and add it to the guard cover M. Clearly the number of guards is at most $O(n^2)$. Further, the points of M is a guard cover.

Now take a minimum Steiner tree T upon the guards M. Clearly T guards P. Also T is no longer than the total length of G. This shows that the minimum geometric guarding network must be a tree made of polygonal curves.

Therefore from the above theorem we can also denote the optimal solution as the minimum geometric guarding tree.

To get a 2-approximation to the minimum geometric guard network, the idea is to make use of a watchman tour for a given polygon P. A watchman tour is a (closed) cycle inside

P that guards *P*. That is, any point of *P* has direct line of sight to at least one point on the tour [10]. Although finding the shortest watchman tour in a general polygon with holes is NP-hard [10], there is an $O(\log n)$ -approximation algorithm for a rectilinear version with restricted visibility [11]. The watchman tour problem for a simple polygon is solvable in polynomial time (for a tour with a fixed starting point see [12], [13], and for the floating tour without a given starting point see [14]).

We show in the following theorem that the optimal watchman tour is a 2-approximation to the minimum geometric guarding network.

Theorem 4. Inside a polygon P, the optimal watchman tour is a 2-approximation to the minimum geometric guarding network. This is true for both cases when a fixed starting point is given, or not given.

Proof: First any watchman tour is clearly a geometric guarding tree. We take the minimum geometric guarding tree T, double all edges in the tree which then form a tour along the tree, visiting each edge exactly twice, once in each direction. This resulting tour is a watchman tour. It has length exactly twice the length of the minimum geometric guarding tree, which is no shorter than the length of the optimal watchman tour. This proves the theorem.

IV. Algorithms for Minimum Guarding Network

In this section we describe algorithms for finding guarding network, when the communication range of camera nodes is a fixed constant r > 0. Our aim is to provide practically interesting algorithms for real system implementation, to be explained in the next section. We use two approaches. The first one is to find minimum watchman tour and place cameras along the tour, keep the same visibility coverage. The second one is to find art gallery solutions to ensure visibility coverage and then add extra relay nodes along a spanning tree to connect the guards. We describe the two algorithms respectively and we compare their performance on realistic floor plans.

A. Connected Visibility region Tracking

In the following we describe the Connected Visibility region Tracking (CVT) algorithm. The first step of CVT is to compute the optimal watchman tour inside a simple polygon. We adapt the algorithm by Chin *et al.* [9] and Tan *et al.* [13] which in runtime $O(n^4)$ finds the shortest watchman route for simple polygon through a given point *s* within the polygon. The concept is to find the 'cuts' in *P* that the watchman route must touch to guard the whole polygon, and visit these cuts using a shortest tour.

For a given polygon P, suppose v_i is a reflex vertex in Pand one of its adjacent vertices is v'. We shoot a ray from v'to v, hitting the polygon at y, then the visibility cut $C = \overline{vy}$ is a cut of P and separated P into two parts. We call the part of P not containing v' the essential piece of P, denoted as P(C). Suppose the watchman route has not visited the part of P(C) yet, then it must at least touch the visibility cut C in the later route to guard P(C). A visibility cut C_j is dominated by another cut C_i if $P(C_j)$ contains $P(C_i)$, which means if the route passes C_i to touch $P(C_i)$, then $P(C_j)$ is covered. We call a cut essential cut if it is not dominated by any other cuts.



Fig. 2: Cut C dominates C' since P(C') contains P(C) – any tour that visits P(C) must also visit C' and guard P(C'). C is also an essential cut.

With the essential cut, the origin watchman route problem is reduced to find the shortest route that touch every essential cut inside a polygon. In brief, we triangulate the given polygon and "reflex" the polygon using the essential cut as mirrors in order. After these reflections we pick the shortest path connecting the starting point s and its image, which becomes a tour in the original polygon P. We elaborate the details below.

We first list the essential cuts in clockwise order, $\{C_1, C_2, \dots, C_k\}$. Starting from point *s*, we want to find a path to visit this cut list. We enumerate on the first cut to visit starting from *s* and take the shortest tour. If the first cut to visit is C_i , then the tour visits $\pi = C_i, C_{i+1}, \dots, C_k, C_1, \dots, C_{i-1}$ in this order.

