

Coverage algorithms for visual sensor networks

VIKRAM P. MUNISHWAR and NAEL B. ABU-GHAZALEH, Computer Science, State University of New York at Binghamton

Visual sensor networks are becoming increasingly popular in a number of application domains. A distinguishing characteristic of VSNs is to self-configure to minimize the need for operator control and improve scalability. One of the areas of self-configuration is camera coverage control: how should cameras adjust their field-of-views to cover maximum targets? This is an NP-hard problem. We show that the existing heuristics have a number of weaknesses that influence both their coverage and their overhead. Therefore, we first propose a computationally efficient centralized heuristic that provides near-optimal coverage for small-scale networks. However, it requires significant communication and computation overhead, making it unsuitable for large scale networks. Thus, we develop a distributed algorithm that outperforms the existing distributed algorithm with lower communication overhead, at the cost of coverage accuracy. We show that the proposed heuristics guarantee to cover at least half of the targets covered by the optimal solution. Finally, to gain benefits of both centralized and distributed algorithms, we propose a hierarchical algorithm where cameras are decomposed into neighborhoods that coordinate their coverage using an elected local coordinator. We observe that the hierarchical algorithm provides scalable near-optimal coverage with networking cost significantly less than that of centralized and distributed solutions.

Categories and Subject Descriptors: G.1.6 [Numerical Analysis]: Optimization—*Integer programming*; C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*wireless communication*

General Terms: Algorithms, Design, Performance, Theory

Additional Key Words and Phrases: hierarchical, coverage, optimization

1. INTRODUCTION

Recent technological developments in processing and imaging have created an opportunity for the development of smart camera nodes that can operate autonomously and collaboratively to meet an application's requirements. Visual sensor networks (VSNs), also known as Smart camera networks, have a wide range of applications in areas such as monitoring and surveillance, traffic management and health care [Akyildiz et al. 2007; Soro and Heinzelman 2009]. Wireless camera networks can integrate with an existing fixed infrastructure to substantially improve and scale the coverage and agility of these networks [Collins et al. 2000; Hampapur et al. 2005]. Such systems can also enable *ad hoc surveillance* where a group of wireless cameras are deployed in situations where infrastructure is unavailable or expensive, or quick deployment is desired.

A distinguishing characteristic of VSNs is their ability to self-configure to optimize their operation. One of the primary areas of such self-configuration is the control of camera Field-of-Views (FoV) to optimize the cameras' coverage of targets, events, or

This research work was funded by Qatar National Research Fund (QNRF) under the National Priorities Research Program (NPRP) Grant No.:08-562-1-095

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© YYYY ACM 1550-4859/YYYY/01-ARTA \$10.00

DOI 10.1145/0000000.0000000 <http://doi.acm.org/10.1145/0000000.0000000>

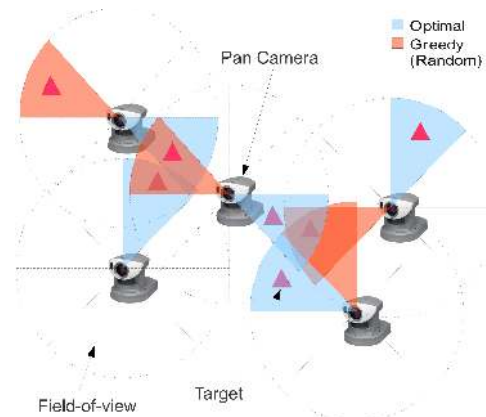


Fig. 1: A general network of wireless Pan cameras. The goal is to maximize the number of targets covered, where each camera can cover only a spherical sector of limited angle (field-of-view). While Optimal FoV configuration can cover 6 targets, the Greedy FoV selection can cover *at most* 5 and *at least* 3 targets, depending on the randomness in the tie-break procedure.

areas of interest. The cameras may be able to change pan, and if available, tilt and zoom to provide the best possible coverage. Since the camera FoVs may overlap with that of the other cameras, coordination with other cameras is needed to optimize coverage. As the scale of cameras grows from tens to hundreds of cameras, it is impractical to rely on human operators to control their setting to achieve the best combined coverage. Thus, supporting autonomous collaborative coverage is a critical problem in smart camera networks.

In this paper, we focus on the problem of target coverage maximization; although generalization to other coverage utility metrics should be straightforward to accommodate. Specifically, we study the problem of maximizing the number of unique targets covered by all cameras; we call this problem as a Camera Coverage Maximization (CCM) problem. Figure 1 presents a general camera network scenario, with multiple pan-only cameras and a set of targets that need to be monitored. In this example, cameras can choose an FoV from 8 discrete FoVs, and their Optimal and Greedy (based on purely local information) FoV selections are shown in the Figure. While Optimal can maximize the coverage by covering 6 out of 7 targets (86% coverage), the Greedy policy covers only 5 targets (71% coverage) in the best case and 3 targets (43% coverage) in the worst case scenario, depending on how the tie-breaks are resolved.

We consider wireless camera networks in our evaluation to further study some of the tradeoffs between delay, messaging overhead and coverage quality in these networks. Although in practice a camera network must track multiple mobile targets in the presence of occlusions, we focus on the problem of coverage of a static set of targets using a number of pre-deployed cameras as the base step in this direction. Later, these base algorithms can be adapted to support target mobility.

Our work is most similar to coverage efforts for directional sensor networks. Most target-oriented coverage techniques in this area assume overprovisioned scenarios where the sensors can cover all the targets of interest [Soto et al. 2009; Cai et al. 2007]. This assumption makes the problem an instance of the set-cover problem, attempting to find the optimal configuration to minimize, for example, the number of sensors needed to cover the targets. In contrast, we consider scenarios where the number of cameras is insufficient to cover the targets, making the problem an instance of the maximum coverage problem [Hochbaum 1996; Ai and Abouzeid 2006]. Impor-

tantly, the above algorithms consider the coverage problem in isolation, and ignore the network-specific aspects completely. In general, communication overhead, especially the delay required to configure all the cameras optimally, can play a significant role in handling dynamism in the scene effectively. Specifically, when the targets are mobile, optimal camera configurations must be computed and communicated to the respective cameras quickly before targets move considerably from their recorded positions.

To provide the context for the problem, we first overview an existing Integer Linear Programming (ILP) formulation of the camera coverage optimization problem developed for directional sensor networks [Ai and Abouzeid 2006]; we modify the existing formulation to account for the different coverage test necessary for camera FoVs. This is an NP-hard problem since the decision version of this problem is based on the classical MAX-COVER [Hochbaum 1996], which is NP-complete. The formulation is presented in Section 2.

Due to the high computational complexity of this problem for large-scale networks, polynomial time heuristics are needed. Our work is similar in spirit to the work by Ai and Abouzeid [Ai and Abouzeid 2006], who propose greedy heuristics for target coverage in directional sensor networks. We show in Section 3 that the existing algorithm leads to inefficient solutions even in simple scenarios, primarily because they only consider the number of targets, and not the likelihood of a camera making pan choices or the redundancy available to cover a target. In response, we propose new algorithms that take into account not only the number of new targets covered by a camera, but the likelihood of a camera selecting a particular FoV when it is making its decision. Thus, in a greedy strategy, our algorithm starts with cameras with more definite choices, significantly reducing the likelihood of making bad decisions early. We also develop a distributed version of the algorithm in Section 4. We show in Section 5 that the proposed centralized and distributed algorithms achieve substantially higher coverage than existing algorithms. In particular, the centralized algorithm tracks optimal in most cases, while the distributed algorithm outperforms even the existing centralized algorithm. We also derive approximation bounds for the algorithms and show that they achieve no worse than 50% of the coverage achieved by the optimal solution.

Overall, the centralized solutions may be attractive to small-scale networks, as they provide high accuracy with acceptable overhead. However, distributed solutions are necessary for large-scales, where networking overheads and solution time are minimized; however they generally lead to a loss in coverage. Moreover, we observed that distributed algorithms can require significant convergence time, primarily due to their iterative nature, which may compromise their agility in highly dynamic scenarios. To address these issues, we propose a hierarchical coverage algorithm where cameras exchange information within their neighborhoods and coordinate on reaching the best configuration jointly. The neighborhoods are created using a clustering algorithm that attempts to find clusters of cameras that have significant overlap in their coverage. We use a tunable parameter to control the size of the clusters in dense scenarios where it is not easy to decouple the cameras. The hierarchical algorithm is discussed in Section 6, and evaluated in Section 7.

Summary of Contributions and Findings: This paper proposes a number of new algorithms for target coverage in smart camera networks. The proposed algorithms achieve better coverage and scalability than the known algorithms that address the same problem. The main contributions of this paper can be summarized as below.

- (1) We propose a new centralized force-directed algorithm (CFA), which outperforms an existing centralized heuristic, and provides nearly optimal coverage for most

of the scenarios we considered. However, being centralized in nature, CFA incurs significantly high communication cost for large scale networks.

- (2) We propose a new distributed force-directed algorithm (DFA) that substantially outperforms the existing distributed algorithm. DFA even outperforms the existing centralized heuristic.
- (3) We develop a lower-bound for both the CFA and the DFA algorithms, showing that they are both 0.5-approximate. We show that this is a tight bound for the existing centralized greedy algorithm (CGA).
- (4) We validate the proposed algorithms using a miniaturized multi-camera testbed.
- (5) Although the distributed algorithms incur considerably less communication cost compared to centralized algorithms, their communication cost increases with the scale and density of the network due to their inherently iterative approach to convergence. To address this problem, we propose a hierarchical algorithm (Hierarchical) to coverage maximization that, by adjusting the tunable parameter to adapt to the network scale, can achieve near-optimal coverage with extremely low communication cost. To create clusters of cameras, we propose a coverage-specific clustering technique.

To evaluate the proposed algorithms, we develop and integrate a camera coverage simulator within a network simulator (QualNet) [Qualnet-Simulator]. Our evaluation shows that the proposed algorithms improve coverage with respect to existing algorithms; the Hierarchical tracks the optimal coverage very closely. More importantly, the main advantage of the Hierarchical comes due to its extremely low communication overhead. This is a critical property for sensor networks for (1) minimizing the power consumption, and (2) handling target mobility effectively by periodically invoking the Hierarchical algorithm. In particular, for a 100 camera deployment scenario, the networking delay to configure the whole network by the hierarchical algorithm was reduced by 97% and by 74% compared to that of the proposed centralized algorithm and the distributed algorithm, respectively. To our best knowledge, this is the first work that takes into account the networking overhead associated with camera coverage maximization algorithms. We present related work in Section 8, followed by concluding remarks and directions for future work in the Section 9.

2. BACKGROUND AND PROBLEM FORMULATION

In this section, we begin with presenting a brief overview of camera's field-of-view and present a coverage test for assessing whether a camera covers a target. We follow with an overview of an ILP based problem formulation of the problem of maximizing the number of targets covered by cameras based on the formulation by Ai and Abouzeid [Ai and Abouzeid 2006]. Finally, we show the impact of different FoV parameters on coverage maximization.

2.1. Field-of-view (FoV) of a camera

Although the work in this paper, considers pan-only cameras, we present an overview of general Pan-Tilt-Zoom (PTZ) camera FoVs. In general, the algorithms and formulation generalize to PTZ cameras after changing the coverage test to account for the FoV of a PTZ camera. An FoV of a camera represents the extent of its visible field. Typically, an FoV is represented by its *angle-of-view* (AoV), α , and *depth-of-field*, R . AoV is represented in terms of the horizontal AoV, α_h , and vertical AoV, α_v , as shown in Figure 2. The depth-of-field represents the area of the visual scene that is acceptably sharp. Thus, for a given target, and a given level of acceptable sharpness, the depth-of-field of a camera spans from R_{min} to R_{max} , and the targets closer to the camera than R_{min} and those farther than R_{max} are not covered by the FoV. When a lens is

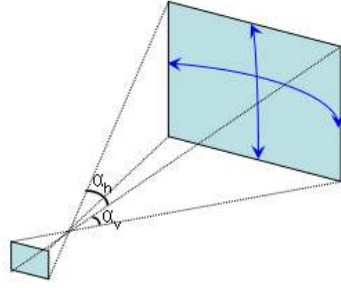


Fig. 2: Horizontal and vertical field-of-view

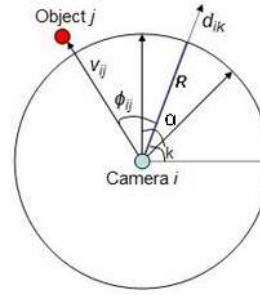


Fig. 3: Camera coverage parameters

focused at a distance that yields maximum depth-of-field, this distance is known as a *hyperfocal distance*. The value of R_{min} is the value of half of the hyperfocal distance, while the value of R_{max} is the maximum distance of a unit size target from the camera, such that the target is acceptably sharp in the image [Adams and Baker 1995]. In order to simplify and abstract away the details, we consider the FoV of a camera as a spherical cone, which can be described by its angle-of-view (α), and depth-of-field (R_{min} and R_{max}). Unlike fixed-FoV cameras, controllable-FoV (PTZ) cameras allow adjusting the FoV in three dimensions: *pan* (horizontal movement), *tilt* (vertical movement), and *zoom* (change in depth-of-field).

2.2. Assumptions

We make the following assumptions in this paper.

