Connected Wireless Camera Network Deployment with Visibility Coverage

HUA HUANG, Stony Brook University
CHIEN-CHUN NI, Yahoo Research
XIAOMENG BAN, Google. Inc
ANDREW T. SCHNEIDER, Temple University
JIE GAO, Rutgers University
SHAN LIN, Stony Brook University

System deployments of IoT systems have drawn research attention, because it is very challenging to meet both physical and cyber constraints in real systems. In this paper, we consider the problem of deploying wireless camera networks inside a complex indoor setting for surveillance applications. We formulate the problem of the minimum connected guarding network whose objective is to place a minimum number of cameras satisfying both visual coverage of the domain and wireless network connectivity. We prove that finding the minimum connected guarding network is NP-hard in both the geometric and discrete settings. We also give a 2-approximation algorithm to the geometric minimum guarding network problem. Motivated by the connection of this problem with the watchman tour problem and the art gallery problem, we developed two algorithms to calculate the locations of camera deployment. By deploying a prototype testbed, we verify the feasibility of the system design. Using simulations on 20 real floor plans, we demonstrate that our solutions reduce the number of cameras by up to 28%, and reduce the number of relay nodes by up to 47%.

CCS Concepts: • **Networks → Sensor networks**; • **Theory of computation → Computational geometry**.

Additional Key Words and Phrases: Camera Network Deployment, Visibility Coverage, Wireless Connectivity

ACM Reference Format:
Hua Huang, Chien-Chun Ni, Xiaomeng Ban, Andrew T. Schneider, Jie Gao, and Shan Lin. 2020. Connected Wireless Camera Network Deployment with Visibility Coverage. 1, 1 (May 2020), 19 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With the recent advancement of pervasive computing, wireless networks, and optical sensing, wireless camera networks are deployed for a wide range of applications [31]. In many existing systems for security 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. However, such support is unavailable in many scenarios like first responder [38] and military applications, where the networks need to be deployed quickly with

Authors' addresses: Hua Huang, hua.huang@stonybrook.edu, Stony Brook University, Stony Brook, New York, 11790; Chien-Chun Ni, cni02@verizonmedia.com, Yahoo Research; Xiaomeng Ban, snoowball@gmail.com, Google. Inc; Andrew T. Schneider, atschneider@temple.edu, Temple University, Philadelphia, Pennsylvania, 19130; Jie Gao, jg1555@cs.rutgers.edu, Rutgers University, Piscataway, New Jersey, 08854; Shan Lin, shan.x.lin@stonybrook.edu, Stony Brook University, Stony Brook, New York, 11790.

XXXX-XXXX/2020/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

little existing infrastructure. For example, soldiers need to deploy a camera network to monitor suspicious activities in a building during military operations. These applications impose a unique set of cyber and physical requirements on the deployment: 1) Full visibility coverage: every point inside the building needs to be monitored; 2) Reliable wireless connectivity: all the camera nodes need to be self-organized to form a connected ad-hoc network, so that pictures or videos recorded can be transferred to a base station for real time monitoring when no wireless infrastructure support is available; 3) Low deployment cost: the minimum number of devices is desirable for the low deployment cost and short deployment time. These constraints, together with the complex building structure, make the camera network deployment problem very challenging.

In order 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 models into an optimization framework. In the visibility model, each camera's sensing range is only restricted by the line of sight. This is a generalization of the cone model [61] of a camera. In the wireless communication model, wireless nodes are connected when they are within each others' communication range r. Based on these two models, the problem looks for the minimum number of cameras that can guard a floor, while ensuring their mutual wireless connectivity.

This problem includes the integration of isotropic sensing and wireless networking, which indicates the unique cyber-physical nature of the deployment. Previous researchers have proposed many camera deployment algorithms [15][46][60][2][23][34][54] to maximize the visual coverage of the network, but little attention was given to ensure the wireless connectivity, which is essential for data communication. Although related sensor network coverage research [28][30] provides valuable insights to network connection, the widely used short range circular sensing model does not apply for optical cameras. Our goal is to fill this missing gap and achieve full visibility coverage and wireless connectivity with the minimal number of necessary devices.

The solution to the minimum connected guarding network problem depends on the communication range between the wireless nodes. Take one extreme, say the communication range of the sensor nodes is large enough such that any two of them can directly communicate with each other. Then this 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 [45]. Take the other extreme, say that the communication range is very small compared to the scale of the building; we need to place the wireless nodes continuously along paths 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 the 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 [10], i.e., finding a route of the minimum length that guards an entire polygon. It is known that the watchman route problem is NP-hard for the general polygon with holes [13]. However, nothing is known about the problem if nodes are not connected as a path but a tree. Recently, alternative proofs about the NP-hardness and the approximation algorithm for the minimum guarding tree problem was provided in [16].

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

In terms of algorithm development for practical implementations, we consider two possible deployment settings. In the first setting, we focus on the first responder applications in which in-field dynamic deployment must be done quickly. We employ the watchman tour based solution, termed the Connected Visibility region Tracking (CVT), which allows the first responders to drop sensors along the (shortest) watchman routes. This can minimize the total traveling distance of the first responder during the deployment process. In the second setting, we assume that the network designer has plenty of time to deploy the network. We design the Connected Visibility region Planning (CVP) algorithm, which first identifies a set of guarding regions as potential deployment areas. The Minimum Steiner Tree-based algorithm is used to deploy relay points to ensure wireless connectivity. This algorithm is computationally intensive but produces near-optimal deployment results in practice.

When the wireless communication range is not long enough, additional wireless camera nodes are needed to ensure network connectivity. We firstly design a Minimum Spanning Tree with Neighbors (MSTN) based algorithm to optimize the locations of the cameras in order to reduce the communication distances between camera nodes. Then we apply two algorithms to deploy wireless relay nodes. The first algorithm we use, termed Uniform, is to deploy wireless devices uniformly along with wireless links [31]. This algorithm ensures that the number of wireless nodes is at most 4 times the optimum [8][37]. The second algorithm, termed the Steiner tree with minimum number of Steiner points (Steiner) algorithm [37][9], searches for more effective locations (Steiner points) for wireless nodes such that more than two components of the network can be connected. This algorithm improves the approximation ratio to 3.

We evaluated the proposed deployment algorithms in real scenarios. We built a wireless camera network testbed to validate the effectiveness and accuracy of our algorithms. Each node of this testbed consisted of four off-the-shelf components: a Beagle Board development board, a Wi-Fi radio adapter, a webcam, and a battery cape. We deployed these camera nodes according to locations generated from the CVP and the CVT algorithm in our campus buildings. Real deployments over time can achieve above 99% wireless connectivity and adequate visibility coverage (our prototype cameras were not panoramic). Compared with the widely used 3-coloring and the random deployment algorithms, our deployments had significantly better wireless communication connectivity. By incorporating wireless transmission power models, we demonstrate that our algorithms reduce the energy consumption rate by up to 30%. Moreover, in simulations on 20 real floor plans, with different communication ranges, we show that our deployment algorithms yield more efficient results, by reducing camera numbers by up to 25%, and significantly reducing the number of wireless cameras under different communication range assumptions.

2 PROBLEM ANALYSIS AND APPROXIMATION ALGORITHM

In this section, we formulate the wireless camera deployment problem and prove its NP-hardness. We assume that the indoor deployment area is modeled by a polygon P. A camera node is a wireless node with a *visual sensing range* defined by line of sight, and a *wireless communication range* defined as a disk of radius r. We would like to place a minimum number of cameras inside P such that the following two conditions are met:

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

This problem is called the *Minimum Connected Guarding Network* problem. Clearly, if the input polygon is convex, then the camera deployment problem is trivial, because placing one camera at 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.