Once the path touches the next essential cut C on the list, we reflect the polygon using C as an mirror. Thus the tour goes straight through the cut in the reflected copy. In the original polygon P the tour is reflected back at C. The path finding process will stop when it visits the last cut C_{i-1} in the list and it goes back to s. In other words, the reflections with respect to the cuts will generate a sequence of k copies of the polygon P glued along the cuts in the same order π – this glued polygon is denoted by \hat{P} . The minimum watchman tour is found by finding the shortest path inside \hat{P} connecting the starting point s and the image of s in the last copy of P. We can get the watchman route T by mapping this path back to the original polygon.

Once the shortest watchman route R of a polygon P is acquired, the cameras will be installed at every vertex of R. Furthermore, we walk through all the cuts and add cameras at the intersection of T with them only when needed (i.e., if the cameras placed at junctions of R cannot cover $P(C_i)$ for a cut C_i , we add one more camera at the intersection of C_i with R). This set of guards is sufficient to guard P. See figure 3 for example.

Once the cameras are placed, we add extra relay nodes (with or without cameras) along the watchman tour to connect the adjacent cameras. The relay nodes are placed uniformly every r units.

B. Connected Visibility region Planning

The second algorithm, Connected Visibility region Planning (CVP), is to first find an art gallery solution to ensure visibility coverage and then add extra relay nodes along a spanning tree to connect the guards. In a simple polygon



Fig. 3: For a given polygon P with a starting point s with watchman route marked as red line and cameras as red dots. The essential cut C_e of P corresponding to s is marked as red dashed line, the visibility cuts of P according to s are marked as green dashed line. Notice that a guard (in green) is added at the intersection of the watchman tour and one visibility cut.



Fig. 4: The guarding region of v_i (shaded).

without holes, only reflex vertices can block the view. The basic idea of CVP is to deploy cameras such that for each reflex point in P that may obstructs the view, at least one camera can look around it.

Let v_i be a reflex vertex of P, and v_{i-1} and v_{i+1} be its two neighboring vertices, then we define a guarding region $R(v_i)$ as the wedge W_i bounded by extending the two edges $v_{i+1}v_i$ and $v_{i-1}v_i$ inside P. We select a guard within W_i . See Figure 4 for an example.

We choose a hitting set $M = \{m_1, m_2, \dots, m_k\}$ of the guarding regions for all reflex vertices, i.e., any guarding region contains at least one guard in M.

Finding a hitting set for a collection of geometric regions is NP-hard. But one could use a greedy algorithm to find a solution with approximation ratio of $\ln n$ [15]. Specifically, at each stage, we will find a guard that lies in the maximum number of uncovered reflex vertices, until all reflex vertices are covered.

We place cameras at the positions of the guards selected above. We add extra relay nodes to connect them into a connected graph, similar to the case of watchman tour based algorithm. In this case, we use the minimum spanning tree connecting the guards. If any edge in the MST is longer than r (the communication range of these nodes), we deploy relays along this edge uniformly for every distance r.



C. Relay Deployment

The camera nodes are deployed to ensure visibility coverage. To perform monitoring tasks, these camera nodes need to be connected into a wireless camera network. Specifically, Connectivity is achieved when two camera nodes are within each other's communication range. However, the distance between adjacent deployment locations of camera nodes in the MST can exceed their communication range. To ensure reliable wireless connectivity, relay nodes need to be deployed as well. The main functionality of wireless relay nodes is to exchange data messages among camera nodes. For sparse camera node deployments, multiple relay nodes might be deployed to form a connected wireless camera network.

The optimization goals of relay deployment are: 1) forming a connected wireless camera network so that every node can reliably transfer their data to a specified base station; and 2) minimizing the total communication cost in terms of energy consumption for longer system lifetime.