- *Camera and Target Localization:* We assume that the locations of cameras are known. Since the cameras are static, this information can be gathered during the deployment time. Otherwise, cameras can be coupled with localizing devices such as GPS. We also assume that the cameras are aware of target locations. Target locations can be obtained through an external localization mechanism, such as using camera calibration [Barton-Sweeney et al. 2006], a low resolution fisheye camera [Miyamoto 1964], stereoscopic localization [Garcia and Solanas 2004], reporting from cooperative targets, or LAsER Detection And Ranging device (LADAR)-based localization [Wang et al. 2003]. This assumption is consistent with those made by other works that assume a network of static and/or wide-angle cameras deployed to inform the locations of the targets to the active (e.g. PTZ) cameras [Schleicher et al. 2006; Kulkarni et al. 2005].
- *Target Coverage:* We assume that targets are covered if a selected point on them is covered; a target is either completely covered or not covered [Papageorgiou et al. 1998]. This point can be the centroid of the target, for example. It is possible to generalize to consider a more continuous measures of coverage, such as different quality metrics [Krahnstoever et al. 2008].
- Targets are static and present along with cameras in an occlusion-free area. However, the model can be generalized to consider occlusions by simply tracking the viewable set of targets. Later, we study some scenarios with target mobility.
- We assume Pan-only cameras. Extension to PTZ cameras is part of our future work.

The assumptions made above define the basic problem. It is possible to relax most of the assumptions. The assumption about point targets can be relaxed, for example by defining a group of points that must be covered for each target. In our testbed experiments, we used red circles as targets, for which we changed the point-target spe-

cific implementation of the coverage test to incorporate targets with given dimensions. Occlusions can also be handled in a number of ways, for example, as part of target localization [Black et al. 2002].

The extension to PTZ cameras is a challenging problem. A primary challenge is the large set of FoV settings possible for each camera, which complicates the single camera assignment and significantly increases the complexity of the multi-camera problem. As part of our future work, we plan to develop an algorithm to detect the optimal set of FoVs for PTZ cameras: most FoVs are either redundant (cover a group of targets identical to other FoVs) or non-maximal (cover a subset of a group targets that are covered by other FoVs). Once the optimal set of FoVs is derived, the multi-camera problem is identical to that for the pan-only camera problem: a set of choices are available at every camera and the optimal assignment of cameras to viable FoVs is attempted by the problem.

2.3. Preliminaries and Notation

For background, we present a formulation of the problem based on the formulation by Ai and Abouzeid [Ai and Abouzeid 2006]. In particular, we modify the objective function to maximize coverage of a preset group of cameras. In Ai et al.'s formulation, the problem considered also seeks to minimize the number of cameras used in the coverage.

The notation used in the formulation is summarized in Figure 3. Consider a set of N static cameras trying to monitor M static targets (or objects) present in an area free of obstacles. Each camera is capable of panning to one of a discrete set of pans, P . The output of the optimization program is the list of $camera \in N, pan \in P$ pairs that collectively maximize the number of $targets \in M$ covered.

Define v_{ij} to be a vector (Euclidean vector) from camera i to object j . Vector d_{ik} denotes the pan direction k of camera i . ϕ_{ij} is a angle between v_{ij} and d_{ik} . α represents the angle-of-view of a camera, defining the width of the FoV in the pan direction. Note that we use horizontal AoV (α_h) of the camera as α , since we consider pan-only cameras in this work. R_{max} represents the maximum length of the field-of-view of camera, while R_{min} represents the minimum FoV length. P represents a set of possible discrete pan values for a camera. In our formulation we assumed that the cameras are homogeneous, but this assumption can be directly relaxed by allowing the camera parameters (α, R_{min}, R_{max} and P) to be defined per camera.

2.4. Coverage Test

The coverage test presented in this subsection is that used by Ai and Abouzeid [Ai and Abouzeid 2006]. The angle between camera and target vectors, ϕ_{ij} , is calculated as:

$$\phi_{ij} = \cos^{-1} \left(\frac{\vec{d}_{ik} \cdot \vec{v}_{ij}}{|\vec{d}_{ik}| |\vec{v}_{ij}|} \right). \quad (1)$$

An object is covered by a camera's FoV when the span of its FoV contains the target and the target is present within the coverable range of the camera. Specifically, the former condition requires the ϕ_{ij} to be less than half of the AoV of camera i , and it can be represented as:

$$\phi_{ij} \leq \frac{\alpha}{2}. \quad (2)$$

The coverage test can be directly adapted to check for tilt constraints by comparing the vertical component of ϕ_{ij} with α_v . The final constraint requires that the distance

between the camera and the target is within R_{min} and R_{max} , and can be represented as:

$$R_{min} \leq |v_{ij}^{\vec{}}| \leq R_{max}. \quad (3)$$

Using the above coverage test, each camera can generate its coverage matrix $A_{N \times P}^M$ such that each element of the matrix, a_{ik}^j , represents whether the camera i can cover object j with pan k .

$$a_{ik}^j = \begin{cases} 1 & \text{if camera } i \text{ with pan } k \text{ covers object } j. \\ 0 & \text{otherwise.} \end{cases}$$

2.5. ILP Formulation of the CCM Problem

The utility (or importance) of the targets is assumed to be equal. Thus, the prime objective is to maximize the overall number of covered targets. A target covered by multiple cameras only counts as one target towards the overall objective.

$$\text{Maximize} \quad \sum_{j \in M} \gamma_j. \quad (4)$$

Where, γ_j is a binary variable that takes value 1 when target j is covered by at least one camera, and 0 otherwise. Coverage is determined by the coverage test described in the previous subsection.

The constraints of the problem can be represented as:

$$\sum_{k \in P} X_{ik} \leq 1 \quad \forall i \in N. \quad (5)$$

$$\frac{\sum_{i \in N, k \in P} a_{ik}^j X_{ik}}{L} \leq \gamma_j \leq \sum_{i \in N, k \in P} a_{ik}^j X_{ik} \quad \forall j \in M. \quad (6)$$

$$X \in \{0, 1\}, \gamma \in \{0, 1\}. \quad (7)$$

Equation 5 represents that a camera can choose only one pan at a time, where X_{ik} is a binary variable which takes value 1 if pan k is selected for camera i . Equation 6 ensures that the utility γ_j for target j can be at most 1, irrespective of the number of cameras covering that target. Here, L is an arbitrary large value ($L \geq \|N\|$). Thus, γ_j can assign itself a value 1 if target j is covered by at least one camera, and 0 otherwise. CCM is an Integer Linear Programming (ILP) problem and it is shown to be NP-hard [Ai and Abouzeid 2006].

Since the CCM problem is NP-hard [Ai and Abouzeid 2006], in Section 3, we present a polynomial-time, near-optimal, heuristic.

2.6. Discussion and Observations

Coverage maximization for camera networks becomes difficult when either: (1) the number of possible FoVs a camera can take increases; and (2) the inter-dependence among cameras (i.e., overlap in the objects they cover) increases. The former condition occurs when the angle-of-view (AoV) decreases, resulting in increased number of non-overlapped discrete FoVs ¹. The latter condition occurs when the depth-of-field (FoV

¹Another way is to consider overlapped discrete FoVs with smaller step size.

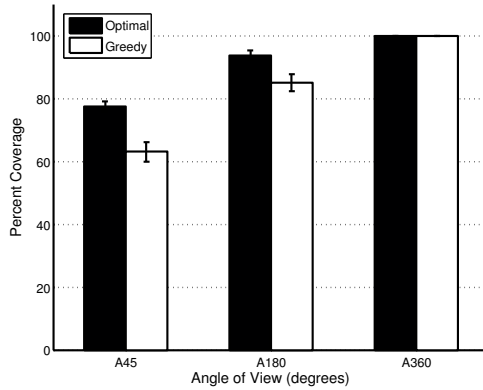


Fig. 4: Impact of different angle-of-views (AoVs)

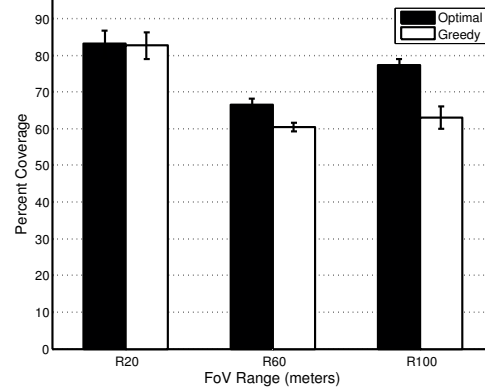


Fig. 5: Impact of different FoV ranges

Range) of cameras is larger, resulting in increased overlap of coverage regions among cameras. Figure 4 and Figure 5 show the impact of different AoVs and FoV ranges on coverage obtained by solving coverage maximization in Optimal and Greedy manner. For this experiment, we consider three AoVs, representing a regular camera (e.g. unzoomed web-cam; AoV: 45°), a fisheye camera (AoV: 180°), and an omni-directional camera (AoV: 360°). We deploy 60 cameras and 100 targets randomly on a 1000×1000 sq. meters terrain. The Y-axis represents *Percent Coverage*, which is calculated as

$$PercentCoverage = \frac{|M_{covered}|}{|M_{potential}|} \quad (8)$$

where, $|M_{covered}|$ are the number of targets covered by all cameras, while $|M_{potential}|$ are the number of *coverable* targets (targets present in the sensing radius) of all cameras. We compare the performance of Optimal against purely local *Greedy* policy, where each camera autonomously assigns a pan that covers highest number of targets out of its coverable targets.

It can be noted from the figures that *as the AoV decreases or FoV range increases, the gap between Optimal and Greedy increases significantly*. Specifically, for AoV of 360° , there is only one feasible FoV a camera can take, and thus Optimal and Greedy perform the same since there are no coverage decisions to be made. As AoV decreases, the number of FoV choices for each camera increases proportionately. As a result, the possibility of choosing a sub-optimal FoV increases, causing an increased gap between Optimal and Greedy. While the AoV affects the number of FoV choices to choose from, FoV range affects the amount of coverage overlaps among cameras. Typically, the degree of overlap among cameras represents the degree of dependency among them for purposes of solving the coverage problem. In other words, if there is no coverage overlap among cameras, then the cameras can select FoVs purely based on their local information (e.g., using a greedy selection), making the greedy assignment optimal. Thus, for 20 meters of FoV range, the degree of overlap among cameras is small, resulting in only a slight gap between Optimal and Greedy. However, as the FoV range increases, Greedy starts performing worse than the Optimal.

Although we do not consider camera zoom in this work, these trends will be more significant for zoom-based cameras: increasing the zoom causes decrease in the AoV as well as increases the FoV range. It also substantially increases the number of FoV

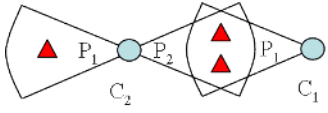


Fig. 6: Dependence of CGA on camera selection order

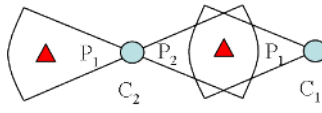


Fig. 7: Dependence of CGA on pan selection order

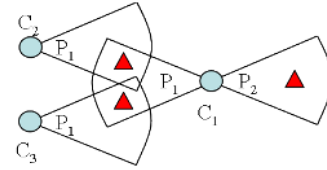


Fig. 8: A counter example for CGA

choices available for each camera. Thus, it is important to explore heuristics that are better than greedy, but that perform more efficiently than optimal.

3. CENTRALIZED FORCE-DIRECTED ALGORITHM (CFA)

Since the optimal problem is NP-hard, it becomes impractical to solve it optimally as the number of cameras and the number of feasible FoVs per camera both increase. Thus, it is necessary to develop heuristics that perform close to optimal. In this section, we first describe the existing state-of-the-art algorithm and analyze some basic cases where it fails to provide optimal configurations. We then present the proposed Centralized Force-based Algorithm (CFA). We also derive an approximation bound for both the existing and the new algorithm.

3.1. Existing solution: Centralized Greedy Algorithm (CGA)

The Centralized Greedy Algorithm (CGA) [Ai and Abouzeid 2006] was proposed in the context of directional sensor networks. CGA begins with making all cameras inactive. In each iteration, it selects the inactive camera that can cover the maximum number of uncovered objects using a single pan direction. Cameras that have been selected already and their covered targets are not considered in successive iterations. The algorithm terminates when all the cameras are assigned a direction.

CGA can lead to a number of basic cases with suboptimal camera-pan configurations. These patterns arise commonly within camera networks.

- *Camera Selection Order Dependence:* CGA is susceptible to the order in which cameras are selected during iterations. Figure 6 shows an example of the scenario, where if camera C_2 is selected before C_1 , then pan P_2 will be assigned to C_2 , leaving no coverable target for C_1 and one uncovered target.
- *Pan Selection Order Dependence:* CGA is also susceptible to the order in which pans are considered for a camera. As shown in Figure 7, if camera C_2 is selected first, it can select either of the pans, since they both cover the equal number of targets. If C_2 selects P_2 , then it results in one uncovered target.
- *Local Greedy Minima:* Even when the camera or pan selection ordering is not a problem, CGA can result in suboptimal operation. Figure 8 shows a case where C_1 with pan P_1 will be selected in the first iteration, since it covers the highest number of targets. This again leaves cameras C_2 and C_3 with no coverable targets and one target remains uncovered.

The above weaknesses can be handled by tracking two metrics:

- (1) *Likelihood of a given pan to be selected:* If a camera has only one feasible pan (a feasible pan is a pan that can cover at least one target) it will surely be selected in the optimal solution. This likelihood decreases as more pans become feasible for a camera. Thus, in the above figures, cameras with the highest likelihood of selecting a pan (cameras with only one feasible pan in these cases) will be selected first, progressing towards the optimal solution.