2.1 Hardness Proof

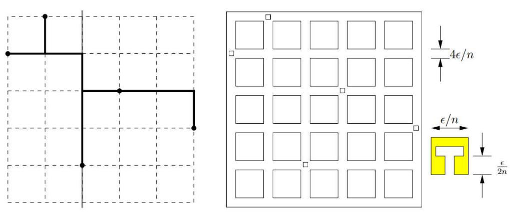


Fig. 1. Given an instance of the minimum geometric rectilinear Steiner tree (left), we turn it to an instance in which each point is replaced by a T-junction obstacle (shown to the right).

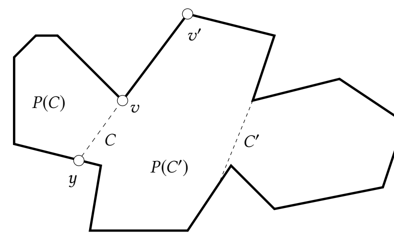


Fig. 2. Cut C dominates C' since $P(C')$ contains $P(C)$ – any tour that visits $P(C)$ must also visit C' and guard $P(C')$. C is also an essential cut.

Theorem 1. *Finding the minimum connected guarding network in a general polygon P is NP-hard.*

Proof. We use a reduction 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 disk of radius r. The optimal solution for the art gallery problem does not change. But the cameras in any guarding solution form a connected network. 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. □

Notice that the proof above depends on the communication range being a fixed constant. When the communication range is much smaller than the size of the deployment domain, i.e., $r \longrightarrow 0$, the connected guarding 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.*

The major difference between *geometric guarding network* and *connected guarding network* is whether or not the guarding points are deployed continuously along the graph. In a geometric guarding network, the guarding points are deployed very closely to their immediate neighbors, and the resulting network is a geometric graph that any point on this graph is a guarding point. On the other hand, in a connected guarding network, each vertex represents a guarding point, and the edges represent the communication links between different vertices. In what follows, we prove the NP-hardness of the minimum geometric guarding network problem.

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. This is illustrated 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 even for a rectilinear polygon with holes. □

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

2.2 Approximation Algorithm for the Minimum Geometric Guarding Network Problem

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

We represent the input polygon P by a sequence of vertices $v_1, v_2, \cdots, v_n$, with $n \geq 4$. For $i = 1, 2, ..., n-1$, $e_i = (v_i, v_{i+1})$ represents the 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. Without loss of generality, we also assume that the vertices of P are in general positions, 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 call a set of points M inside P 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 guarding network is a tree of polygonal curves.*

Proof. For 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 $v_i v_{i+1}$, to form two cuts, c_{i-} and c_{i+}. We add intersections between each cut c_i and the guarding network G into M. Clearly the number of guards is at most $O(n^2)$. Further, the set of points 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. □

The idea to get a 2-approximation solution to the minimum geometric guard network in a simple polygon P is to make use of a watchman tour. 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 [11]. Although finding the shortest watchman tour in a general polygon with holes is NP-hard [11], there is an $O(\log n)$-approximation algorithm for a rectilinear version with restricted visibility [40], and an $O(\log^2 n)$-approximation algorithm for general polygons with or without holes. The watchman tour problem for a simple polygon is solvable in polynomial time (for a tour with a fixed starting point see [50, 53], and for the floating tour without a given starting point [51]). We show in the following theorem that in a simple polygon, the optimal watchman tour is a 2-approximation solution to the minimum geometric guarding network.

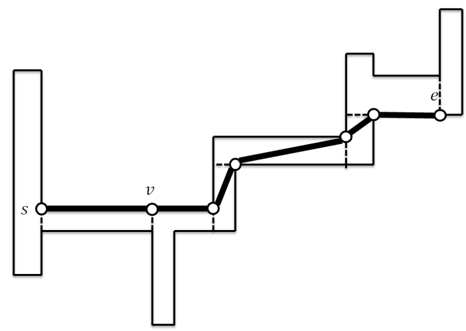


Fig. 3. For a given polygon P with a starting point s, the watchman route is represented by the thick line, and the cameras are represented by circular dots. The essential cuts and visibility cuts of P corresponding to s are marked as a dashed lines. Notice that a guard v is added at the intersection of the watchman tour and one visibility cut.

Theorem 4. *Inside a simple polygon P, the optimal watchman tour is a 2-approximation to the minimum geometric guarding network.*

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. □

3 ALGORITHMS FOR MINIMUM CONNECTED GUARDING NETWORK

In this section we describe algorithms for finding a 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 for two different scenarios. The first algorithm is used when there is limited deployment time and the network needs to be set up quickly. For this case, our algorithm computes the minimum length route for the deployer to travel and drop camera nodes along the route [38] without the need for much detours. The second algorithm is used when the deployment time is more ample and the deployer can install the cameras anywhere in the building. In this case our algorithm will further optimize the locations of the wireless camera nodes so that they are more efficient in visibility coverage and wireless connectivity.

3.1 Connected Visibility region Tracking

We firstly describe the Connected Visibility region Tracking (CVT) algorithm. The basic idea of this algorithm is to find a minimum watchman tour and place cameras along the tour, such that the same visibility coverage is kept. The first step of CVT is to compute the optimum watchman tour for the input polygon. We adapt the algorithm by Chin *et al.* [13] and Tan *et al.* [50] with $O(n^4)$ runtime to find the shortest watchman route for a simple polygon through a given point s within the polygon. The basic idea is to find the 'essential cuts' in P that the watchman route must touch to guard the whole polygon, and visit these cuts using a shortest tour.

Specifically, the concepts of a cut C, an essential piece $P(C)$ and an essential cut are shown in Figure 2. For a given polygon P, suppose v is a reflex vertex in P and one of its adjacent vertices is v'. If 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 it separates 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

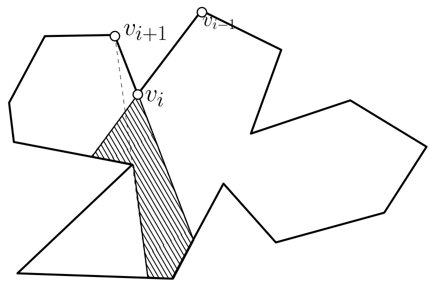


Fig. 4. The guarding region of v_i (shaded)

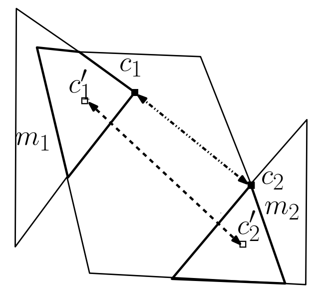


Fig. 5. Different Camera Deployment Locations

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 guarded automatically. We call a cut to be an *essential cut* if it is not dominated by any other cuts. Two cuts C_1 and C_2 can also intersect. If they intersect, i.e., the essential piece $P(C_1)$ and $P(C_2)$ partially overlap, we will only consider the region that is outside the union of the two $P(C_1) \cup P(C_2)$. And each cut is shortened to be only the part on the boundary of $P(C_1) \cup P(C_2)$.