The key parameter here is the communication range. Wireless communication range in real systems is dynamic, affected by the deployment environments. For example, the communication range of a node in an open corridor is longer than that of a node in a closed office. For deployment planning CVP, we can choose a conservative communication range for all nodes that ensure a connected graph. For dynamic camera deployment CVT, variable communication ranges are used for nodes at different locations. The specific communication range can be obtained at deployment time [16].

If relay nodes are equipped with camera sensors, their visibility coverage after deployment can overlap with that of original guards. Therefore, it provides opportunity to remove the original guards while maintaining full visibility coverage. Intuitively, for any camera $g_i \in G$, consider the two, if exists, relays g_j, g_k that have the closest distance from g_i . If $R(g_i) \subseteq R(g_j) \cup R(g_k)$ and $distance(g_j, g_k) < r$, then the camera g_i can be eliminated. An example is shown in Figure 5.

V. SYSTEM IMPLEMENTATION

We implemented a wireless camera network testbed for indoor deployment and evaluation. This testbed has 12 batterypowered wireless camera nodes, each node is built based on the off-the-shelf BeagleBone low power development board from BeagleBoard [17]. The BeagleBone is a credit-cardsized (3.4" 2.1") Linux computer with an AM335x 720MHz ARM processor. We choose the BeagleBone as our deployment platform because of its processor is powerful enough for video processing, but its low power design allows it to be powered by 4 AA batteries. Moreover, it has rich hardware interfaces to expand the board's functionality, including plug-in sensing



Fig. 6: Battery-powered Wireless Camera Node

System Parameters	Value
Maximum Picture Resolution	2048 x 1536
Maximum Frame Rate	30fps
Wireless Communication Range	60+feet
Wireless Bandwidth	54Mbps
Processing Speed	720MHz

TABLE I: Camera Node Specification

components like cameras and wireless radio adapters. A picture of the camera node is shown in Figure 6.

For each camera node, we installed the ngstrm distribution [18], which is an Linux distribution specified for low power embedded devices. For Wi-Fi communication, each node acquired a USB Wi-Fi dongles with Realtek RTL8192CU chipset. We also attempted an alternative setup to run the node by Ubuntu version 13.04 and with Belkin F5D7050v3 USB Wi-Fi dongle. In our observation, there was no discernible difference in the results of the implementation between these two setups.

Each BeagleBone was equipped with a 3.1 megapixel Aptina CMOS digital image sensor MT9T111 via the extension board. This camera integrates on-chip camera function such as anti-shake and auto focus, and is programmable through serial interface. To applied in face identification applications, the horizontal pixel density is approximately 5 pixel/cm. Therefore, by applying the auto focus function, the maximum scene width is 4 meters, with variable monitoring distance within the building (< 50 f eet). Major system parameters are shown in the Table I.

A. Experiments

We firstly plot the floor plan of the Computer and Information Science department building in Temple University as sample input polygon for deployment algorithms. As shown in Figure 7a, the thick line represents the input polygon. In Figure 7b, 7c, and 7d, the black circles represent camera deployment locations and the dash lines represent the wireless links. From these figures, we can clearly see that all these three algorithms require using 8 panoramic cameras to guard



Fig. 7: Simulation with a Real Floor Plan



(a) System Deployment (b) Pictures taken by sensor node

Fig. 8: Sensor Node Deployment

the entire hall way. The actual camera system deployed on the wall is shown in Figure 8.

Next we evaluate the network's wireless connectivity. We set the Wi-Fi mode to be Ad-Hoc, and tune 2.412 Ghz as the communication frequency. The power management function is turned off so that the Wi-Fi communication will be running at highest performance.

For each wireless link in this network deployed in Figure 7b, we execute the "ping" command between each pair of nodes for 50 times and record the packet loss rate. The result is shown in Table II. We can see the lengths of the wireless links range from 8 to 44 feet, and all links have packet loss ratios smaller than 5%.

In order to further establish the wireless link quality, we conducted a long term experiment. We deploy two cameras at the two ends of the corridor, whose length is 43 feet. We ping from one of the node to the other once per second for one hours and record the success rate. This experiment is repeated for three different times. The results are shown in Table III.