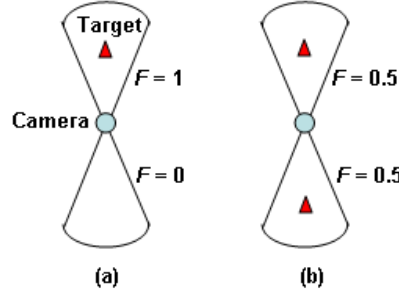


Fig. 9: Illustration of Force-directed Algorithm

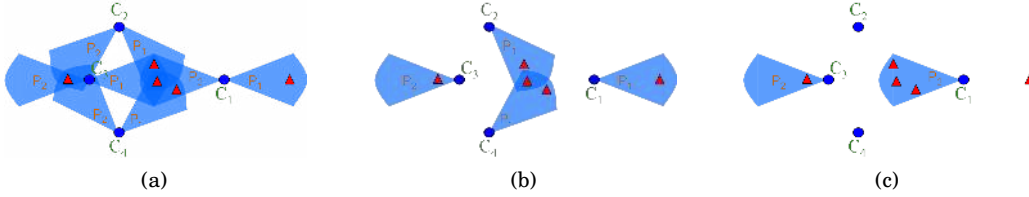


Fig. 10: Counter example for CFA. (a) Scenario where CFA does not produce optimal configurations. (b) Optimal configurations (c) Configurations generated by CFA

- (2) *Coverability of a target:* A target is considered to be difficult to cover if it is *coverable* by fewer cameras, and vice versa. Thus, more preference is given to the targets that are difficult to cover. In the above figures, targets that can be covered by the least number of cameras will be covered first, reaching the optimal solution.

While the first algorithm needs only local information available at a camera, the second algorithm needs the coverage information from all the cameras that can cover a given target. Thus, we use the first algorithm for our proposed algorithm since it is better suited for distributed implementations.

3.2. Centralized Force-directed Algorithm (CFA)

CFA uses a force-based algorithm to determine the likelihood of a given pan to be selected. Essentially, CFA focuses on the number of feasible FoVs a camera can take, and assigns weights to the FoVs depending on the *force* exerted by targets on them. The force in a pan P_i for a camera C_j is computed as the ratio of number of targets coverable by P_i to the total number of targets coverable by the camera C_j . The intuition is that the higher the force, the more certain the camera decision: a camera with all targets in one pan, has no choice but to cover that pan. Consequently, CFA selects a camera-pan pair that has the highest force. Like CGA, when selecting the next best camera-pan pair, CFA removes the already selected camera and its covered targets, and recomputes weights for each FoV of each camera. We now explain the procedure to compute FoV weight, and the working details of CFA.

We consider the objects covered in a particular pan as exerting a force proportional to the number of covered objects in that direction. The force F_{ik} on camera i for selecting pan k is computed as:

$$F_{ik} = \frac{|M_{ik}|}{|M_i|} \quad i \in N, k \in P \quad (9)$$

where $|M_{ik}|$ is the number of targets covered by camera i in pan k , and $|M_i|$ is the number of coverable targets—the targets that camera i can potentially cover (in all possible pans). Figure 9 presents an example of the computation of the force. Essentially, CFA is a greedy solution that iteratively selects camera-pan assignment pairs in decreasing order of their force magnitudes, as presented in Algorithm 1.

ALGORITHM 1: Centralized Force-directed Algorithm

input : $N = \{\text{set of cameras}\}$; $M = \{\text{set of targets}\}$; $P = \{\text{set of discrete pans}\}$
output: Z ; Camera-pan pairs given by CFA.

```

1 while  $N \neq \{\}$  do
2   Compute  $F_{ik}$  for each  $i \in N$  and  $k \in P$ ;
3   Let a camera  $n$  and its pan  $p$  cover  $M'$  targets with force  $F'_{np}$  such that;
4    $F'_{np} = \arg \max_k F_{ik}; \forall i \in N; \forall k \in P; n \in N; p \in P$ ;
5    $Z = Z \cup \{n, p\}$ ;
6    $N = N \setminus n$ ;
7    $M = M \setminus M'$ ;
8 end

```

CFA has three loops: the outer while loop executes N times, and the inner loops execute for each camera-pan pair (i.e. $|N| \times |P|$ times). Thus, the time complexity of CFA is $O(N^2P)$. Given a large network where N is large, the complexity is $O(N^2)$.

CFA handles all three cases we presented earlier where CGA fails to provide optimal coverage. Specifically, in Figure 6 and Figure 7, CFA will first select camera C_1 with pan P_1 , since it has force equal to 1. Similarly, in Figure 8, cameras C_2 and C_3 will be selected first since they both have higher force than that of C_1 .

CFA Counter-case. While our experiments show that CFA gives near-optimal solutions in most cases, since it is a heuristic, there are still possible scenarios where it fails to produce optimal coverage. Figure 10(a) shows one of the scenarios where CFA finds a suboptimal configuration. Optimal can cover all the targets as shown in Figure 10(b). However, CFA loses one target as shown in Figure 10(c). In CFA, camera C_1 first selects pan P_1 as it has the highest value of $F_{1,1} = 0.75$. Subsequently, for the remaining cameras, the value of force along pan P_1 becomes $F_{2,1} = 1, F_{3,1} = 1, F_{4,1} = 1$, while for pan P_2 of the remaining cameras, the force is zero. Thus, any camera can proceed given their equal force magnitudes, and the algorithm will terminate since all cameras have either their pans assigned, or have zero force. Nonetheless, in the experimental evaluation section we discover that in large random deployments of cameras and targets, CFA performs very close to optimal in most cases.

3.3. Lower-bound for the Centralized Greedy Algorithm

CGA and CFA differ primarily in the order of camera selection: CGA selects a camera that covers the most targets, and CFA selects a camera that has the maximum force towards its pan. While CFA may not select a camera that covers the maximum number of targets, it guarantees to select the pan that covers the maximum number of targets compared to the other pans for the selected camera. This can be easily verified based on the notion of force. In this section, we consider a general class of such greedy algorithms, and present a tighter lower-bound for them.

General Greedy Algorithm: Cameras are selected in any arbitrary order. In each iteration, the selected camera covers the maximum coverable targets from the currently uncovered targets. We term this general algorithm as a *Greedy Algorithm* (GA).

Once a camera is selected, both CGA and CFA configure it in a greedy way: they select the pan that covers the maximum number of targets. The algorithms differ only in the order of selection of cameras to configure. Since the general greedy algorithm admits arbitrary order of camera selection, both CGA and CFA represent special cases of the general greedy algorithm. We show that both GA (and therefore both CGA and CFA) have an approximation bound of 0.5.

We use the following notation. Consider a scenario S_0 with N cameras and M coverable targets. Let $m(C_i^g)$ and $m(C_i^o)$ represent the number of targets covered by camera $i \in N$ following the Greedy Algorithm (GA) and the Optimal Algorithm (OA), respectively. Let $M(C_i^o)$ and $M(C_i^g)$ denote the set of targets covered by camera i using OA and GA, respectively. OA covers a total of m^o targets, and GA covers a total m^g targets.

THEOREM 3.1. $m^g \geq \frac{1}{2}m^o$.

We start with an informal outline of the proof. The lower bound is achieved when the first camera selected covers a pan with k targets such that all these targets would have been covered by other cameras in OA. Simultaneously, the camera does not cover another k targets in a different pan such that none of these targets are covered by another camera in OA. No more than k targets can be lost relative to OA since the greedy principle would switch to the uncovered pan if it has more than k targets. While k targets are uncovered, we do cover k targets from the optimal set, leading to the 50% lower bound during this step.

The proof is completed by showing that in the worst case, OA applied to the resulting state after removing the configured camera and its targets covers $m^o - 2k$ targets. While OA could cover more (up to $m^o - k$), greedy in that case would cover more as well and the lower bound is not achieved. Applying the proof iteratively to the remaining state until we configure all cameras or have no remaining coverable targets, we cover 50% of optimal at every step in the worst case, leading to an overall lower bound of 50%. Finally, we show a case where CGA covers 50% – this is a tight bound for CGA. We now formally present the proof.

PROOF. Let GA select a camera C_1 first. In the worst case:

- All k targets covered by C_1 would have been covered by OA applied to the scenario S_0 . If this condition is not true, then greedy covers additional targets not covered by OA, reducing the loss of coverage and achieving a higher bound.
- At the same time, C_1 has k targets in another pan. None of these targets are coverable by another camera in OA. There can be no more than k targets in this other pan of C_1 , or the pan selection would be invalid. If there are less than k targets, greedy will outperform the lower bound (due to the greedy property, its not possible to have more than k target in the alternative pan). Finally, it is sufficient to consider one alternative pan since a different assignment could only cover one pan. If this condition is not true, then these targets are covered by another camera in OA and the loss of coverage is less than 0.5 resulting in a higher bound.

In the first step, in the worst case, GA leads to covering k targets, while OA could have covered $2k$ targets and 50% coverage is achieved. We now consider the network after removing C_1 and the targets it covers under the GA decision to produce scenario S_1 . To decouple the steps, we allow OA to be re-executed on S_1 resulting in an assignment OA_1 , covering m^{o1} targets. Note that in general (i.e., not just in the worst case scenario)

$$m^o - 2k \leq m^{o1} \leq m^o - k \quad (10)$$

The bound comes from observing that we already covered k targets in the first step. If it is possible to cover more than $m^o - k$ targets in S_1 then OA was not optimal.

The lower bound occurs when all targets covered by the GA with C_1 could have been covered by OA_1 , and none of the targets that are not covered in the alternative pan for C_1 can be covered by OA_1 . This represents a lower bound on m^{o1} since no other decision on C_1 could lead to covering more than an additional k targets.

Note that by allowing OA to be reexecuted at every step, the steps are decoupled. The same reasoning can now be applied to S_1 to show that the next camera selected in GA can lead to no worse than 50% of OA_1 .

For deriving the lower bound it is sufficient to consider the lower bound on m^{o1} . Since the steps are decoupled, if more than the lower bound is coverable by OA_1 then the proof shows that no less than 50% of this new and higher bound is coverable by GA. Thus, the overall coverage of GA will be more than 50% of the original OA assignment applied to S_0 .

To get the bound, in the worst case, OA_1 covers $m^o - 2k$. We apply the same reasoning for the next camera selection and do so recursively until all cameras are assigned or no more coverable targets exist. At every step, GA in the worst case achieves 50% coverage of OA, leading to an overall GA approximation bound of 50%. \square

Since GA yields a 0.5-approximate solution, both CGA and CFA, which are special cases of GA, also provide a 0.5-approximate solution to the CCM problem.

We now show the minimal example where CGA achieves exactly half of the coverage of optimal, showing that 0.5 is a tight bound. Consider a scenario with 2 cameras C_1 and C_2 as shown in Figure 7. C_2 has one target t_1 coverable in pan P_1 and another t_2 coverable in pan P_2 . C_1 can cover only t_2 in pan P_1 . The optimal solution covers both targets by configuring C_1 to cover t_2 and C_2 to cover t_1 . However, if greedy picks camera C_2 to assign first, and assigns it to pan P_2 , C_1 has no targets to cover, and only one target is covered. Note that this is our worst case scenario where C_2 picks a pan where the target would have been covered by OA and there exists another pan where the target could not have been covered otherwise by OA. It is interesting to note that CFA achieves optimal assignment for this scenario. We were not able to come up with a scenario where CFA achieves 0.5 coverage, so there is a possibility that there exists a tighter bound than 0.5 for it.

4. DISTRIBUTED COVERAGE MAXIMIZATION PROTOCOL

In this section, we first describe the state-of-the-art distributed algorithm for coverage maximization in the context of directional sensor networks, followed by our proposed distributed algorithm.

4.1. Existing Solution: Distributed Greedy Algorithm (DGA)

DGA [Ai and Abouzeid 2006] begins by assigning a unique random priority value to each sensor. Each sensor detects the total number of targets it can cover in each direction, and selects the direction covering maximum targets. It then sends this direction information to its sensing neighbors (neighbors within $2R$ where R is the sensing range). Overlaps in the target coverage are avoided by accepting the decision of higher priority sensor node. The priority assignment scheme of DGA ensures that the algorithm will terminate. Specifically, the algorithm converges in iterative fashion, stabilizing from the highest priority node in the network.

While the unique priority assignment in DGA ensures its convergence, the random nature in assigning priorities can lead to highly sub-optimal coverage, even compared to the centralized version of the algorithm.

4.2. Distributed Force-based Algorithm (DFA)

DFA can be thought of as a distributed implementation of CFA. It may also be thought of as an extension of DGA where the maximum pan force is used to decide priority among cameras (rather than using arbitrary priority). Specifically, in this algorithm, we use the notion of *force* exerted by targets on selecting a particular pan for a camera as described in Section 3. A camera with higher force for selecting a particular pan will have higher chances of selecting that pan, and thus should get a higher priority. Thus, the force-based priority is essentially the maximum force a camera has from one of its pan. Force-based priority, PR_i , for camera, $i \in N$, can be defined as:

$$PR_i = \arg \max_k F_{ik}; \forall k \in P \quad (11)$$

where, F_{ik} is the force on camera i for selecting pan $k \in P$. Cameras coordinate with their neighbors, similar to DGA, but the decisions of cameras with lower force (priority) will be superseded by that of the cameras with higher force. The DFA algorithm is presented in Algorithm 2.