Based on the concepts of essential cut, we can reduce the watchman route problem to finding the shortest route that touches every essential cut inside a polygon. We first list the essential cuts in clockwise order, $\{C_1, C_2, \cdots, C_k\}$. Starting from point s, we want to find a path to visit this cut list. It has been proved that the optimal watchman tour will visit the cuts in a clockwise order, making a 'bounce' at each cut similar to a ray bouncing off a mirror. There is still one unknown parameter as which cut is visited first from s. We simply enumerate all possibling starting cuts. We reorder this list so that the first cut being visited comes first in the list. Specifically, if the first cut to visit is C_i, then the tour visits $\pi = C_i, C_{i+1}, \cdots, C_k, C_1, \cdots, C_{i-1}$.

Once the path touches the next essential cut C on the list, we calculate the reflection image of the polygon using C as a mirror − the watchman 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. We denote this glued polygon 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, a camera will be installed at every vertex of R. Furthermore, we walk through all the intersections between R and each of the polygon cut, and add cameras 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 ensure the visibility coverage of P. See Figure 3 for an example.

If the network is not connected, additional wireless camera nodes are necessary. To reduce the detour for the network deployer, we add extra nodes along the watchman tour to connect the adjacent cameras. The camera nodes are placed uniformly every r units.

3.2 Connected Visibility region Planning

The second algorithm is called Connected Visibility region Planning (CVP). The basic idea is to determine guarding regions that can look around reflex vertices in the polygon, and place a camera in each guarding region to ensure visibility coverage. Then we deploy wireless relay nodes to ensure wireless connectivity using the Steiner Minimum Tree with Minimum number of Steiner Points and bounded edge length algorithms.

In a simple polygon 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 obstructs the view, at least one camera can look around it. Specifically, let v_i be a reflex vertex of P, and v_{i-1}, v_{i+1} be its two neighboring vertices. We define a *guarding region* $W(v_i)$ as the region such that any point within $W(v_i)$ has line of sight to both v_{i-1} and v_{i+1}. See Figure 4 for an illustration.

After obtaining the guarding regions for all the reflex vertices, we further reduce the redundancy by selecting only the intersections between these visibility regions. We calculate the intersection guarding region set $\hat{W} = \{w_1, w_2, ...w_n\}$, such that $w_1 \cup w_2 \cup ... \cup w_n = W(v_1) \cup W(v_2) \cup ... \cup W(v_N)$, and $w_i \cap w_j = \emptyset, \forall i \neq j$. Some elements in the set $\hat{W}$ are intersections of more than one elements in $W(v_i)$, while others are partial portions of $W(v_i)$.

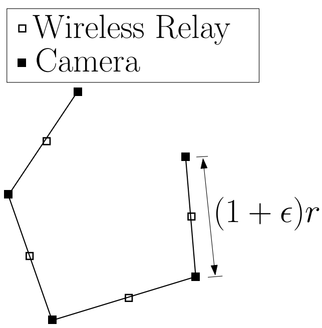
Our goal is to find the minimum region set $M = \{m_1, m_2, \cdots, m_k\}$ from the intersection guarding region set $\hat{W}$, such that the entire polygon is covered. Finding such a deployment region set for a collection of geometric regions is NP-hard. We adopt a greedy algorithm to achieve this goal. Initially, we set the guarding set G to be empty. At each iteration, we find a deployment region $\hat{w}$ from $\hat{W}$ that is contained in the maximum number of uncovered guarding regions, deploy a camera g' at the center of $\hat{w}$, and add g' into the guarding set G. Then we check the visibility coverage area of the guarding set G, if it covers the entire polygon, then the algorithm terminates. Otherwise we select the next deployment region $\hat{w}$ and repeat the procedure.

3.2.1 Optimizing Camera Locations. By deploying one camera at the centroid of each deployment region m_i, the visibility coverage of the entire area is ensured. However, since we have the freedom to select locations inside each deployment region, we can deploy them in such a way that the total length of the spanning tree of the cameras is reduced. This is illustrated in Figure 5. In this simple example, there are two deployment regions, and we need to deploy one camera at each of them. To achieve the minimum distance between cameras, we should deploy cameras at point c_1 and c_2. On the other hand, if we deploy the cameras at the centroids of m_1 and m_2, i.e., c_1' and c_2', their mutual distance will be larger.

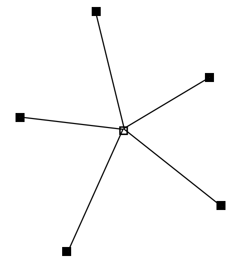
Specifically, our goal is to determine the location of the camera within each deployment region m_i, such that the total length of the spanning tree of these cameras is minimized. This is reduced to the problem of Minimum Spanning Tree with Neighborhoods, and we adopt the algorithm proposed in [59]. For any two regions m_i and m_j, we define their distance d_{ij} as the minimum Euclidean distance from any points in m_i to any point m_j. We use s_{ij} to denote the shortest segment that connects points in region m_i and m_j. Let S be the set of $n^2 - n$ distance segments of the set of deployment regions M, called the distance set. We say regions m_i and m_j are connected if the segment s_{ij} is selected in a tree. The minimum connecting tree T is defined as the subset of S such that all the regions are connected and the total length $\sum_{s_{ij} \in T} d_{ij}$ is minimized.

We use the standard Minimum Spanning Tree algorithm to find the minimum connecting tree T from the distance set S. After the minimum connecting tree T is obtained, let $n_{i1}, n_{i2}, ...n_{ik}$ denote the set of endpoints of distance segments from T that are incidental to the region m_i (k represents the number of segments incidental to region m_i). We randomly select one point n_i^c from these points as the location for camera deployment. We repeat this process until one camera is placed in each region m_i. We denote this set of cameras by N.

3.2.2 Ensuring Wireless Connectivity. After the camera set N is deployed to ensure visibility coverage, our next goal is to form a connected network. Different from the scenario of CVT when

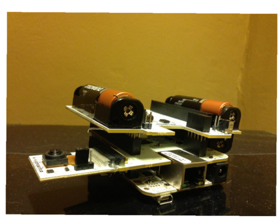


(a) Uniform Deployment

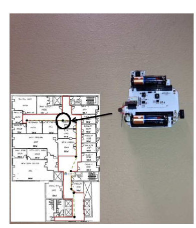


(b) Steiner Tree Deployment

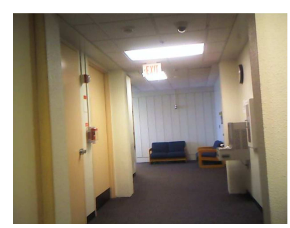
Fig. 6. Deployment of Additional Wireless Nodes to Ensure Network Connectivity.



(a) Battery-powered Wireless Camera Node



(b) System Deployment



(c) Pictures taken by sensor node

Fig. 7. Sensor Node Deployment

the deployment time is limited, we now consider the case when there is ample time so that the locations of the wireless nodes can be further optimized.

Specifically, we formulate the problem of deploying additional wireless camera nodes as what follows: given the set of wireless cameras N in the area and the upper bound r of the transmission range, compute the minimum number of wireless nodes such that all the wireless cameras form a connected network T_s. This problem is called the *Steiner Minimum Tree with Minimum number of Steiner Points and bounded edge length* [8][9], which is an NP-hard problem. There are several approximation solutions for it, with constant approximation ratios.