From Table III, we can see that the long term wireless connectivity is ensured. The packet loss rates are all smaller

Link ID	Length(feet)	Packet Loss Rate
1	25	4%
2	8	0%
3	30	4%
4	15	0%
5	43	0%
6	13	0%
7	44	0%

TABLE II: CVP Wireless link packet loss rate

ID	Packet Transmitted	Packet Received	Avg RTT	Time Duration
1	3580	3573	16.875 ms	3593893ms
2	3909	3901	18.007ms	3927517ms
3	3693	3689	35.513ms	3722935ms

TABLE III: Long term wireless connectivity

than 1%. Since the length of the wireless link is 43 feet, which is the longest one in the CVP deployment in our department building floor plan, we can see that the CVP deployment ensures high quality wireless connectivity in the long term, and no need for extra wireless relay.

Finally we test the system's power by measuring their voltages and currents when the Wi-Fi module is on and off. We found that the power of a node is about 1.22w and 0.72 when the Wi-Fi is on and off, respectively. An ordinary AA Alkaline long-life battery has a capacity of about 5000 J, therefore, the system is expected to sustain for hours. We configure a node to take pictures and exchange hello messages with its neighbours, and its battery life is above 5 hours.

B. Discussion

Since we adopted visibility sensing model and circular communication models in our deployment algorithms, it is important to validate these models with real experiments. We have tested qualities of sensing and communication using our testbed in real deployment experiments. Experimental results show that both models are valid. However, for effective and efficient camera network deployment, both models can be further improved under realistic constraints.

Visibility Model. Line of sight is a simplified abstraction for optical sensing. In real deployment, various factors influence the optical sensing quality, especially indoor obstacles and light settings. Building construction including interior doors and furniture can block the view of cameras. Solution to this problem is to adaptively adjust the height of camera deployments produced by deployment algorithm. In our experiments, most of the obstacles can be avoided after in-situ tuning during the deployment. Light settings can also significantly affect the quality of pictures taken by the optical sensor. In our building, although the lights are always on 24 hours, the intensity of lights is not consistent in different areas. To obtain clear views of all the areas, extra camera nodes are needed to cover relatively dark areas.

Wireless Communication Model. The realistic communication model is not circular. The communication range is highly directional under complex indoor building structure. The communication range of a node in the corridor is much larger than that of a node in the office room. This observation suggests that relay nodes' positions can be fine tuned during deployment. After nodes are deployed, their communication quality over Wi-Fi is stable over hours. Another issue is that the Wi-Fi channel needs to be tuned at deployment time, to avoid interference with existing wireless networks if any.

From these observations, we can see that it is also important to perform in-situ tuning on sensing and communication after deployment planning.

VI. SIMULATION

A. Experiment Setup

We have implemented a simulation framework for camera deployment algorithm. In the experiments, we take the floor plans of 20 realistic buildings as inputs. These buildings include hotel, classroom, house and museum. The CVP, CVT and 3-Coloring algorithm are implemented. Simulation shows that all these three algorithms are able to ensure full visibility coverage, so we will focus on comparing their camera numbers, relay numbers and wireless transmission power.

We compare our algorithms with the well known 3-Coloring algorithm [19]. In the 3-Coloring algorithm, the input polygon P is firstly triangulated. The vertices of the polygon are then 3-colored in such a way that every triangle has all three colors. Once a 3-coloring is found, the vertices with any one color form a valid guard set, because every triangle of the polygon is guarded by its vertex with that color. Since the three colors partition the n vertices of the polygon, the color with the fewest vertices forms a valid guard set with at most |n/3| guards [20].

B. Camera Number

The camera number is an important metric to evaluate the performance of a deployment algorithm. The camera number is directly related to the construction cost. Besides, as the



Fig. 9: Camera Number



Fig. 10: Relay Number

camera number grows, the video data size also increases. This will cast heavier burden on power supply because wireless data communication is energy expensive. To evaluate the algorithms' performance, we simulate them on the 20 floor plans and record the required camera numbers. The results are shown in Figure 9.