ALGORITHM 2: Distributed Force-directed Algorithm (runs separately on each camera)

input : $M_i = \{\text{set of coverable targets for camera } i \in N\}$; $P_i = \{\text{set of discrete pans for camera } i \in N\}$ $NB_i = \{\text{set of sensing neighbors of camera } i \in N\}$

output: $p = \text{Best pan chosen for camera } i \in N$.

```

1 Set priority  $PR_i = \arg \max_k F_{ik}; \forall k \in P_i$ ;
2 SetAndAnnounceBestPan ( $M_i$ );
3 while (a protocol message is received from camera  $n \in NB_i$  such that  $PR_n \geq PR_i$ ) do
4   if ( $PR_n = PR_i$ ) then
5     if ( $|M_n| < |M_i|$ ) then
6       continue;
7     end
8     else if ( $|M_n| = |M_i|$ ) then
9       if ( $n < i$ ) then
10        continue;
11       end
12     end
13   end
14   Update  $M'_n$  as the set of covered targets by camera  $n$  in  $NB_i$ ;
15   Update  $M_i$  to exclude targets covered by higher priority neighbors;
16   SetAndAnnounceBestPan ( $M_i$ );
17 end
18 SetAndAnnounceBestPan ( $M_i$ )
19
20 Let  $M_{ik} \in M_i$  represents the total targets covered by camera  $i$  in pan  $k$ ;
21 Set pan  $p \in P_i$  such that  $M_{ip} = \arg \max_{k \in P_i} M_{ik}$ ;
22 if (Pan  $p$  is different than the previously selected pan) then
23   Send a protocol message including priority and a set of covered targets to cameras
24    $\in NB_i$ ;
25 end

```

Tie-breaking procedure. The DGA with random priority scheme ensures that the neighboring cameras receive unique priorities. However, the force-based priority in DFA may not adhere to the unique priority requirement. Thus, in DFA, we allow a camera to accept packets from a neighbor even if they both have the same priority, and apply tie-breaking mechanisms to decide whether to accept or discard the packet (lines 4-13). Specifically, if both the cameras have the same priority, the camera that covers a higher number of targets than the other camera wins. If they both cover the same number of targets, then we break the tie by favoring a camera having a higher ID number.

Note that since the operation of DFA is similar to that of DGA, DFA is guaranteed to terminate [Ai and Abouzeid 2006]. Furthermore, DFA does not require any explicit synchronization among cameras, since the priorities of cameras do not change over time.

4.3. Analysis of DFA

In this section, we present an analysis of DFA with respect to the optimal policy.

THEOREM 4.1. *DFA (and DGA) provides a 0.5-approximate solution to the CCM problem.*

PROOF. In DFA, a camera with the highest priority within its neighborhood is configured first. The neighboring cameras then ignore the targets already covered by the higher priority camera, and configure themselves with the pan that covers the maximum number of the remaining uncovered targets.

If there are two cameras with the same highest priority in a neighborhood, the tie-breaking procedure guarantees that one of them will be configured first (provided that the camera ids are unique).

Thus, although DFA does not guarantee the camera selection order, it guarantees that the selected camera selects the pan that covers the maximum number of targets, and the remaining cameras ignore the already covered targets. Thus, the lower-bound for the general greedy algorithm (GA) applies to DFA (and DGA).

Thus, DFA (and DGA) provides a 0.5-approximate solution to the CCM problem. \square

5. PERFORMANCE EVALUATION OF CENTRALIZED AND DISTRIBUTED ALGORITHMS

We first demonstrate that the proposed centralized and distributed algorithms work in miniaturized testbed environment. We then consider different scales and distributions of targets in simulation, and study their impact on coverage.

5.1. Methodology

In this section, we describe the experimental setups used for the evaluations of the proposed algorithms. We also present the metrics of evaluation.

Policies Considered for Evaluation: We express the ILP formulation using AMPL-CPLEX to produce the optimal solution [ILOG 2001]. We compare the proposed policies *CFA* and *DFA* with the *Optimal* (by solving the ILP). We also evaluate the existing centralized (*CGA*) and distributed (*DGA*) algorithms [Ai and Abouzeid 2006] as representative of the state of the art. Finally, we use as a baseline the pure *Greedy* algorithm where each camera simply points to the pan with the most targets.

Miniaturized Testbed: The purpose of the testbed evaluation is to (1) validate the accuracy of FoV model that is used to generate optimal camera-pan pairs based on the location information of cameras and targets; and (2) to establish the presence of one of the base patterns (camera-selection order dependence) that leads to suboptimal coverage by the existing algorithms (*CGA* and *DGA*). The testbed includes Axis 213 cameras [axis-213] integrated with the OpenCV library which provides implementa-

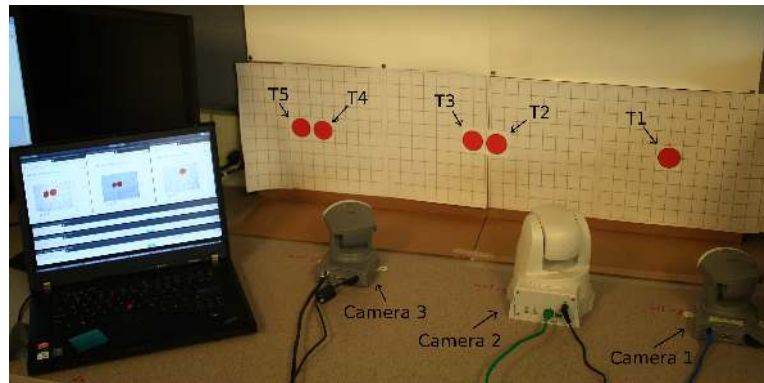


Fig. 11: Miniaturized camera network testbed.

tions of common vision algorithms [G. Bradski et al. 2000], and FFmpeg to capture video streams from the cameras [Bellard et al. 2007]. We abstract a target into a circle (or dot) primarily to miniaturize the testbed space to efficiently perform the testbed specific operations. Figure 11 shows the miniaturized camera testbed with three cameras. As seen from the testbed, the pan movement of cameras is limited. Thus, we use 30° as a step-size for discrete pans, resulting in total 12 different pans per camera. Moreover, we assume that the locations of cameras and targets can be supplied by an external localization system—we obtain the coordinates of cameras and targets manually. The testbed-based evaluation is presented in Section 5.2.

Coverage and Networking Simulator: The limitation of testbed-based scenarios is that it is hard to setup experiments with large scale targets forming different spatial distributions. To address this difficulty, we have developed and integrated a camera coverage simulator within a network simulator (QualNet) [Qualnet-Simulator]. Nodes are categorized into cameras and targets, and are assigned respective functionalities. For most experiments, each camera is set with a pre-defined maximum length of the FoV $R_{max} = 100$, minimum length of FoV $R_{min} = 0$, and FoV angle $\alpha = 45^\circ$ ². The camera can choose from 8 discrete pan values as its orientation, however the proposed algorithms can use higher number of pans, which could be overlapping. We use a fixed terrain of dimensions 1000×1000 sq. meters, where we place cameras and targets. We study the impact of density by increasing the number of cameras and targets rather than changing the size of the area. The rest of the evaluations are performed using the coverage and networking simulator.

Metrics: We track primarily the following three metrics: (1) *Percent coverage*: is the ratio of total covered targets to the total coverable (with some feasible pan setting) targets. In many scenarios some targets are located at positions that cannot be covered by any cameras—such targets are omitted from consideration when computing coverage; (2) *Delay*, is the total delay required for configuring all the cameras; and (3) *Messaging overhead* is the total number of packets communicated within the network as a part of protocol execution. This overhead is captured at the Medium Access Control (MAC) layer to consider MAC-level retransmissions, as well.

5.2. Testbed Validation

In this section, we validate and demonstrate the effectiveness of the proposed algorithms in a miniaturized visual sensor system. Figure 12 shows the different FoVs

²However, full camera FoV functionalities based on the Axis-213 camera is supported in our model; the model is available for distribution from the authors.

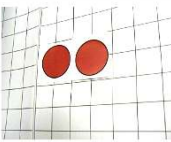
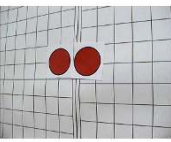
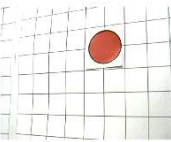


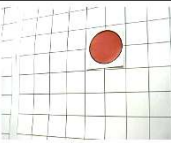

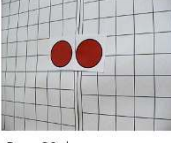



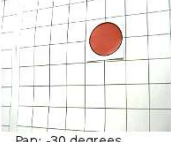
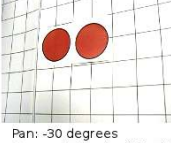
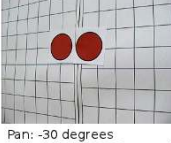
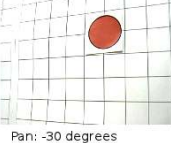
	Camera 3	Camera 2	Camera 1
Optimal	 Pan: -30 degrees Covered Targets: {T4, T5}	 Pan: -30 degrees Covered Targets: {T2, T3}	 Pan: -30 degrees Covered Targets: {T1}
CGA	 Pan: 30 degrees Covered Targets: {T4, T5}	 Pan: 0 degrees Covered Targets: {}	 Pan: -30 degrees Covered Targets: {T1}
CFA	 Pan: -30 degrees Covered Targets: {T4, T5}	 Pan: -30 degrees Covered Targets: {T2, T3}	 Pan: -30 degrees Covered Targets: {T1}
DGA	 Pan: 30 degrees Covered Targets: {T4, T5}	 Pan: 0 degrees Covered Targets: {}	 Pan: -30 degrees Covered Targets: {T1}
DFA	 Pan: -30 degrees Covered Targets: {T4, T5}	 Pan: -30 degrees Covered Targets: {T2, T3}	 Pan: -30 degrees Covered Targets: {T1}

Fig. 12: Evaluation of the centralized and distributed policies.

(pans) selected by cameras using the centralized and distributed algorithms, when the cameras and targets were distributed as shown in Figure 11. The straight orientation (pan) of a camera is assumed to be 0° . In this scenario, camera 1 can cover target {T1} in pan -30° . Camera 2 (C_2) can cover targets {T2, T3} in pan -30° , and target {T1} in pan 30° . Finally, Camera 3 (C_3) can cover targets {T4, T5} in pan -30° , and targets {T2, T3} in pan 30° .

CGA: Since both C_2 and C_3 can cover the same maximum number of targets, it falls into the *Camera Selection Order Dependence* case, which is described in Section 3.1. Since C_3 is selected first in this case, CGA results in 60% coverage of Optimal.

CFA: Since C_1 has only one feasible pan choice, it is guaranteed to be a part of the optimal solution. After C_1 is selected, C_2 is left with only one feasible pan, which must be a part of the optimal solution. Similarly, after C_1 and C_2 , C_3 has only one feasible pan, resulting in the exact optimal solution. However, as we show in Section 3, cases where CFA does not provide optimal solutions exist.

DGA: In DGA, the ID of a camera is assigned as a priority of that camera. Since the priority assignment is random, the wrong pan choice of a higher priority camera

results in loss of coverage. This happens with C_3 resulting in C_2 having no feasible pan that covers at least one target, and thus 60% of the overall coverage is achieved. Table I shows the operation of DGA in step-by-step manner. We exclude the steps that involve message transfer from lower- to the higher-priority camera, since such messages are discarded.

Steps	Cam 3	Cam 2	Cam 1	Message Received
Priority	3	2	1	N/A
Initial (Pans)	30°	-30°	-30°	
Targets Covered	$\{T_2, T_3\}$	$\{T_2, T_3\}$	$\{T_1\}$	N/A
Step 1 (Pans)	30°	-30°	-30°	
Targets Covered	$\{T_2, T_3\}$	$\{T_2, T_3\}$	$\{T_1\}$	From Cam2 to Cam1
Step 2 (Pans)	30°	-30°	-30°	
Targets Covered	$\{T_2, T_3\}$	$\{T_2, T_3\}$	$\{T_1\}$	From Cam3 to Cam1
Step 3 (Pans)	30°	0°	-30°	
Targets Covered	$\{T_2, T_3\}$	$\{\}$	$\{T_1\}$	From Cam3 to Cam2

Table I: Steps of DGA. Priorities of cameras are assumed to be the camera Ids. Since C_3 has the highest priority, and it makes a wrong decision initially, DGA ends up covering only 3 targets.

DFA: In DFA, maximum force value for a camera is considered to be its priority. Thus, DFA converges in the order of C_1 , C_2 , and finally C_3 , resulting in the 100% coverage. Table I shows the operation of DGA in step-by-step manner.

Steps	Cam 3	Cam 2	Cam 1	Message Received
Priority	0.5	0.67	1	N/A
Initial (Pans)	30°	-30°	-30°	
Targets Covered	$\{T_2, T_3\}$	$\{T_2, T_3\}$	$\{T_1\}$	N/A
Step 1 (Pans)	30°	-30°	-30°	
Targets Covered	$\{T_2, T_3\}$	$\{T_2, T_3\}$	$\{T_1\}$	From C_2 to C_3
Step 2 (Pans)	-30°	-30°	-30°	
Targets Covered	$\{T_4, T_5\}$	$\{T_2, T_3\}$	$\{T_1\}$	From C_1 to C_2
Step 3 (Pans)	-30°	-30°	-30°	
Targets Covered	$\{T_4, T_5\}$	$\{T_2, T_3\}$	$\{T_1\}$	From C_1 to C_3

Table II: Steps of DFA. Priorities of cameras are computed based on Equation 11. DFA covers all 5 targets.