A basic algorithm is to deploy wireless nodes every r distance along the Minimum Spanning Tree (MST) of the existing camera nodes. This algorithm in theory achieves a 4-approximation ratio [8]. In particular, for any wireless link that has a length l greater than r, we deploy $\lfloor l/r \rfloor$ relay nodes uniformly along the link. This is illustrated in Figure 6a. We can see that when the distance between two cameras (adjacent black blocks) is larger than r, we will deploy an additional relay node in the middle.

To reduce the number of wireless relay nodes, we apply the Steiner Tree based camera deployment algorithm. The intuition is that wireless nodes can be used more efficiently by connecting multiple components of the network. This is illustrated in Figure 6b. If we generate the minimum spanning tree of the camera nodes and deploy wireless nodes along edges, then we will need 4 additional nodes to ensure connectivity (Figure 6a). However, for this case, one additional wireless node will be sufficient to connect the entire network, as is shown in Figure 6b. Based on this idea, we design a Steiner Tree based algorithm to reduce the number of wireless nodes.

The algorithm proceeds as follows. Initially, the network topology T_s consists of all the camera nodes, with no edges. Through three loops, we include necessary wireless nodes into the network and compute the topology of the network.

Loop 1: In the first loop, we find out camera node pairs that do not need any relay nodes. We scan the edges $e_i \leq r$ in the order of increasing lengths. We include edge e_i if the two endpoints of e_i belong to different connected components of T_s.

Loop 2: In the second loop, we find out all the potential relay node positions that can connect more than two components in the network. We scan all camera node triples $\{c_i, c_j, c_k\} \in N^3$. If c_i, c_j, and c_k are in three different connected components of the graph T_s, then we check whether there exists a Steiner point v_s such that these three components can be connected to it. Such a point v_s is found by using the following approach: If $\triangle c_i c_j c_k$ is an acute triangle, v_s is the circumcenter of $\triangle c_i c_j c_k$. Otherwise if $\triangle c_i c_j c_k$ is an obtuse or right triangle, then v_s is the center point of the longest edge of $\triangle c_i c_j c_k$.

If the longest distance of these three nodes to v_s is shorter than the communication range r, it means that all the three components can be connected using a single wireless node, so a wireless camera v_s and the three edges $\overline{v_s c_i}$, $\overline{v_s c_j}$, and $\overline{v_s c_k}$ are included in the topology T_s.

In some cases v_s is located outside of the polygon P. In such cases, we scan all reflex vertices of P that are inside of $\triangle c_i c_j c_k$. If any of these reflex vertices has direct line of sight to c_i, c_j and c_k and has distances at most r to c_i, c_j and c_k, then a camera node is placed at that point. Otherwise this camera triple is skipped.

Loop 3: In the final loop, we deploy relay nodes uniformly to ensure wireless connectivity. We scan all the edges e_i that have larger length than r in increasing order. If any edge e_i connects two different components of T_s, then we deploy wireless nodes along e_i every r distance. This loop proceeds until all the devices are able to form a connected network. Using this approach, we can achieve 3-approximation ratio [9].

3.3 Discussion

Extension to Polygons with Holes. The CVT and CVP algorithm are designed for simple polygons. However, in terms of application, the cases when internal holes exist are also of interest. The first step of the CVT algorithm is to find the shortest watchman route, which is NP-hard when the polygon has holes [17]. For a polygon P with diameter $diam(P)$, perimeter $per(P)$ and k holes, an approximation algorithm can achieve a length of $O(per(P) + \sqrt{k} \cdot diam(P))$ relative to the optimal solution. Therefore, the CVT algorithm can be adapted by using an watchman route algorithm designed for polygon with holes. To extend the CVP algorithm to general polygons, it's necessary to define the guarding regions for the internal holes that can also obstruct view. Once the guarding regions are defined, we can use the same region intersection algorithms described in Section 3.2 to determine the camera locations.

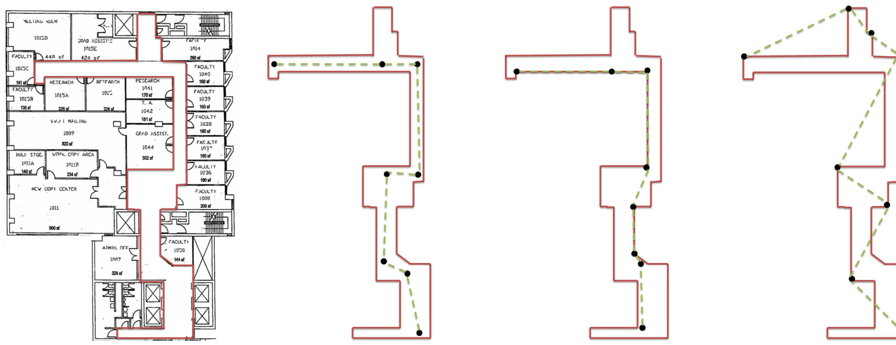
Time Complexity. The computation complexity of finding the minimum watchman route is $O(n^4)$[13, 50]. This determines the overall big-O time complexity of CVT. To find the set of guarding regions in CVP, the time complexity is $O(n^2)$ ($O(n^2)$ to find the mutual intersections of all guarding regions, $O(n \log n)$ time for sorting, and $O(n)$ time to find a subset using greedy algorithm). To determine the location of camera deployment, the Minimum Spanning Tree with Neighborhoods algorithm is used. Its time complexity is $O(n^2 \log n)$. For the relay node deployment algorithm, the time complexity will be $O(n^3)$. Therefore, the overall big-O time complexity of CVP is $O(n^3)$.

4 SYSTEM IMPLEMENTATION

In this section we introduce the design of our prototype wireless camera network testbed. Our goal is to demonstrate the ability of the system to ensure basic wireless connection. This testbed has

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

Table 1. CVP Wireless link packet loss rate



(a) Floor Plan (b) CVP (c) CVT (d) 3-Coloring

Fig. 8. Deployment Locations in a Real Floor Plan

12 battery-powered wireless camera nodes, and each of them is built based on the off-the-shelf BeagleBone low power development board. A 3.1 megapixel Aptina CMOS digital image sensor MT9T111 and a USB Wi-Fi dongles with Realtek RTL8192CU chipsets are plugged to the board's USB extension board, and the board is powered by 4 AA batteries. A picture of the camera node is shown in Figure 7a.

Experiments. We firstly conduct ad hoc node to node communication between wireless camera nodes. We have deployed our prototype system in the building of the Computer and Information Sciences Department building in Temple University, which is shown in Figure 8. In these figures, the thick line represents the input polygon. In Figure 8b, 8c, and 8d, the black dots represent camera deployment locations and the dash lines represent the wireless links. The actual camera system deployed on the wall is shown in Figure 7b. A sample picture taken by this system is shown in Figure 7c.

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 8b, we execute the ping command between each pair of nodes 50 times and record the packet loss rate. The result is shown in Table 1. We can see the lengths of the wireless links range from 8 to 44 feet, and most links achieve 100% delivery rate. The only exception is in link 1, where there is a 4% packet loss rate. One possible reason for the packet loss is that there are lots of other WiFi devices being used in the office during the experiments. The mutual interference between our system and the other WiFi devices can cause packet loss.

In order to further establish the wireless link quality, we conduct a long term experiment. We deploy two cameras at the two ends of the corridor, whose length is 43 feet. We ping from one node to the other every second for one hour and record the success rate. This experiment is repeated three different times. All three sessions of experiment achieve the packet loss rates smaller than 1%, and the average round trip times are 16.875, 18.007, and 35.513, respectively. 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.