From Figure 9, we can see that the CVP, CVT and 3-Coloring algorithm require 4.2, 4.8 and 5.9 cameras on average, respectively. Besides, the 3-Coloring algorithm's camera number has a standard deviation of 2.1, while that of CVP and CVT are 1.6. This shows that CVP and CVT are able to achieve smaller camera numbers than the 3-Coloring algorithm.

C. Relay Number

In order to ensure the connectivity of the networks, wireless relays are necessary under some conditions. For example, in testbed experiments, we have found that using the Belkin F5D7050v3 USB Wi-Fi dongle, the effective communication range is about 60 feet. However, the communication range depends on many factors. We compare the number of necessary relays when the communication range varies. The results are shown in Figure 10.

TABLE IV: Radio Characteristics

Operation	Energy Dissipated
Transmitter Electronics $(E_{Tx-elec})$	T O T (1).
Receiver Electronics $(E_{Rx-elec})$	50 nJ/bit
$(E_{Tx-elec} = E_{Rx-elec} = E_{elec})$	
Transmit Amplifier (ϵ_{amp})	$100 \ pJ/bit/m^2$

From Figure 10, we can see that the relay numbers drop dramatically when the communication range increases from 10 to 40 feet, but when the communication range is larger than 40 feet, the relay numbers remain relatively stable. The relay numbers are approximately inverse proportional to the communication range.

We can also discover that the CVT has a good performance in terms of relay numbers. On average, it requires 14.4, 6.25, 2.25, 0.95 and 0.6 relays when the communication range is 10, 20, 40, 60 and 80 feet, respectively. This is because the CVT deploys cameras along the shortest watchman route, which is the shortest path that can guard the polygon. Since the total distance is small, the relay numbers is also smaller when compared with CVP and 3-Coloring algorithm.

It is noted that using CVP and CVT algorithm, the wireless links are mostly in direct line of sight, which greatly facilitate the deployment of wireless relays. Besides, when two camera nodes are in direct line of sight, their mutual communication range is larger compared to the condition when obstacles exist because obstacles significantly affect wireless communication quality. In experiment, we can see that using 3-Coloring deployment, obstacles between connected cameras frequently occur.

D. Transmission Cost

The wireless transmission cost is another important issue in the wireless camera network, because the battery capacity is limited and the power consumption of wireless transmission is large. Currently, there is a great deal of research in the area of radio transmission cost in wireless sensor networks. In this paper, we will adopt the radio model constructed in [21]. To transmit a *k*bit message a distance *d* using this radio model, the radio expends:

$$E_{Tx}(k,d) = E_{Tx-elec}(k) + E_{Tx-amp}(k,d)$$

$$E_{Tx}(k,d) = E_{elec} * k + \epsilon_{amp} * k * d^2$$
(1)

and to receive this message, the radio expends:

$$E_{Rx}(k) = E_{Rx-elec}(k)$$

$$E_{Rx}(k) = E_{elec} * k$$
(2)

The physical meaning of the parameters in Equation 1 and 2 are summarized in Table IV.

The camera nodes are modeled after the VIVOTEK CC8130 1MP Panoramic View camera. It is operating at frame rate of 10 fps with image resolution at 1280×800 . The images are encoded in H.264 format and the video will be compressed with a ratio of 30%. Therefore, the data bandwidth of each



Fig. 11: Communication Power

camera is 254 KBit/s. If the camera network keeps operating for 5 hours, then each camera node will generate 4.5Gb of data.

In order to collect these video data, the camera nodes send them link after link to arrive at a data sink. In this experiment, we select one camera node for each camera network to be the data sink, and compute power consumption of the network during the data transmission process.

From Figure 11, we can see that using CVP, CVT and 3-Coloring algorithm to deploy camera networks, the network communication power is 0.18, 0.23 and 0.34 mW, respectively. One reason why CVP achieves lower communication cost than 3-Coloring is that CVP deploys cameras in the internal regions of the polygons, while the 3-Coloring algorithm deploys cameras at the vertices of the polygons. On average, deploying cameras in the internal regions can reduce the mutual distances between the camera nodes. Since the wireless power consumption is proportional to the square of link distance, reducing camera nodes' mutual distances can effectively reduce the communication power.