Since it is hard to evaluate the algorithms for large scales and different distributions of cameras and targets, we perform our remaining evaluations with the integrated coverage and network simulator.

5.3. Coverage Opportunity: Gap between optimal and greedy

In the simulation based environment, we first study how the densities of cameras and targets affect the optimal coverage. This study allows us to understand the expected coverage gain from deploying smart algorithms (vs. greedy) under a range of scenarios starting from those that are underprovisioned (number of cameras insufficient to cover the large number of targets) to those that are well provisioned or even overprovisioned.

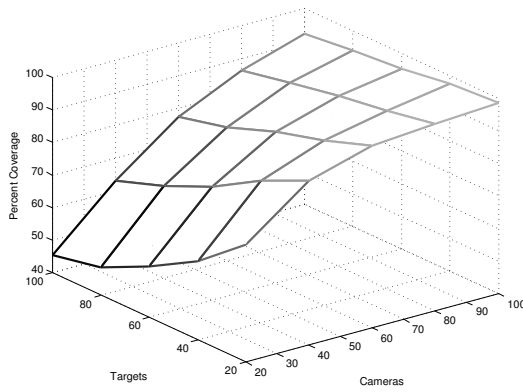


Fig. 13: Impact on total targets covered

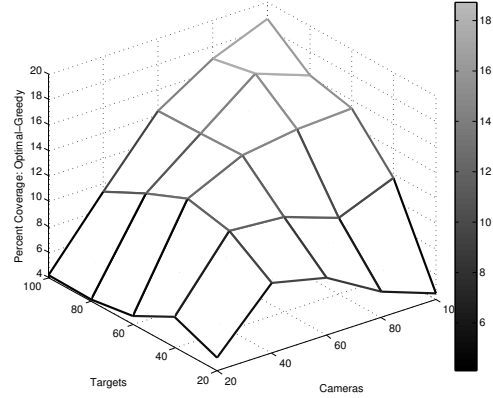


Fig. 14: Gap between Optimal and Greedy

Figure 13 shows the impact of varying camera and target densities on the targets covered. Figure 14 illustrates the difference of the coverage obtained by Optimal and Greedy policies to highlight the region of operation where intelligent coverage algorithms are needed. For large numbers of targets, the difference between optimal and greedy increases as the camera density increases. This is because increasing the camera density means packing more cameras in the deployment region. This causes higher FoV overlaps and thus increases overall dependencies among cameras. Since each camera operates independent of others in the greedy policy, the cameras may cover more redundant targets, resulting in increased loss of coverage.

Similarly, Figure 13 shows that the percentage of targets covered decreases as the target density increases, due to the increase in the relative number of targets present in the uncovered region of the FoV. It can also be observed that for very high densities of cameras (i.e. 80 or 100 cameras), the coverage is almost independent of the targets density. Thus, we choose the next maximum number of cameras (60 cameras) for evaluations with varying target density. Similarly, for 100 total coverable targets, the coverage shows highest variation with varying number of cameras, thus we choose 100 targets for the evaluations with varying camera density.

5.4. Impact of Camera Scale and Density

In this study, we start the evaluation of the proposed algorithms. Like the base study above, we vary the camera density to analyze performance under a range of operating conditions. The number of targets is set to 100, and cameras are deployed with uniform distribution. The centralized algorithms, Optimal, CGA, and CFA work with global information, where each camera sends its location and coverable targets locations to a central location (base-station), which computes the best possible camera-pan pairs and distributes the pan values back to the respective cameras. We use TCP to ensure 100% reliability for the communication of control packets; however, we discover that TCP introduces high delays under large scenarios, inviting a custom solution for reliable collection of the camera state information.

5.4.1. Coverage Accuracy. As can be noted in Figure 15, CFA outperforms CGA and in fact tracks the Optimal very closely. CFA deviates slightly from the optimal for higher camera densities due to the existence of the counter-example cases for the CFA, as described in Section 3. However, since such cases are relatively rare, CFA provides near-optimal performance in general. Similarly, the DFA significantly outperforms the DGA, reducing the loss of coverage w.r.t. the Optimal by about 75% on an average. It

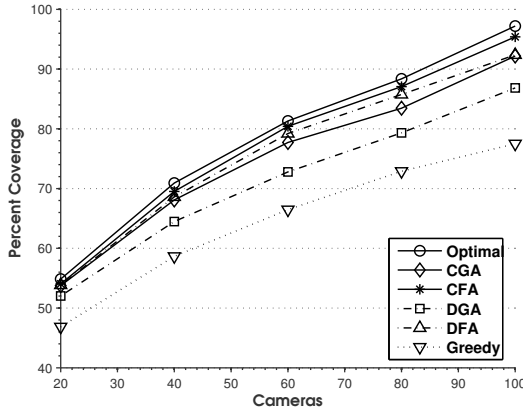


Fig. 15: Impact of camera density on coverage.

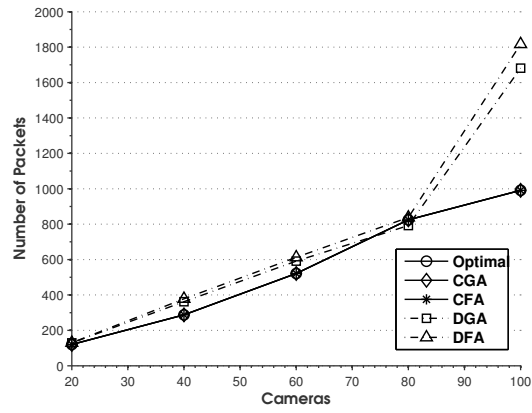


Fig. 16: Impact of camera density on messaging overhead.

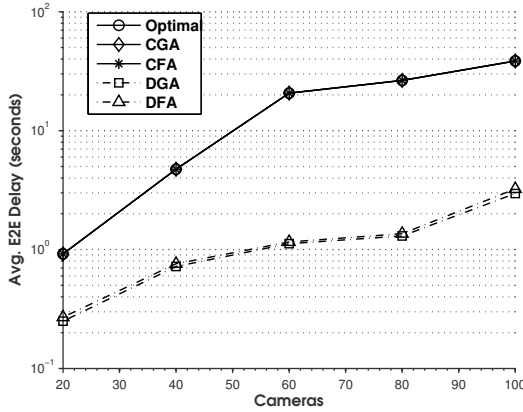


Fig. 17: Impact of camera density on end-to-end delay.

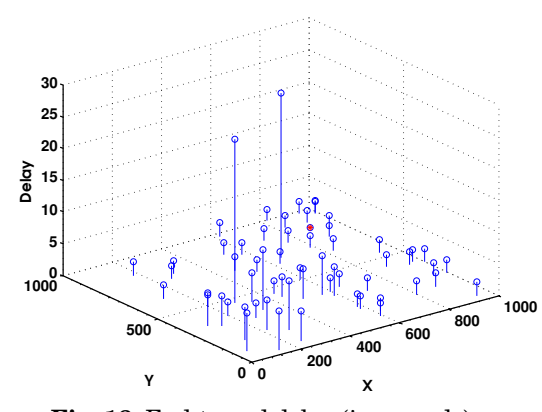


Fig. 18: End-to-end delay (in seconds) per camera for centralized policies.

is more interesting to note that the proposed distributed algorithm, DFA, even outperforms the existing centralized algorithm, CGA.

5.4.2. Delay and Messaging Overheads. Figure 16 shows the impact on messaging overhead, while Figure 17 shows the impact on the end-to-end delay. To capture the overall time for the whole network to get configured, we use worst-case end-to-end delay measurements, which represent the delay between the first camera sending its status packet to the base-station and the last camera configuring itself successfully, using the configuration sent by the base-station. The Y-axis represents an average of end-to-end delays and packet counts taken over 10 different seeds for the experiment.

As it can be noted from the delay measurements, the delay for centralized policies increases drastically as the number of cameras increases, since the overall response time is severely affected even if the data for one camera is delayed to/from the base station. Specifically, contention due to many-to-one (from nodes to base-station) and one-to-many flows (from base-station to nodes) in the network causes packet loss, which results in multiplicative backoffs. Particularly, the base-station suffers the most, since it handles multiple TCP connections simultaneously [Balakrishnan et al. 1998]. We

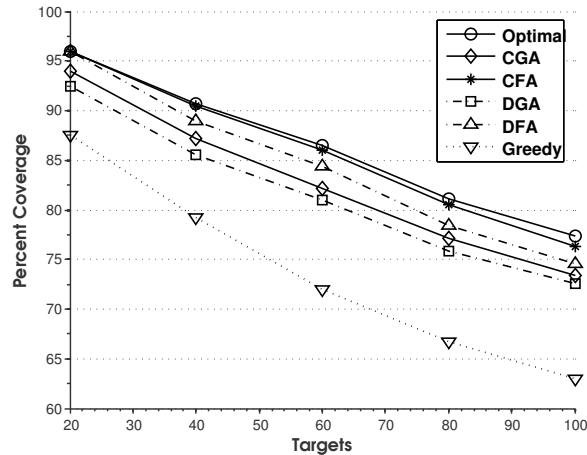


Fig. 19: Impact of varying target density on coverage.

Targets	Avg. E2E Delay		Number of Packets	
	DGA	DFA	DGA	DFA
20	0.52	0.53	277.40	282.10
40	0.70	0.73	403.80	416.30
60	0.82	0.88	488.60	510.10
80	0.88	0.93	518.00	544.70
100	0.90	0.95	536.80	562.80

Table III: Avg. end-to-end delay (in seconds) and number of packets for DGA and DFA. Other policies are omitted because their overhead is independent of target density.

also observe destructive interactions with the TCP congestion management algorithm where packet losses cause exponential backoff of the retransmit timer adding to the overall delay.

Figure 18 shows per camera end-to-end delay (on Z-axis) for running the Optimal policy. The X- and Y-axis represent X and Y coordinates of cameras on the terrain, respectively, and the small circles represent cameras deployed on the terrain. The shaded circle shows the location of the base-station (randomly placed) for this scenario. As it can be noted, the delays for only two cameras are order of magnitude higher than the remaining cameras. Since there is no traffic other than the control packets, these delays are the artifacts of TCP's congestion control mechanism.

The distributed algorithms incur relatively less convergence delay, but even they are susceptible to delays, as the network scale and density increase. This delay is due to their inherently iterative nature—they converge from the highest to the lowest priority cameras, in general.

In terms of messaging overhead, for lower camera density, both the centralized and distributed algorithms transmit the same number of packets, while for the increased camera density, packet overhead increases due to the increased medium access contention and packet loss in the network. The high messaging overhead of the distributed algorithms for high camera density is due to the increase in the number of neighboring cameras for each camera.

5.5. Impact of Target Scale and Density

For this study, we deploy 60 cameras and varying number of targets in random fashion. As shown in Figure 19, CFA and DFA provide close-to-optimal results irrespective of the target scale and density, with even DFA outperforming the existing centralized algorithm, CGA. Next, we found that while the overhead of centralized algorithms and the Greedy algorithm are oblivious to the target scale and density, the overhead of distributed algorithms are correlated with it. In general, the end-to-end delay and number of packets increase monotonically with the density of targets, as depicted in Table III. DFA has slightly more overhead than DGA, since increasing the target density increases the number of feasible pans a camera can take, where a pan is feasible if it covers at least one target. Thus, upon preemption by a higher priority neighboring camera, a camera can choose from large number of available pan options, resulting in a slightly delayed convergence.

6. ADDRESSING THE SCALABILITY PROBLEM: HIERARCHICAL ALGORITHM

While centralized algorithms work well for small sized networks, they scale poorly for large-sized, dense networks due to the extremely high communication overhead. In particular, the delay increases quickly with the scale and density of the network. This can be observed in Figure 17, where the end-to-end delay for 20 cameras is around 1 seconds, and increases quickly (in 10's of seconds) for camera numbers greater than 40. On the other hand, the distributed solutions have relatively less overhead. However, as the network density increases, they fail to provide effective coverage and their overhead also increases due to their iterative nature.

Overall, it is important to keep delays low, particularly when camera configurations need to be adjusted periodically depending on target mobility, in order to maintain the desired coverage accuracy. Similarly, lower messaging overheads have significant impact on improving network lifetime in battery-operated networks.

In response to these concerns, we propose a hierarchical algorithm (Hierarchical), which addresses the scalability problem by spatially decomposing a camera network into multiple neighborhoods and running centralized algorithms within each neighborhood. This does not hamper the coverage accuracy significantly if the dependencies among cameras are exploited appropriately. Specifically, it is crucial to create the neighborhoods such that the camera dependency within a neighborhood is much higher compared to that of across the neighborhoods.

The dependency among cameras can vary based on aspects such as: (1) *Distance among cameras*: the more separated the cameras are, the less likely they are to have overlap in terms of coverage; (2) *Pan bound*: the limited pan movement, possibly arising from the mechanical limits or the mounting place (e.g. wall), can decrease the possible overlaps among cameras; and (3) *Occlusions*: though we do not consider occlusions in this work, they can also cause independence among cameras by reducing the possible coverable area for a camera. However, in some scenarios, particularly in a highly dense camera network, it may not be possible to find considerably independent neighborhoods. Thus, we provide a parameter, maximum allowed cluster size (S_{max}), that limits the size of the neighborhood such that the communication overhead is kept below the limit.