Finally we test the system's power by measuring each node's voltage and current when the Wi-Fi module is on and off. We found that the power of a node is about $1.22w$ and $0.72w$ 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 with 4 batteries is expected to sustain for hours. We configure a node to take pictures occasionally and exchange hello messages with its neighbors, and its battery life is above 5 hours.

5 SIMULATION

5.1 Experiment Setup

We have implemented a simulation framework to evaluate our algorithms. In the experiments, we take the floor plans of 20 realistic buildings as input. These buildings include hotels, classrooms, houses, and museums. The goal of the evaluation is two-folded. Firstly we need to verify that both the CVP and CVT algorithms can ensure the full visibility coverage of the floor plans. Secondly we need to compare the number of cameras, wireless transmission power, and number of relay nodes. In each specific floor plan, the numbers of cameras and relay nodes are positive integers. To evaluate the algorithm performance, we compute statistics of node numbers, which can be fractional numbers.

We selected the 3-Coloring algorithm [55] and the random deployment algorithm[39], which are two classical algorithms that are widely used in sensor deployment literature, as baseline algorithms. 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. By choosing a color with fewest vertices, this algorithm forms a valid guard set with at most $\lfloor n/3 \rfloor$ guards [20]. Then we compute the minimum spanning tree of the guard set, and deploy wireless nodes every r distance along each link. In the random deployment algorithm, we randomly select the deployment locations for cameras one by one inside the building, until the entire area is covered [39].

5.2 Number of Cameras

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 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.

We can see that the CVP, CVT, the 3-Coloring algorithm, and the random deployment algorithm require 4.2, 4.8, 5.9, 14.75 cameras on average, respectively. This demonstrates that the CVP and CVT algorithms achieve more efficient use of cameras. The 3-Coloring algorithm always places cameras at the vertices of the room, which limits the cameras' guarding areas. The random deployment algorithm requires more cameras because their locations are randomly selected, and there exists much redundancy in the monitoring areas of the cameras. On the contrary, the CVP and CVT algorithms deploy cameras such that larger areas can be monitored. As a result the total number of cameras is reduced.

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 the fluctuation of the camera numbers with different structures is reduced when using CVP and CVT.

5.3 Number of Relay Nodes

In this experiment, we simulate and find out the number of wireless cameras necessary to ensure both visibility coverage and wireless connectivity in each of the 20 floor plans. When the camera

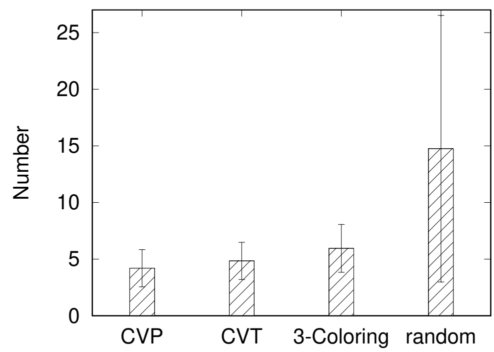


Fig. 9. Camera Number

nodes cannot form a connected network, we deploy additional wireless relay nodes uniformly along the wireless link to ensure connectivity. In testbed experiments, we have found that using the Belkin F5D7050v3 USB Wi-Fi dongle, the effective communication range is about 60 feet. When the distances between nodes are larger, packet loss rates begin to grow. Since the communication range depends on many factors, we compare the number of wireless relay nodes when different assumptions on communication range are used. The results are shown in Figure 10.

From Figure 10, we can see that the number of wireless relay nodes drops dramatically when the communication range increases from 10 to 40 feet, but when the communication range is larger than 40 feet, it remains relatively stable. The number of wireless relay nodes is approximately inversely proportional to the communication range.

We can also discover that the CVT has a good performance in terms of number of wireless relay nodes. On average, it requires 19.2, 11.05, 7.05, 5.75 and 5.4 nodes when the communication range is 10, 20, 40, 60 and 80 feet, respectively. This is because the CVT deploys wireless relay nodes along the shortest watchman route, which is the shortest path that can guard the polygon. Since the total distance is small, the number of wireless nodes is also smaller when compared with the CVP and the 3-Coloring algorithm.

We can see that the random deployment algorithm requires the largest number of relay nodes, 78 when the communication range is 10 feet, to ensure connectivity. This is because a large number of cameras result in larger total distances of the wireless links, which is linearly correlated to the number of relay nodes when the communication range is small. The 3-Coloring algorithm also requires a larger number of wireless nodes to ensure connectivity, because all the cameras are deployed at the vertices of the building, which results in larger total distances of the wireless communication links. On the other hand, the CVT and CVP algorithms are able to a achieve a much better performance in number of wireless camera nodes, because they are optimized in this aspect.

5.3.1 Wireless Relay Node Deployment Algorithms. Next we evaluate the performance of the wireless relay node deployment algorithms, the Minimum Spanning Tree with Neighbors (MSTN) and the Steiner Minimum Tree algorithms. In this experiment, we normalize the building sizes by scaling all floor plans to be within the area 0.5×1. We apply the CVP algorithm to find the deployment regions, then we find the locations of cameras either by using the MSTN algorithm, or by deploying a camera at the centroid (Centroid) of each deployment region. After the locations of cameras are found, we deploy wireless camera nodes either uniformly along each edge of the minimum spanning tree (Uniform), or by using the Steiner tree algorithm (Steiner). When the communication range varies from 0.12 to 0.28, with step size being 0.02, the number of wireless nodes needed to ensure connectivity is shown in Figure 11.

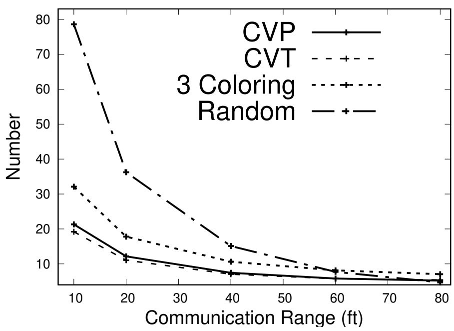


Fig. 10. Number of Relay Nodes using Different Camera Deployment Algorithms

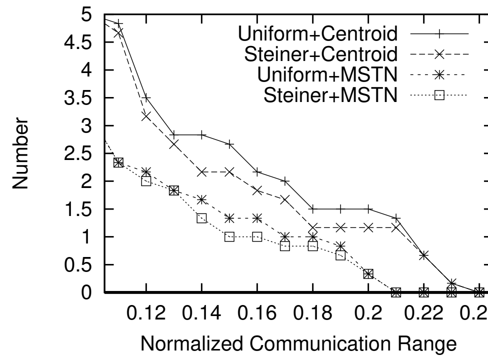


Fig. 11. Performance of Different Relay Node Deployment Algorithms

The MSTN Algorithm. From this figure, we can see that the MSTN algorithm can effectively reduce the number of wireless camera nodes. The improvement of MSTN is more significant when the communication range is smaller. For example, when the communication range is 0.11 units, the MSTN based algorithms require about 2.4 relay nodes, while the Centroid based algorithms require more than 4.6 relay nodes. This is because compared with deploying cameras at the centroids of deployment regions, MSTN algorithm can reduce the total distances of the spanning tree. When the communication range is small, the number of wireless camera nodes is almost proportional to the total distances between the cameras. On the other hand, when the communication range is larger (when $r \leq 0.23$), the number of relay nodes approaches zero.