In summary, CVP and CVT outperform the classical 3-Coloring algorithm in terms of camera number, relay number and communication power. Specifically, CVP can achieve a near optimal performance in necessary camera numbers, while CVT can reduce the relay numbers significantly. Both these algorithms can reduce communication power compared with 3-Coloring algorithm.

VII. RELATED WORK

The Art Gallery problem, finding the minimum number of guards to see a given polygon, is a classical problem in computational geometry and is well known to be NP-hard [7]. It is also well known that $\lfloor n/3 \rfloor$ cameras are occasionally necessary and always sufficient to cover a simple polygon with *n* vertices [22]. In a polygon with *n* vertices and *h* holes, $\lfloor (n+h)/3 \rfloor$ vertex guards are always sufficient [23].

The Watchman Tour Problem deals with finding a route in a simple polygon P such that each point in the interior of Pcan be seen from at least one point along the route[24]. [25] proposes an $O(n^4)$ algorithm to solve the problem of finding the shortest watchman route that is restricted to pass through a starting point s on the boundary of P. [26] gave an $O(f(n)n^2)$ time solution to the problem without restricting to any starting points. The authors of [27] applies the concept of essential cut to solve this problem. They design an algorithm to find out all essential cuts in the polygon, then a simple route that visits all these essential cuts will ensure coverage of the entire polygon.

To solve the camera network deployment problem, previous researchers proposed many different camera models and deployment algorithms. [28] "cone" shape of camera views. The authors of [29] construct a detailed model for the 3dimensional field of view of cameras. [4] introduces a camera deployment scenario that requires that neighbouring cameras should have hand-off regions. [30] models obstacles in a probabilistic way, and designs a deployment algorithm that ensures visibility coverage with high confidence. However, all these papers only focus on visibility coverage, and none of these papers consider about wireless connectivity. Instead, in our paper we assume the camera's sensing range is only restricted by line of sight, and we optimize the communication costs while ensuring visibility coverage.

VIII. CONCLUSION

We formally define the *Connected Minimum Guarding Set Problem.* We proved the hardness of the problem and designed 2-approximation algorithm in the geometric setting. We also developed and tested two algorithms to solve this problem using watchman tour and art gallery problem respectively. Experiments are conducted on an implemented prototype of the proposed system to evaluate the feasibility and the effectiveness of the system on ensuring both visibility coverage and wireless connectivity. Simulations are also conducted to evaluate the performance of the proposed algorithms on realistic floor plans. The results verify that the proposed algorithms can ensure visibility coverage and reduce camera numbers and communication power significantly.

REFERENCES

- Y. Yao, C.-H. Chen, B. Abidi, D. Page, A. Koschan, and M. Abidi, "Can you see me now? sensor positioning for automated and persistent surveillance," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 1, pp. 101–115, 2010.
- [2] C. K. Cowan and P. D. Kovesi, "Automatic sensor placement from vision task requirements," *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on*, vol. 10, no. 3, pp. 407–416, 1988.
- [3] S. Ram, K. Ramakrishnan, P. Atrey, V. Singh, and M. Kankanhalli, "A design methodology for selection and placement of sensors in multimedia surveillance systems," in *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*. ACM, 2006, pp. 121–130.
- [4] Y. Yao, C.-H. Chen, B. Abidi, D. Page, A. Koschan, and M. Abidi, "Sensor planning for automated and persistent object tracking with multiple cameras," in *Computer Vision and Pattern Recognition*, 2008. *CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.
- [5] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems 5*. Springer, 2002, pp. 299–308.
- [6] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, 2005.