6.1. Modified Single-Linkage Algorithm (SLA)

In order to find least dependent neighborhoods (or clusters) of cameras, we use an existing hierarchical clustering technique [Johnson 1967], Single-Linkage Clustering [Sibson 1973], where distance between two cluster is considered as the minimum distance between two points from the two clusters. Although clustering is a heavily

studied topic in sensor networks, to the best of our knowledge, this is the first attempt to apply clustering to balance the overhead (computational and communication) of centralized coverage algorithms against the lack of accuracy of distributed solutions that results from the limited amount of local information they act on.

The modifications to the SLA algorithm specific to the camera-coverage problem are as follows:

- (1) *Termination condition for cluster merging*: Single-linkage clustering algorithm is a bottom-up clustering algorithm, which starts with each camera as a separate cluster, and merges clusters ‘close-enough’ with each other into a single cluster at different levels of the hierarchy. We employ a limit on the merging operation if the smallest distance between two clusters is greater than $2R_{max}$, where R_{max} is maximum sensing radius of a camera, since decisions of such clusters will be completely independent of each other.
- (2) *Maximum Cluster Size (S_{max})*: It is an important parameter that can be tuned to adjust computational and communication load of the network. In Hierarchical, a camera node may act as a cluster-head to compute optimal pans for its cluster-members. Since a centralized algorithm is run within each cluster, the communication overhead is directly proportional to the cluster size. Thus, S_{max} can be adjusted depending on the desired coverage accuracy and overhead levels, and the cluster-merging operation is terminated if the resultant size of the merged cluster will be greater than S_{max} .

Working of the modified SLA: The modified Single-Linkage Algorithm (SLA) is presented in Algorithm 3. Essentially, we use the cluster formation mechanism of SLA, but restrict combining two coverage independent clusters into one cluster, and limit the maximum allowed cluster size to ensure low-latency communication. During each iteration, the *while* loop increments d (line 22), which represents the maximum allowed distance between two cluster members. For a given value of d , a pair of clusters is merged if the combined size of both the clusters is less than S_{max} and the minimum distance between the clusters is at most d (lines 7-17). If at least one merged-cluster is created, it is added to a new level in the hierarchy (lines 18-21). Once the clusters are formed, the cluster-member that is at a minimum hop distance away from the rest of the cluster members is chosen as the cluster head. More details about the working of SLA and hierarchical clustering in general can be found in the literature [Johnson 1967; Sibson 1973].

Using the clusters provided by the modified SLA algorithm, optimal camera configurations are computed for individual clusters, where each cluster-member sends a request to its cluster-head, and the cluster-head computes and sends the optimal camera configurations back to the members. Note that the main focus here is to show the effectiveness of Hierarchical when S_{max} is set appropriately. However determining the value of S_{max} depending on the underlying camera topology and target distributions is a significant work in itself, and thus is a part of our future work.

Hierarchical may produce sub-optimal results when the cameras on the boundaries of different clusters end up covering the same targets due to the computations performed for individual clusters. However, in most of our experimental evaluation, Hierarchical performed close to optimal. Thus, we decided to forgo a coordination mechanism to correct decisions between boundary cameras. It is possible that coordination between border cameras can improve performance in some scenarios where the clustering does not successfully discover clear separation between neighboring clusters.

ALGORITHM 3: Modified Single Linkage Algorithm

input : A = Adjacency matrix of Euclidean distances among cameras; S_{max} = Maximum cluster size; R = Coverage range of a camera
output: Z ; Set of final clusters

```

1  $\ell = 0$ ; Current level of hierarchy;
2  $d$  = Bound on distance between two cluster members;
3  $\Delta$  = Distance increment;
4  $Clusters_\ell$  = List of clusters. Each camera is an individual cluster initially;
5 while  $d \leq 2R$  do
6    $NewClusters = \{\}$ ;
7   foreach  $C_i \in Clusters_\ell$  do
8      $NewC = \{\}$ ;
9     foreach  $C_j \in Clusters_\ell$ , where  $j \geq i$  do
10      if  $\|C_i\| + \|C_j\| \leq S_{max}$  and  $MinDist(C_i, C_j) \leq d$  then
11         $NewC = NewC \cup C_j$ ;
12      end
13    end
14    if  $\|NewC\| > 0$  then
15       $add(NewClusters, NewC)$ ;
16    end
17  end
18  if  $\|NewClusters\| \neq \|Clusters_\ell\|$  then
19     $Clusters_{\ell+1} = NewClusters$ ;
20     $\ell = \ell + 1$ ;
21  end
22   $d = d + \Delta$ ;
23 end
24  $Z = Clusters_\ell$ ;

```

6.2. Discussion

The motivation behind hierarchical coverage is primarily the scalability, and not optimality. Hierarchical provides optimal coverage within each cluster, since it runs the optimal policy inside each cluster. However, Hierarchical operates in a greedy manner across clusters, since clusters do not collaborate with each other. Thus, if all clusters are coverage-independent of each other, Hierarchical provides globally optimal solution.

When two clusters are not independent, suboptimality can originate only from the decision of the cameras on borders of those clusters that have a common set of coverable targets. Since the border cameras are configured independently of each other, they may end up selecting the same group of targets. In other words, if those border cameras are covering different sets of targets, the solution remains optimal. Thus the suboptimality is present only when the border cameras cover some redundant targets. The maximum deviation from the optimal policy is the amount of redundant targets covered by the border cameras. This is because the border cameras could have potentially covered some other uncovered targets. The number of these uncovered targets must be less than the number of redundant targets, since optimality is guaranteed within a cluster.

In summary, clusters in the Hierarchical policy operate independently. Thus Hierarchical may perform as badly as a pure greedy policy in the worst case, if all cameras from all the clusters are border cameras. However, such worst-case scenarios are extremely difficult to occur in practice, when appropriate value for S_{max} is used.

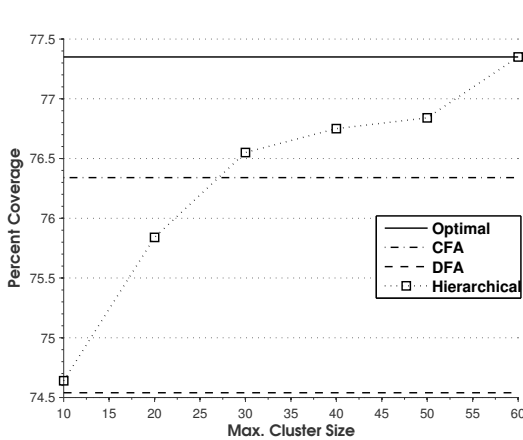


Fig. 20: Impact of S_{max} on coverage

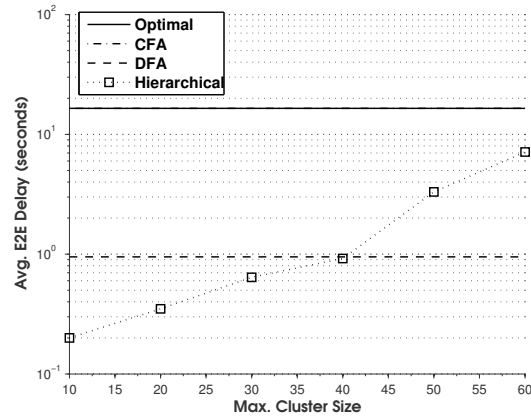


Fig. 21: Impact of S_{max} on end-to-end delay (Optimal and CFA are overlapping.)

7. EVALUATION: HIERARCHICAL ALGORITHM

In this section, we present evaluation of the hierarchical algorithm with respect to the proposed centralized (Optimal, CFA) and distributed (DFA) algorithms for varying camera and target density. We do not consider the other policies since the force-directed algorithms provide better coverage with equal overheads. The goal of this section is to evaluate (1) tunability of S_{max} and its impact on coverage; and (2) impact of spatial decomposition of the network on communication overheads. We first study the impact of S_{max} on coverage and overhead in order to determine an appropriate value for the parameter. We then use this value of S_{max} for the remaining experiments and present our evaluations.

7.1. Determining S_{max}

Based on the study presented in Section 5.3, we set the number of cameras to be 60, and the number of targets to be 100, and track coverage accuracy and end-to-end delay for different values of S_{max} . Cameras and targets are deployed in random manner. For the hierarchical algorithm, end-to-end delay is computed for each cluster, and maximum delay is considered as worst-case delay.

For different values of S_{max} , Figure 20 plots the variation in the coverage, while Figure 21 plots the overhead in terms of end-to-end delay observed. As the value of S_{max} increases, the coverage accuracy of Hierarchical tends towards Optimal; the other algorithms are oblivious to S_{max} . For a 60 camera network, S_{max} set to 60 provides optimal coverage. While Hierarchical always seems to provide better coverage compared to the DFA, it outperforms the centralized heuristic (CFA), for $S_{max} \geq 30$. On the contrary, the end-to-end delay for Hierarchical is even less than that of DFA for $S_{max} \leq 30$. Thus, for the remaining experiments, we set S_{max} to be 30. This shows the benefit of spatial decomposition of the network and the tunability parameter, S_{max} .

The difference in end-to-end delay for Centralized and Hierarchical algorithms for $S_{max} = 60$ is because for hierarchical policy, the cluster-head itself is a camera, while for the centralized policies, the base-station is not. Moreover, for hierarchical algorithm, cluster-head is generally selected to minimize the communication overhead within a cluster, which may not be the case for centralized algorithms, since the location of base-station is selected randomly within the network in order to avoid its placement-centric deviations in the results.

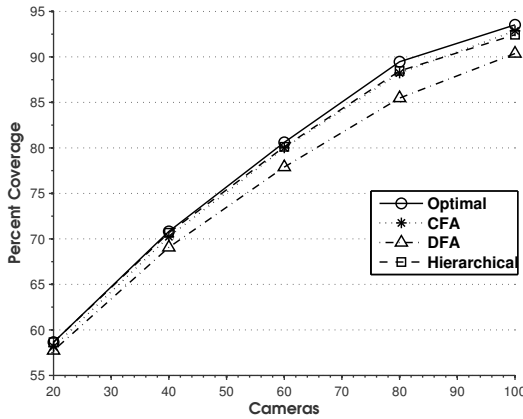


Fig. 22: Impact of camera density on coverage when targets are deployed in clustered manner.

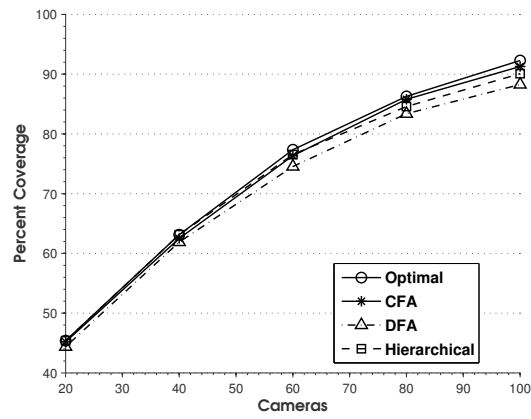


Fig. 23: Impact of camera density on coverage when targets are deployed in random manner.

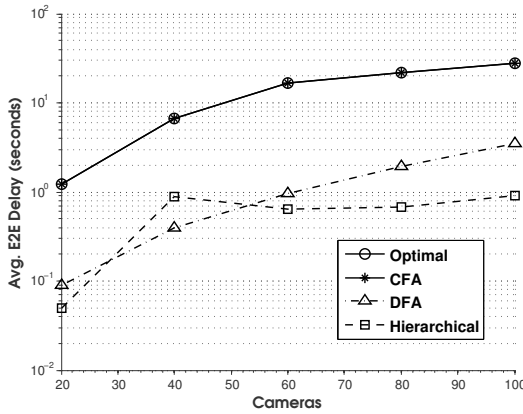


Fig. 24: Impact of camera density on end-to-end delay.

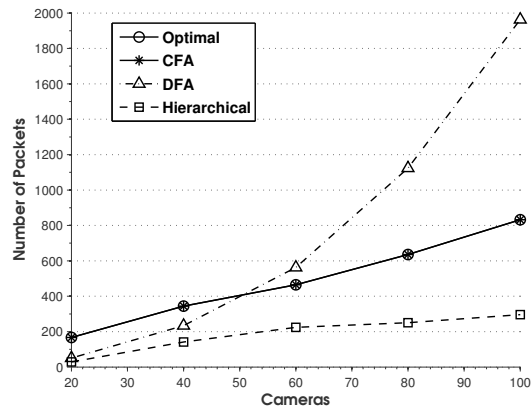


Fig. 25: Impact of camera density on messaging overhead.

7.2. Impact of Camera and Target Placement

In this experiment, we study the performance of Hierarchical specific to varying scale and density of cameras and distributions of targets, when S_{max} is set to be 30. Specifically, we deploy cameras in random fashion, and consider random and clustered distribution for targets. Figure 22 shows the impact of varying camera density when targets are deployed in a clustered manner. To generate clustered distribution of targets, we use the existing algorithm to generate inhomogeneous node distributions [Bettstetter et al. 2007]. It can be observed that the coverage achieved by Hierarchical is always greater than that of DFA, and it closely tracks the Optimal, which shows that Hierarchical performs better even for the clustered distribution of targets.