The Steiner Tree Algorithm. When the communication range is small, the Steiner algorithm achieves the same performance as the Uniform algorithm. This is shown in Figure 11 when communication range is smaller than 0.12. We can see that the values of Uniform+MSTN almost coincide with the values of Steiner+MSTN. This is because under this condition, no wireless nodes are able to connect more than two wireless nodes.

When the communication range is larger, the steiner minimum tree algorithm becomes more effective. For example, when the communication range is 0.14, the Uniform+Centroid algorithm requires 3 wireless nodes, while Steiner+Centroid algorithm requires 2.2 on average. This is because when the communication range is larger, the possibility of finding steiner points that connect more than two components becomes higher.

5.4 Network Energy Consumption

In practical deployment of sensor networks, the battery constraint needs to be considered. The network needs to ensure a sufficient amount of time in order to provide high quality monitoring. Therefore, we adopt the radio model constructed in [26]. To transmit and receive a k bit message over a distance d, the transmission and reception node consumes E_{Tx} and E_{Rx} amount of energy, respectively. These are calculated using Equation 1. The physical meaning of the parameters in this equation are summarized in Table 2. In this simulation, we assume that the wireless cameras are able to transmit over any communication range, at the cost of quadratic growth of power consumption.

$$\begin{aligned} E_{Tx}(k,d) &= E_{Txe}(k) + E_{Tx-amp}(k,d)) \\ &= E_{elec} * k + \epsilon_{amp} * k * d^2 \\ E_{Rx}(k) &= E_{Rxe}(k) = E_{elec} * k. \end{aligned} \tag{1}$$

The camera nodes are modeled after the VIVOTEK CC8130 1MP Panoramic View camera. It operates at frame rate of 10 fps with image resolution at 1280×800. The images are encoded in H.264 format and the video is compressed with a ratio of 30%. Therefore, the data bandwidth of each camera is 254 KBit/s. 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 the energy consumption rate of the network during the data transmission process.

From Figure 12, we can see that by using CVP, CVT and the 3-Coloring algorithm, the networks consume 0.18, 0.23 and 0.34 mW, respectively on communication. One reason why CVP achieves lower communication cost than the 3-Coloring is that CVP algorithm reduces the mutual distances between camera nodes. On the other hand, the 3-Coloring algorithm deploys cameras at the vertices of the polygons, which results in the increase of mutual distances between nodes. Since the wireless power consumption is proportional to the square of link distance, increase in camera nodes' mutual distances will increase the communication power.

Operation	energy cost
Transmitter Electronics (E_{Txe}) Receiver Electronics (E_{Rxe}) ($E_{Tx-elec} = E_{Rx-elec} = E_{elec}$)	$50\ nJ/bit$
Transmit Amplifier (ϵ_{amp})	$100\ pJ/bit/m^2$

Table 2. Radio Characteristics

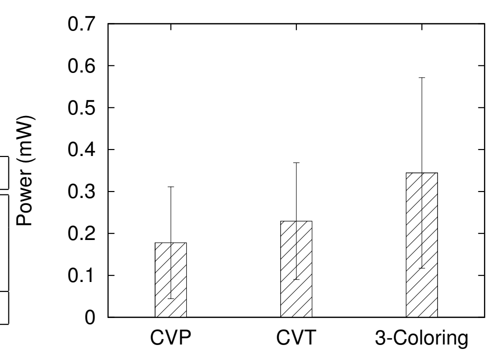


Fig. 12. Communication Power

In summary, the CVP and the CVT outperform the classical 3-Coloring algorithm in terms of camera number, communication power, and number of relay nodes. Specifically, the CVP can achieve a near optimal performance in necessary camera numbers, while the CVT can reduce the number of relay nodes significantly. Both of these algorithms can reduce communication power compared with the 3-Coloring algorithm. When the communication range is small the MSTN algorithm can significantly reduce the number of relay nodes. When the communication range is large, the effectiveness of the steiner minimum tree algorithm begins to be more evident.

6 RELATED WORK

Recent development in camera network deployment research mainly focused on addressing new and different coverage goals [35, 36, 42, 43, 60, 64]. The full-view area target coverage problem in camera sensor networks aims at deploying the cameras in a way such that all the 360° direction of the objects can be monitored [21, 29, 32, 56, 58, 62]. The barrier coverage problem, another important new research direction in camera network deployment, aims at deploying sensors that guarantees the detection when an intruder enters the area through any routes [3, 4, 7, 21, 24, 48, 49, 57, 63]. By modeling the human movements patterns, the authors of [5, 7, 42] developed algorithms to improve the expected coverage ratio while minimizing the number of cameras. However, these works did not address the wireless connectivity issue, which is crucial for the operation of a sensor network because the lack of connectivity means there is no guarantee that the data will arrive at the sink for processing [1, 19]. Besides, while these works address the diverse monitoring requirements for different monitoring scenarios, in certain scenarios, such as the emergency rescuing, battle field monitoring, and natural environment monitoring, full area coverage is still necessary because the events of interest can happen at any locations within the area. Besides, all these papers only focus on visibility coverage, with little attention given to wireless connectivity. In our paper we focus on algorithms that not only ensure the full visual coverage of the area, but also ensure the wireless connectivity of the network.

The joint optimization for both sensing coverage and wireless connectivity has attracted much research attentions in recent years. In particular, the Simulated Anealing algorithm [18], Genetic Algorithm [22, 25, 47], and the local search algorithm [47] are used to developed time-efficient deployment algorithms to ensure both the network coverage and connectivity requirements while reducing the number of sensor nodes. The Integer Programming technique, which has exponential computation complexity, is used to find the optimal deployment of the wireless sensor nodes [47]. In [39], the directional sensing and communication models are used, and the joint optimization of both sensing coverage and connectivity is discussed. However, all these algorithms focus on the disc-shaped or directional sensing models with limited sensing range, which is different from the Line-of-Sight (LoS) sensing model we adopted. The LoS sensing model more accurately describes the performance of a class of sensors whose sensing ranges are limited only by line of sight and have long monitoring distances, such as panoramic cameras, infrared sensors, lidars, and radars, etc. Furthermore, we proposed a 2-approximation algorithm to the minimum geometric guarding network problem with polynomial time complexity.

The Art Gallery problem is a classical problem in computational geometry and is well known to be NP-hard [45]. This problem is about finding the minimum number of guarding points such that any point in the entire polygon is guarded. 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 [14]. In a polygon with n vertices and h holes, $\lfloor (n+h)/3 \rfloor$ point guards are always sufficient [27]. If the polygon is orthogonal (having only horizontal and vertical edges), $\lfloor n/4 \rfloor$ point guards are always sufficient [33]. In [41], the guarded guard set problem was formulated, where each guard in the guard set G must be visible to at least another guard in G, while G can completely ensure visibility coverage of a simple polygon. Different from their work, we focus on the requirement that each guard is wirelessly connected to at leas another guard. These papers provide valuable insights on designing deployment algorithms to ensure visibility coverage, and we adopt their line-of-sight visual sensing model.

The Watchman Route Problem deals with finding a route in a simple polygon P such that each point in the interior of P can be seen from at least one point along the route[52]. In [12], the authors propose an $O(n^4)$ algorithm to solve the problem under the constraint that the watchman route must pass through a starting point s on the boundary of P. To remove the constraint of given starting point, the authors of [44] give an $O(f(n)n^2)$ time solution. The authors of [6] apply 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. In our paper, we utilize the polynomial time solution to the Watchman Route problem as a building block for our CVT algorithm.