- [7] J. O'rourke, Art gallery theorems and algorithms. Oxford University Press Oxford, 1987, vol. 1092.
- [8] W. Chin and S. Ntafos, "Optimum watchman routes," in 2nd Proc. ACM Symposium on Computational Geometry. New York, New York, USA: ACM Press, 1986, pp. 24–33.
- [9] W. P. Chin and S. Ntafos, "Shortest watchman routes in simple polygons," *Discrete & Computational Geometry*, vol. 6, no. 1, pp. 9–31, 1991.
- [10] W. Chin and S. Ntafos, "Optimum watchman routes," *Inform. Process. Lett.*, vol. 28, pp. 39–44, 1988.
- [11] C. S. Mata and J. S. B. Mitchell, "Approximation algorithms for geometric tour and network design problems (extended abstract)," in *Proceedings of the eleventh annual symposium on Computational* geometry, ser. SCG '95. New York, NY, USA: ACM, 1995, pp. 360– 369. [Online]. Available: http://doi.acm.org/10.1145/220279.220318
- [12] X. Tan, T. Hirata, and Y. Inagaki, "An incremental algorithm for constructing shortest watchman routes," *International Journal of Computational Geometry & Applications*, vol. 3, no. 04, pp. 351–365, 1993.
- [13] X. TAN, "Corrigendum to an incremental algorithm for constructing shortest watchman routes," *Int. J. Comput. Geom. Appl.*, vol. 9, pp. 319–323, 1999.
- [14] X. Tan, "Fast computation of shortest watchman routes in simple polygons," *Information Processing Letters*, vol. 77, no. 1, pp. 27–33, 2001.
- [15] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of operations research*, vol. 4, no. 3, pp. 233–235, 1979.
- [16] H. Liu, J. Li, Z. Xie, S. Lin, K. Whitehouse, J. A. Stankovic, and D. Siu, "Automatic and robust breadcrumb system deployment for indoor firefighter applications," in *Proceedings of the 8th international conference on Mobile systems, applications, and services.* ACM, 2010, pp. 21–34.
- [17] Beagleboard. [Online]. Available: http://beagleboard.org
- [18] The ngstrm distribution. [Online]. Available: http://www.angstromdistribution.org
- [19] J. Urrutia, "Art gallery and illumination problems," Handbook of computational geometry, pp. 973–1027, 2000.
- [20] S. Fisk, "A short proof of chvátal's watchman theorem," Journal of Combinatorial Theory, Series B, vol. 24, no. 3, p. 374, 1978.
- [21] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energyefficient communication protocol for wireless microsensor networks," in System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on. IEEE, 2000, pp. 10–pp.
- [22] V. Chvátal, "A combinatorial theorem in plane geometry," J. Combin. Theory Ser. B, vol. 18, pp. 39–41, 1975.
- [23] F. Hoffmann, M. Kaufmann, and K. Kriegel, "The art gallery theorem for polygons with holes," in *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, 1991, pp. 39–48.
- [24] X. Tan, "Fast computation of shortest watchman routes in simple polygons," *Information Processing Letters*, vol. 77, no. 1, pp. 27–33, 2001.
- [25] W.-p. Chin and S. Ntafos, "Optimum watchman routes," *Information Processing Letters*, vol. 28, no. 1, pp. 39–44, 1988.
- [26] B. J. Nilsson and S. Carlsson, Guarding art galleries: Methods for mobile guards, 1994.
- [27] S. Carlsson, H. Jonsson, and B. J. Nilsson, "Finding the shortest watchman route in a simple polygon," *Discrete & Computational Geometry*, vol. 22, no. 3, pp. 377–402, 1999.
- [28] A. T. Murray, K. Kim, J. W. Davis, R. Machiraju, and R. Parent, "Coverage optimization to support security monitoring," *Computers, Environment and Urban Systems*, vol. 31, no. 2, pp. 133–147, 2007.
- [29] J. Zhao, S. Cheung *et al.*, "Multi-camera surveillance with visual tagging and generic camera placement," in *Distributed Smart Cameras*, 2007. ICDSC'07. First ACM/IEEE International Conference on. IEEE, 2007, pp. 259–266.
- [30] A. Mittal and L. S. Davis, "Visibility analysis and sensor planning in dynamic environments," in *Computer Vision-ECCV 2004*. Springer, 2004, pp. 175–189.