For the remaining experiments, we place 100 targets in random manner. Figure 23 shows that the Hierarchical always outperforms the DFA, and tracks Optimal, even as the network scale increases. In terms of end-to-end delay, as shown in Figure 24, Hierarchical significantly outperforms the centralized algorithms, and even the DFA as the scale of the network increases. Specifically, as the number of cameras grow beyond S_{max} , the end-to-end delay for Hierarchical almost stabilizes to an average of 0.78 sec-

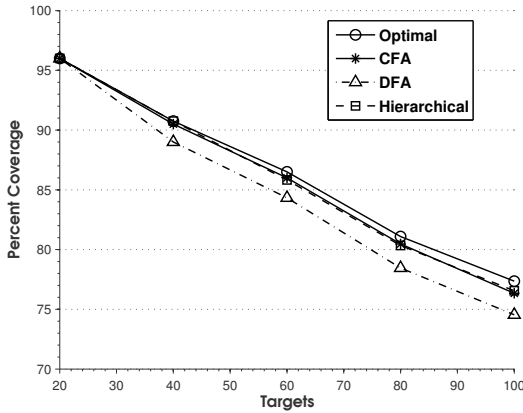


Fig. 26: Impact of varying target density on coverage.

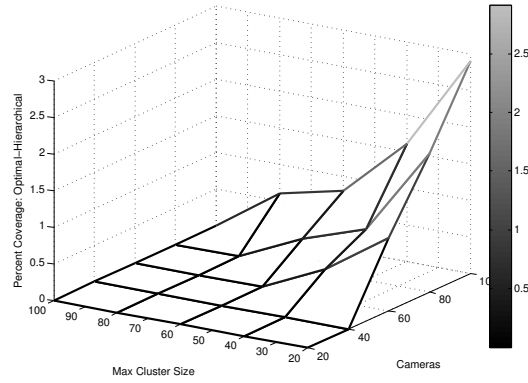


Fig. 27: Impact of S_{max} and camera density on overall coverage achieved by the Hierarchical with respect to the Optimal (Hierarchical coverage is subtracted from the Optimal coverage.)

onds, which is almost 96% better than the centralized algorithms, and 54% better than the distributed algorithm, on average. For smaller scales, the delay required for Hierarchical is only slightly worse than DFA. Similarly, as shown in Figure 25, Hierarchical incurs significantly less messaging overhead than the centralized and distributed algorithms due to the message exchanges limited only within spatially nearby cameras, and involving extremely less number of hops. This shows that Hierarchical provides close to the Optimal coverage and orders of magnitude less overhead, irrespective of the scale and density of the network.

Figure 26 shows the impact on coverage by varying the density of targets deployed in random manner. We deploy 60 cameras and vary the number of targets placed on a fixed 1000×1000 sq. meters terrain. As it can be observed from the figure, the benefit in coverage for Hierarchical is almost independent of the target density, and it continues to track the Optimal closely. In terms of communication overheads, the end-to-end delay observed for this experiment was 0.64 seconds and 224.10 packets were exchanged in total, as shown in the Figure 24 and the Figure 25 for 60 cameras scenario.

7.3. Sensitivity of S_{max} to Camera Density

As we discussed before, the Hierarchical first attempts to exploit natural separation among cameras to create less dependent clusters, but when the camera density is very high, it forces them to decompose by setting appropriate S_{max} . Thus, in this study, we evaluate the correlation between S_{max} and camera density. We place 100 targets randomly on a 1000×1000 sq. meters terrain.

For varying S_{max} and camera density, Figure 27 shows the difference of coverage obtained by Optimal and Hierarchical. It is interesting to note that for lower S_{max} values, even a small increment in S_{max} brings a large benefit in the coverage of Hierarchical. For instance, changing the S_{max} from 20 to 40 results in reducing the gap between Hierarchical and Optimal by almost 50%.

7.4. Sensitivity of FoV Overlaps

For the evaluations so far, we used 8 discrete non-overlapping FoVs (Pans), with AoV set to 45° , which is also the step size since the FoVs are non-overlapping. However, the discrete but overlapping FoVs with smaller step sizes can increase the coverage accu-

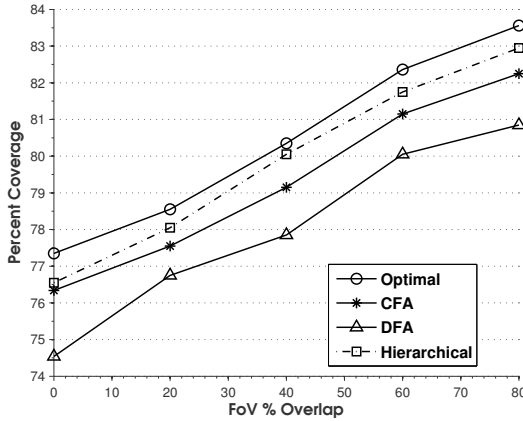


Fig. 28: Impact of varying FoV overlaps on coverage. The X-axis denotes the overlap in % between two consecutive FoVs.

FoV % Overlap	Avg. E2E Delay (in seconds)	Number of Packets
0	0.95	562.80
20	0.98	571.70
40	0.98	578.00
60	1.00	592.70
80	1.02	599.50

Fig. 29: Avg. end-to-end delay (in seconds) and number of packets for DFA. The other policies are omitted because their overhead is oblivious to the degrees of FoV overlaps.

racy, by possibly covering more target combinations. Thus, in this study, we evaluate the impact of overlapping FoVs on coverage accuracy of the proposed policies.

We place 60 cameras and 100 targets randomly on a 1000×1000 sq. meters terrain and vary the step size to generate FoVs corresponding to the varying degrees of overlaps. Figure 28 shows that the overall coverage accuracy increases as the % overlap between consecutive FoVs increases. In terms of communication overhead, since the centralized and semi-centralized algorithms do not communicate FoV-specific data, their overhead is oblivious to the degrees of FoV overlaps. An increase in the FoV overlaps translates to a possible increase in the number of feasible FoVs. Since DFA forces cameras to update their neighbors whenever cameras change their FoVs, the communication overhead for DFA increases, as shown in Figure 29.

7.5. Handling Target Mobility

So far we assumed that the targets are static and developed coverage maximization algorithms for them. However, if the targets are mobile, then the coverage accuracy will decrease as the targets move away from their original position. To address this problem, the same algorithms can be invoked periodically to achieve continued close-to-optimal coverage. Specifically, the lower the interval between two successive invocations (say periodic interval), the higher the coverage accuracy. However, for it to be effective, the cameras must be reconfigured in less than the periodic interval time. As a result, the communication overhead, especially the worst-case end-to-end delay, plays an important role for coverage maximization for mobile targets.

In this study, we compare the coverage accuracy achieved by centralized (with *Optimal* algorithm being run on the central base-station), distributed (*DFA*), and semi-centralized (*Hierarchical*) algorithms, when invoked periodically (every 5 seconds). We place 60 cameras and 100 targets randomly on a 1000×1000 sq. meters terrain. The targets are configured to move with the random-waypoint mobility model, and their minimum and maximum speeds are set to that of the pedestrian mobility model. All the policies begin to execute at 50 second simulation time.

Figure 30 shows that the centralized policy begins with high coverage accuracy, however due to the extremely high reconfiguration delays, as shown in Figure 17 and Figure 18, its performance deteriorates with time. Specifically, the targets move so much

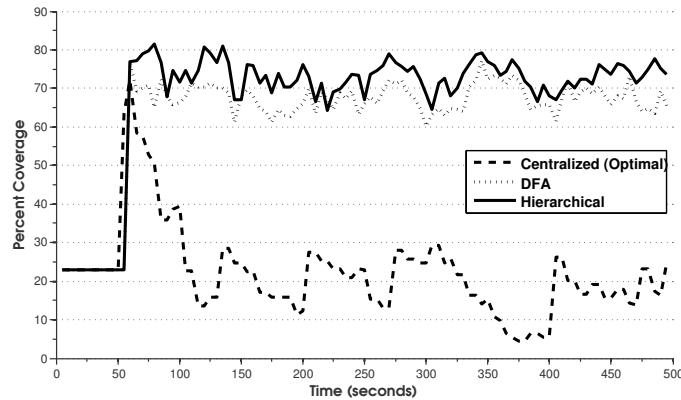


Fig. 30: Impact of target mobility on coverage accuracy for centralized, distributed and semi-centralized policies. These policies are invoked periodically (every 5 seconds) to reconfigure the cameras. *Centralized (Optimal)* represents that the *Optimal* policy is run at the central base-station to generate optimal camera-pan pairs. Targets are moving with random-waypoint mobility model, and their minimum and maximum speeds are set to that of the pedestrian mobility model.

from their recorded positions for computing optimal camera-pan pairs, that the newly received pan values turn out to be far from the optimal. On the other hand, the DFA and Hierarchical achieve close-to-optimal coverage due to their fast reconfiguration operations. Since Hierarchical has the lowest reconfiguration latency, it outperforms DFA even for the mobile target scenario. In terms of messaging overhead, the centralized algorithm sends 2887 packets, DFA sends 38741 packets, while the Hierarchical sends less than the half of that of DFA, 17620 packets. The low messaging overhead for centralized is due to the increased TCP backoffs over time, causing fewer packets to be transmitted.

8. RELATED WORK

The distinguishing feature of smart camera networks in comparison to traditional multi-camera surveillance systems is their ability to self-configure to best match the application requirements. This self-configuration minimizes the need for operator involvement both in monitoring and controlling the video stream. Moreover, they can be more vigilant and effective in monitoring than human operators who are prone to error due to fatigue and lapses in attention [Miller et al. 1998]. In fact, in the experiment conducted by Sandia National Laboratories for the U.S. Department of Energy demonstrates that even a dedicated and well-intentioned individual has at most 20 minutes of attention span of acceptable levels [Green 1999]. Adding the intelligence in cameras enables them to coordinate to adapt their coverage in response to what they are observing, to optimize their use of the network resources, and to summarize their observations in a way that reduces or eliminates operator involvement. This allows such systems to scale (as measured, for example, by the ratio of cameras to operators needed.)

Coverage management in sensor networks is an important problem and it has been studied thoroughly in omni-directional sensor networks [Meguerdichian et al. 2001; Wang et al. 2003; Zhang and Hao 2005; Huang and Tseng 2005; Howard et al. 2002; Kumar et al. 2004]. The problem becomes more challenging for directional or visual sensor networks, since the coverage quality can change significantly depending on the current orientation of the sensor. In general, usefulness of coverage by directional sen-

sors can be improved by either placing sensors carefully or configuring them (orientation and zoom, if available) so that minimum number of sensors can be used to meet the application specific coverage requirement. Thus, in this section, we present related work on sensor placement and configuration to maximize overall covered area or targets.

Sensor Placement. One of the classic coverage optimization problems, the Art Gallery Problem [Urrutia 2000], focuses on placing a minimum number of security guards in an art gallery so that all points in the whole gallery will be under observation. A similar problem with visual sensors is formulated by Hrster et al. [Hörster and Lienhart 2006] and several heuristics are proposed to solve them. Our work assumes that the cameras are placed already (using some placement algorithm), and focuses on configuring the cameras' FoV to maximize coverage.

Area Coverage Maximization. For a given placement of cameras, Kansal et al. [Kansal et al. 2007] focus on configuring pan, tilt, and zoom parameters of the cameras to maximize the overall area covered by all the cameras, and propose a distributed mechanism to configure the cameras. For directional sensor networks, Cheng et al [Cheng et al. 2007] propose the Maximum Directional Area Coverage (MDAC) problem to maximize the total area covered by a DSN, while minimizing the number of active sensors. In another instance, Erdem et al. [Erdem and Sclaroff 2006] consider the problem of determining automatic camera deployment layout to satisfy task-specific and floorplan-specific coverage requirements. Hoffmann et al [Hoffmann et al. 2008] propose a distributed algorithm to maximize the total area coverage, where each camera tries to minimize coverage overlaps with its neighboring cameras. Ma et al. [Ma et al. 2009] propose a 3D model for camera coverage, where cameras can adjust their pan and tilt values. They propose a simulated annealing based heuristic to orient cameras such that the overall area covered gets maximized. Fusco et al. [Fusco and Gupta 2010] focus on different variations of the problem of minimizing total uncovered area by placing and rotating directional sensors. Although these works address the problem of maximizing coverage by directional sensors, they focus mainly on maximizing the coverage of a given area, which is different from our goal of maximizing target-oriented coverage.

Camera Scheduling. When it is essential to cover all the coverable targets and when there are not enough cameras to satisfy that requirement, cameras can be scheduled to cover targets over time. Krahnstoever et al. [Krahnstoever et al. 2008] propose a solution for typical surveillance class applications, where the goal is to cover each target for a fixed interval. Thus, this paper focuses on obtaining an optimal scheduling pattern for covering targets, such that the total information gain from the overall coverage is maximum. Qureshi et al. [Qureshi and Terzopoulos 2009] present a planning strategy to achieve a close-up biometric coverage of selected pedestrians till they are present in the coverage region. Wang et al. [Wang and Cao 2011] attempt to minimize the service delay (orientation delay) of directional sensors to cover targets over time. They prove that the problem is NP-complete and provide centralized and distributed greedy solutions with bounded performance. These works differ from our work in terms of the objective: we focus on applications where it is important to cover as many targets as possible continuously.