7 CONCLUSION

We focus on a wireless camera network deployment problem for indoor monitoring applications, where both physical sensing and cyber networking constraints are imposed by the application requirements. We formally define the *Minimum Connected Guarding Network Problem.* We proved the NP-hardness of the problem and designed a 2-approximation algorithm in the geometric setting. We developed the Connected Visibility Tracking (CVT) algorithm to minimize the deployment time, which is crucial for first responder applications. To further reduce deployment cost, we designed the Connected Visibility Planning (CVP) that utilizes close to minimal number of cameras. We further apply the Minimum Spanning Tree with Neighborhoods (MSTN) and Steiner Minimum Tree algorithms to optimize the network deployment to improve wireless connectivity. Experiments are conducted on an implemented prototype of the proposed system to verify the feasibility for the system. Simulations are conducted to evaluate the performance of the proposed algorithms on realistic floor plans. The results demonstrate that the proposed algorithms can ensure visibility coverage and reduce the camera numbers and the number of relay nodes by up to 28% and 47%, respectively. The communication power consumption is also reduced when our algorithm is applied.

ACKNOWLEDGMENT

This research work is supported by NSF through IIS-1460370, CNS-1463722, 1536086, and 1553273.

REFERENCES

[1] Jamal N Al-Karaki and Amjad Gawanmeh. 2017. The optimal deployment, coverage, and connectivity problems in wireless sensor networks: revisited. *IEEE Access* 5 (2017), 18051–18065.

[2] Yoav Amit, Joseph SB Mitchell, and Eli Packer. 2010. Locating guards for visibility coverage of polygons. *International Journal of Computational Geometry & Applications* 20, 05 (2010), 601–630.

[3] Catalina Aranzazu-Suescun and Mihaela Cardei. 2018. Weak-Barrier Coverage with Adaptive Sensor Rotation. In *International Conference on Combinatorial Optimization and Applications*. Springer, 739–754.

[4] Nguyen Thi My Binh, Huynh Thi Thanh Binh, Vu Le Loi, Vu Trung Nghia, Dang Lam San, and Chu Minh Thang. 2019. An efficient approximate algorithm for achieving (k-!) barrier coverage in camera wireless sensor networks. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, Vol. 11006. International Society for Optics and Photonics, 1100613.

[5] Tisha Brown, Zhonghui Wang, Tong Shan, Feng Wang, and Jianxia Xue. 2017. Obstacle-Aware Wireless Video Sensor Network Deployment for 3D Indoor Monitoring. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 1–6.

[6] S. Carlsson, H. Jonsson, and B. J. Nilsson. 1999. Finding the Shortest Watchman Route in a Simple Polygon. *Discrete & Computational Geometry* 22, 3 (1999), 377–402. https://doi.org/10.1007/PL00009467

[7] Chih-Yung Chang, Chih-Yao Hsiao, Ming Hour Yang, and Sheng-Shih Wang. 2018. Surveillance algorithms for barrier coverage in wireless camera sensor networks. In *2018 International Conference on Electronics Technology (ICET)*. IEEE, 111–115.

[8] Donghui Chen, Ding-Zhu Du, Xiao-Dong Hu, Guo-Hui Lin, Lusheng Wang, and Guoliang Xue. 2000. Approximations for Steiner Trees with Minimum Number of Steiner Points. *Journal of Global Optimization* 18, 1 (2000), 17–33. https://doi.org/10.1023/A:1008384012064

[9] Xiuzhen Cheng, Ding-Zhu Du, Lusheng Wang, and Baogang Xu. 2008. Relay sensor placement in wireless sensor networks. *Wireless Networks* 14, 3 (2008), 347–355.

[10] W Chin and S Ntafos. 1986. Optimum watchman routes. In *2nd Proc. ACM Symposium on Computational Geometry*. ACM Press, New York, New York, USA, 24–33.

[11] W. Chin and S. Ntafos. 1988. Optimum watchman routes. *Inform. Process. Lett.* 28 (1988), 39–44.

[12] Weipang Chin and Simeon Ntafos. 1988. Optimum watchman routes. *Inform. Process. Lett.* 28, 1 (1988), 39–44.

[13] W P Chin and S Ntafos. 1991. Shortest watchman routes in simple polygons. *Discrete & Computational Geometry* 6, 1 (1991), 9–31.

[14] V. Chvátal. 1975. A combinatorial theorem in plane geometry. *J. Combin. Theory Ser. B* 18 (1975), 39–41.

[15] Cregg K. Cowan and Peter D Kovesi. 1988. Automatic sensor placement from vision task requirements. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 10, 3 (1988), 407–416.

[16] Adrian Dumitrescu, Joseph SB Mitchell, and PAWEL ŻYLIŃSKI. 2014. The minimum guarding tree problem. *Discrete Mathematics, Algorithms and Applications* 6, 01 (2014), 1450011.

[17] Adrian Dumitrescu and Csaba D. Tóth. 2012. Watchman tours for polygons with holes. *Computational Geometry* 45, 7 (2012), 326 – 333. https://doi.org/10.1016/j.comgeo.2012.02.001

[18] Yasser El Khamlichi, Abderrahim Tahiri, Anouar Abtoy, Inmaculada Medina-Bulo, and Francisco Palomo-Lozano. 2017. A hybrid algorithm for optimal wireless sensor network deployment with the minimum number of sensor nodes.

Algorithms 10, 3 (2017), 80.

[19] Mohammed Farsi, Mostafa A Elhosseini, Mahmoud Badawy, Hesham Arafat Ali, and Hanaa Zain Eldin. 2019. Deployment Techniques in Wireless Sensor Networks, Coverage and Connectivity: A Survey. *IEEE Access* 7 (2019), 28940–28954.

[20] Steve Fisk. 1978. A short proof of Chvátal's watchman theorem. *Journal of Combinatorial Theory, Series B* 24, 3 (1978), 374.

[21] Xiaofeng Gao, Rui Yang, Fan Wu, Guihai Chen, and Jinguang Zhou. 2017. Optimization of full-view barrier coverage with rotatable camera sensors. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 870–879.

[22] Jobin George and Rajeev Mohan Sharma. 2016. Relay node placement in wireless sensor networks using modified genetic algorithm. In *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*. IEEE, 551–556.

[23] Subir Kumar Ghosh. 2010. Approximation algorithms for art gallery problems in polygons and terrains. In *International Workshop on Algorithms and Computation*. Springer, 21–34.

[24] Ling Guo, Deying Li, Yuqing Zhu, Donghyun Kim, Yi Hong, and Wenping Chen. 2016. Enhancing barrier coverage with β quality of monitoring in wireless camera sensor networks. *Ad Hoc Networks* 51 (2016), 62–79.

[25] Suneet K Gupta, Pratyay Kuila, and Prasanta K Jana. 2016. Genetic algorithm for k-connected relay node placement in wireless sensor networks. In *Proceedings of the second international conference on computer and communication technologies*. Springer, 721–729.

[26] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. 2000. Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*. IEEE, 10–pp.

[27] F. Hoffmann, M. Kaufmann, and K. Kriegel. 1991. The art gallery theorem for polygons with holes. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.* 39–48.

[28] Andrew Howard, Maja J Matarić, and Gaurav S Sukhatme. 2002. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Distributed Autonomous Robotic Systems 5*. Springer, 299–308.

[29] Yang-Pu Hsiao, Kuei-Ping Shih, and Yen-Da Chen. 2017. On full-view area coverage by rotatable cameras in wireless camera sensor networks. In *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 260–265.

[30] Chi-Fu Huang and Yu-Chee Tseng. 2005. The coverage problem in a wireless sensor network. *Mobile Networks and Applications* 10, 4 (2005), 519–528.

[31] Hua Huang, Chien-Chun Ni, Xiaomeng Ban, Jie Gao, Andrew T Schneider, and Shan Lin. 2014. Connected wireless camera network deployment with visibility coverage. In *INFOCOM, 2014 Proceedings IEEE*. IEEE, 1204–1212.

[32] Jinglan Jia, Cailin Dong, Yi Hong, Ling Guo, and Ying Yu. 2019. Maximizing full-view target coverage in camera sensor networks. *Ad Hoc Networks* 94 (2019), 101973.

[33] J. Kahn, M. M. Klawe, and D. Kleitman. 1983. Traditional galleries require fewer watchmen. *SIAM J. Algebraic Discrete Methods* 4 (1983), 194–206.

[34] James King. 2013. Fast vertex guarding for polygons with and without holes. *Computational Geometry* 46, 3 (2013), 219–231.

[35] Guillermo J Laguna and Sourabh Bhattacharya. 2017. Hybrid system for target tracking in triangulation graphs. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 839–844.

[36] Guillermo J Laguna and Sourabh Bhattacharya. 2019. Adaptive target tracking with a mixed team of static and mobile guards: deployment and activation strategies. *Autonomous Robots* (2019), 1–13.

[37] Guo-Hui Lin and Guoliang Xue. 1999. Steiner tree problem with minimum number of Steiner points and bounded edge-length. *Inform. Process. Lett.* 69, 2 (1999), 53–57.

[38] Hengchang Liu, Jingyuan Li, Zhiheng Xie, Shan Lin, Kamin Whitehouse, John A Stankovic, and David Siu. 2010. Automatic and robust breadcrumb system deployment for indoor firefighter applications. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 21–34.

[39] Huadong Ma and Yonghe Liu. 2005. On coverage problems of directional sensor networks. In *International Conference on Mobile Ad-Hoc and Sensor Networks*. Springer, 721–731.

[40] Cristian S. Mata and Joseph S. B. Mitchell. 1995. Approximation algorithms for geometric tour and network design problems (extended abstract). In *Proceedings of the eleventh annual symposium on Computational geometry* (Vancouver, British Columbia, Canada) *(SCG '95)*. ACM, New York, NY, USA, 360–369. https://doi.org/10.1145/220279.220318

[41] T. S. Michael and Val Pinciu. 2003. Art Gallery Theorems for Guarded Guards. *Comput. Geom. Theory Appl.* 26, 3 (Nov. 2003), 247–258. https://doi.org/10.1016/S0925-7721(03)00039-7

[42] Anurag Mittal and Larry S Davis. 2004. Visibility analysis and sensor planning in dynamic environments. In *Computer Vision-ECCV 2004*. Springer, 175–189.

[43] Alan T Murray, Kamyoung Kim, James W Davis, Raghu Machiraju, and Richard Parent. 2007. Coverage optimization to support security monitoring. *Computers, Environment and Urban Systems* 31, 2 (2007), 133–147.

[44] Bengt J Nilsson and Svante Carlsson. 1994. *Guarding art galleries: Methods for mobile guards.*

[45] Joseph O'Rourke. 1987. *Art gallery theorems and algorithms*. Vol. 1092. Oxford University Press Oxford.

[46] Siva Ram, KR Ramakrishnan, PK Atrey, VK Singh, and MS Kankanhalli. 2006. 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, 121–130.

[47] Maher Rebai, Hichem Snoussi, Faicel Hnaien, Lyes Khoukhi, et al. 2015. Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks. *Computers & Operations Research* 59 (2015), 11–21.

[48] Pengju Si, Chengdong Wu, Yunzhou Zhang, Zixi Jia, Peng Ji, and Hao Chu. 2017. Barrier coverage for 3d camera sensor networks. *Sensors* 17, 8 (2017), 1771.

[49] Simone Silvestri and Ken Goss. 2017. MobiBar: An autonomous deployment algorithm for barrier coverage with mobile sensors. *Ad Hoc Networks* 54 (2017), 111–129.

[50] X Tan. 1999. Corrigendum to an incremental algorithm for constructing shortest watchman routes. *Int. J. Comput. Geom. Appl.* 9 (1999), 319–323.

[51] X Tan. 2001. Fast computation of shortest watchman routes in simple polygons. *Inform. Process. Lett.* 77, 1 (2001), 27–33.

[52] Xuehou Tan. 2001. Fast computation of shortest watchman routes in simple polygons. *Inform. Process. Lett.* 77, 1 (2001), 27–33.

[53] X Tan, T Hirata, and Y Inagaki. 1993. An incremental algorithm for constructing shortest watchman routes. *International Journal of Computational Geometry & Applications* 3, 04 (1993), 351–365.

[54] Davi C Tozoni, Pedro J de Rezende, and Cid C de Souza. 2013. The quest for optimal solutions for the art gallery problem: A practical iterative algorithm. In *International Symposium on Experimental Algorithms*. Springer, 320–336.

[55] Jorge Urrutia. 2000. Art gallery and illumination problems. *Handbook of computational geometry* (2000), 973–1027.

[56] Yi Wang and Guohong Cao. 2011. On full-view coverage in camera sensor networks. In *INFOCOM, 2011 Proceedings IEEE*. IEEE, 1781–1789.

[57] Fan Wu, Yang Gui, Zhibo Wang, Xiaofeng Gao, and Guihai Chen. 2016. A survey on barrier coverage with sensors. *Frontiers of Computer Science* 10, 6 (2016), 968–984.

[58] Chao Yang, Weiping Zhu, Jia Liu, Lijun Chen, Daoxu Chen, and Jiannong Cao. 2014. Self-orienting the cameras for maximizing the view-coverage ratio in camera sensor networks. *Pervasive and Mobile Computing* (2014).

[59] Yang Yang, Mingen Lin, Jinhui Xu, and Yulai Xie. 2007. Minimum spanning tree with neighborhoods. In *Algorithmic Aspects in Information and Management*. Springer, 306–316.

[60] Yi Yao, Chung-Hao Chen, Besma Abidi, David Page, Andreas Koschan, and Mongi Abidi. 2008. Sensor planning for automated and persistent object tracking with multiple cameras. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 1–8.

[61] Yi Yao, Chung-Hao Chen, Besma Abidi, David Page, Andreas Koschan, and Mongi Abidi. 2010. Can you see me now? Sensor positioning for automated and persistent surveillance. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 40, 1 (2010), 101–115.

[62] E. Yildiz, K. akkaya, E. Sisikoglu, and M.Y. Sir. 2014. Optimal Camera Placement for Providing Angular Coverage in Wireless Video Sensor Networks. *Computers, IEEE Transactions on* 63, 7 (July 2014), 1812–1825. https://doi.org/10.1109/TC.2013.45

[63] Zuoming Yu, Fan Yang, Jin Teng, Adam C Champion, and Dong Xuan. 2015. Local face-view barrier coverage in camera sensor networks. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 684–692.

[64] Jian Zhao, S Cheung, et al. 2007. Multi-camera surveillance with visual tagging and generic camera placement. In *Distributed Smart Cameras, 2007. ICDSC'07. First ACM/IEEE International Conference on*. IEEE, 259–266.