Target-oriented Coverage. Some of the recent works in the context of *directional* sensor networks have provided optimization based solutions to address a basic instance of the problem of covering maximum targets [Cai et al. 2007; Ai and Abouzeid 2006]. Ai et al. [Ai and Abouzeid 2006] formulate the coverage optimization problem as a Maximum Coverage with Minimum Sensors (MCMS) problem, and show that MCMS

is NP-complete. They formulate the MCMS problem as an Integer Linear Programming (ILP) problem to propose a centralized algorithm to solve the problem. However, since the centralized algorithm is not scalable, they provide greedy based centralized (CGA) and distributed (DGA) algorithms for coverage maximization for DSNs. Cai et al. [Cai et al. 2007] focus on the problem of maximizing the lifetime of a directional sensor network. Essentially, they compute Multiple Directional Cover Sets (MDCSs) for a given set of targets. A Directional Cover Set (DCS) is a subset of directions of the sensors, which can cover all the targets present in the area under observation. Thus, by computing multiple such DCSs, they can alternately activate only one DCS at any given time, while keeping the other sensors in a sleep state. This helps them increase the overall network lifetime. The authors show that the problem of computing DCS and MDCS is NP-complete and propose heuristics to solve MDCS. Soto et al. [Soto et al. 2009] present a game-theoretic algorithm to obtain optimal camera configuration to acquire all targets at desired resolution. Fusco et al. [Fusco and Gupta 2009] focus on achieving k -coverage of a given set of targets (or area) using minimum number of directional sensors. They prove the problem to be NP-hard, and propose a greedy algorithm along with performance guarantees.

In the context of camera networks, Qureshi et al. [Qureshi and Terzopoulos 2008] focus on the problem of selecting cameras to carryout the given tasks, such as tracking a given set of pedestrians. They propose a distributed camera network control strategy that enables performing the surveillance tasks by groups of cameras. For evaluation, they have developed a simulated network of smart cameras that can monitor simulated pedestrians in a large virtual public space, such as a train station. In another interesting work, Johnson et al [Johnson and Bar-Noy 2011] focus on configuring pan and zoom based cameras (for instance, cameras in unmanned aerial vehicles (UAVs)) to cover maximum number of targets with highest precision possible.

Our problem is closely related to that of Ai et al. [Ai and Abouzeid 2006]: and we implement and use their proposed algorithms as benchmarks for comparison. On the other hand, the works by Cai et al. [Cai et al. 2007] and Soto et al. [Soto et al. 2009] are different since they assume that *all* the targets can be covered with certain camera-pan configurations. This assumption makes the problem an instance of the set-cover problem, attempting to find the optimal configuration to minimize the number of sensors needed to cover the targets. In contrast, we consider scenarios where the number of cameras is insufficient to cover the targets, making the problem an instance of the maximum coverage problem [Hochbaum 1996; Ai and Abouzeid 2006].

There is significant research on tracking single or multiple targets [Ercan et al. 2007; Porikli and Divakaran 2003; Ukita et al. 2000]. However, this research is not geared towards maximizing the number of targets under observation (the coverage maximization problem.)

9. CONCLUSION AND FUTURE WORK

In smart camera networks, coverage control is necessary to allow automatic tracking of targets without human intervention, allowing these systems to scale. In this paper, we consider the problem of automatic control of the cameras to maximize coverage of a set of targets. We presented an ILP-based formulation of the coverage maximization problem. Since the ILP problem is NP-hard, we proposed novel centralized (CFA) and distributed (DFA) heuristics that provide near-optimal performance, and provide better coverage accuracy compared to the state-of-the-art. In fact, even the proposed distributed algorithm (DFA) outperforms the existing centralized algorithm (CGA). We proved that the centralized and distributed algorithms are 0.5-approximate. We also demonstrated the effectiveness of the proposed algorithms using a miniaturized camera network testbed.

Centralized solutions work well for small-scale networks, however as the network scale increases, the distributed solutions are favored at the cost of coverage accuracy. Thus, we proposed an Hierarchical algorithm which exploits the dependencies among cameras to decompose them into less dependent clusters, where coverage maximization can be run separately in each cluster. Our evaluation shows that the Hierarchical algorithm provides close-to-optimal coverage accuracy that is significantly better than that of the DFA. Furthermore, for a 100 camera deployment scenario, the networking delay to configure the whole network by hierarchical algorithm was reduced by 97% and by 74% compared to that of the centralized algorithm and the distributed algorithm, respectively. Finally, we showed that these algorithms can be extended to provide close-to-optimal coverage when targets are mobile, by invoking them periodically. Even for mobile targets scenario, the Hierarchical algorithm outperforms the centralized and distributed algorithms due to its faster camera reconfiguration operations.

Moving forward, we plan to extend and optimize the proposed solutions for ensuring maximum coverage in case of mobile targets, and study the effectiveness of centralized, distributed, and semi-centralized algorithms for different target-mobility scenarios in simulations as well as on a real testbed. Next, we would like to focus on maximizing coverage when static/mobile obstacles are present in the deployment area.

REFERENCES

- ADAMS, A. AND BAKER, R. 1995. The Camera (Ansel Adams Photography, Book 1). *Bulfinch, reprint edition*.
- AI, J. AND ABOUZEID, A. 2006. Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization*.
- AKYILDIZ, I., MELODIA, T., AND CHOWDHURY, K. 2007. A survey on wireless multimedia sensor networks. *Computer Networks* 51, 4, 921–960.
- axis-213. Axis 213 ptz network camera. Available at http://www.axis.com/products/cam_213/.
- BALAKRISHNAN, H., PADMANABHAN, V., SESHAN, S., STEMM, M., AND KATZ, R. 1998. Tcp behavior of a busy internet server: Analysis and improvements. In *INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Vol. 1. IEEE, 252–262.
- BARTON-SWEENEY, A., LYMBERPOULOS, D., AND SAWIDES, A. 2006. Sensor localization and camera calibration in distributed camera sensor networks. In *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*. Ieee, 1–10.
- BELLARD, F., NIEDERMAYER, M., ET AL. 2007. Ffmpeg.
- BETTSTETTER, C., GYARMATI, M., AND SCHILCHER, U. 2007. An inhomogeneous spatial node distribution and its stochastic properties. In *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*. ACM, 404.
- BLACK, J., ELLIS, T., AND ROSIN, P. 2002. Multi view image surveillance and tracking. In *Motion and Video Computing, 2002. Proceedings. Workshop on*. IEEE, 169–174.
- CAI, Y., LOU, W., LI, M., AND LI, X. 2007. Target-Oriented Scheduling in Directional Sensor Networks. *IEEE Infocom*.
- CHENG, W., LI, S., LIAO, X., CHANGXIANG, S., AND CHEN, H. 2007. Maximal Coverage Scheduling in Randomly Deployed Directional Sensor Networks. In *Parallel Processing Workshops, 2007. ICPPW 2007. International Conference on*. 68–68.
- COLLINS, R., LIPTON, A., KANADE, T., FUJIYOSHI, H., DUGGINS, D., TSIN, Y., TOLLIVER, D., ENOMOTO, N., HASEGAWA, O., BURT, P., ET AL. 2000. *A System for Video Surveillance and Monitoring*. Carnegie Mellon University, the Robotics Institute.
- ERCAN, A., EL GAMAL, A., AND GUIBAS, L. 2007. Object tracking in the presence of occlusions via a camera network. *Proceedings of the 6th international conference on Information processing in sensor networks*.
- ERDEM, U. AND SCLAROFF, S. 2006. Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. *Computer Vision and Image Understanding* 103, 3, 156–169.
- FUSCO, G. AND GUPTA, H. 2009. Selection and orientation of directional sensors for coverage maximization. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on*. 1–9.

- FUSCO, G. AND GUPTA, H. 2010. Placement and orientation of rotating directional sensors. In *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*. 1–9.
- G. BRADSKI ET AL. 2000. OpenCV (open source computer vision). [Online] <http://opencv.willowgarage.com/wiki/>.
- GARCIA, M. AND SOLANAS, A. 2004. 3D simultaneous localization and modeling from stereo vision. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*. Vol. 1. IEEE, 847–853.
- GREEN, M. 1999. The appropriate and effective use of security technologies in us schools: A guide for schools and law enforcement agencies series: Research report. *NCJ*.
- HAMPAPUR, A., BROWN, L., CONNELL, J., EKIN, A., HAAS, N., LU, M., MERKL, H., PANKANTI, S., CENTER, I., AND HAWTHORNE, N. 2005. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. *Signal Processing Magazine, IEEE* 22, 2, 38–51.
- HOCHBAUM, D. 1996. Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. In *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 143.
- HOFFMANN, M., WITTKKE, M., HAHNER, J., AND MULLER-SCHLOER, C. 2008. Spatial Partitioning in Self-Organizing Smart Camera Systems. *IEEE Journal of Selected Topics in Signal Processing, vol. 2, issue 4, pp. 480-492* 2, 480–492.
- HÖRSTER, E. AND LIENHART, R. 2006. On the optimal placement of multiple visual sensors. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*. ACM, 111–120.
- HOWARD, A., MATARIC, M., AND SUKHATME, G. 2002. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proc. of the sixth International Symposium on Distributed Autonomous Robotics Systems (DARS)*.
- HUANG, C. AND TSENG, Y. 2005. The coverage problem in a wireless sensor network. *Mobile Networks and Applications Journal* 10, 4, 519–528.
- ILOG, I. 2001. ILOG AMPL CPLEX System. ILOG, Inc., Incline Village, NV.
- JOHNSON, M. AND BAR-NOY, A. 2011. Pan and scan: Configuring cameras for coverage. In *INFOCOM, 2011 Proceedings IEEE*. IEEE, 1071–1079.
- JOHNSON, S. 1967. Hierarchical clustering schemes. *Psychometrika* 32, 241–254. 10.1007/BF02289588.
- KANSAL, A., KAISER, W., POTTIE, G., SRIVASTAVA, M., AND SUKHATME, G. 2007. Reconfiguration methods for mobile sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 3, 4, 22–es.
- KRAHNSTOEVER, N., YU, T., LIM, S., PATWARDHAN, K., AND TU, P. 2008. Collaborative Real-Time Control of Active Cameras in Large Scale Surveillance Systems. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*.
- KULKARNI, P., GANESAN, D., SHENOY, P., AND LU, Q. 2005. SensEye: a multi-tier camera sensor network. In *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM New York, NY, USA, 229–238.
- KUMAR, S., LAI, T., AND BALOGH, J. 2004. On k -coverage in a mostly sleeping sensor network. In *Proc. International Conference on Mobile Computing and Networking (MobiCom)*.
- MA, H., ZHANG, X., AND MING, A. 2009. A coverage-enhancing method for 3d directional sensor networks. In *Proc. INFOCOM Mini-conference*. 2791–2795.
- MEGUERDICHIAN, S., KOUSHANFAR, F., POTKONJAK, M., AND SRIVASTAVA, M. 2001. Coverage problems in wireless ad hoc sensor networks. In *Proc. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. 1380–1387.
- MILLER, J., SMITH, M., MCCAULEY, M., CENTER, U. S. C. G. R. . D., RESOURCES, U. S. C. G. H., AND MONTEREY TECHNOLOGIES, I. 1998. *Crew Fatigue and Performance on US Coast Guard Cutters*. US Coast Guard Research & Development Center.
- MIYAMOTO, K. 1964. Fish eye lens. *JOSA* 54, 8, 1060–1061.
- PAPAGEORGIU, C., OREN, M., AND POGGIO, T. 1998. A general framework for object detection.
- PORIKLI, F. AND DIVAKARAN, A. 2003. Multi-camera calibration, object tracking and query generation. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*. Vol. 1.
- QUALNET-SIMULATOR. Scalable Network Technologies. Inc.[Online]. Available: www.qualnet.com.
- QURESHI, F. AND TERZOPOULOS, D. 2008. Smart camera networks in virtual reality. *Proceedings of the IEEE* 96, 10, 1640–1656.
- QURESHI, F. AND TERZOPOULOS, D. 2009. Planning Ahead for PTZ Camera Assignment and Handoff. In *ACM/IEEE International Conference on Distributed Smart Cameras*. 1–8.

- SCHLEICHER, D., BERGASA, L. M., BAREA, R., LOPEZ, E., AND OCANA, M. 2006. Real-time simultaneous localization and mapping using a wide-angle stereo camera. In *DIS '06: Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*. IEEE Computer Society, Washington, DC, USA, 55–60.
- SIBSON, R. 1973. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal* 16, 1, 30–34.
- SORO, S. AND HEINZELMAN, W. 2009. A survey of visual sensor networks. *Advances in Multimedia 2009*, 640386, 1–21.
- SOTO, C., SONG, B., AND ROY-CHOWDHURY, A. 2009. Distributed multi-target tracking in a self-configuring camera network. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 1486–1493.
- UKITA, N., NAGAO, T., AND MATSUYAMA, T. 2000. Versatile Cooperative Multiple-Object Tracking by Active Vision Agents. In *MVA2000, LAPR Workshop on Machine Vision Applications*.
- URRUTIA, J. 2000. Art gallery and illumination problems. *Handbook of Computational Geometry*, 973–1027.
- WANG, C., THORPE, C., AND SUPPE, A. 2003. Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*. IEEE, 416–421.
- WANG, X., XING, G., ZHANG, Y., LU, C., PLESS, R., AND GILL, C. 2003. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proc. First International Conference on Embedded Networked Sensor Systems (SenSys)*.
- WANG, Y. AND CAO, G. 2011. Minimizing service delay in directional sensor networks. *IEEE INFOCOM, 2011*, 1790 – 1798.
- ZHANG, H. AND HAO, J. 2005. Maintaining sensing coverage and connectivity in large sensor networks. *Ad hoc and Sensor Wireless Networks Journal* 1, 89–124.