Journal of Network and Computer Applications 75 (2016) 1–15

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

The coverage problem in visual sensor networks: A target oriented approach

Hafsa Zannat, Taslima Akter, Mashrura Tasnim, Ashikur Rahman*

Department of Computer Science & Engineering (CSE) Bangladesh, University of Engineering & Technology (BUET), Dhaka 1000, Bangladesh

ARTICLE INFO

Article history:
Received 7 April 2016
Received in revised form
31 July 2016
Accepted 15 August 2016
Available online 20 August 2016

Keywords:
Sensor network
Target oriented heuristic
Performance lower bound

ABSTRACT

A visual sensor network (VSN) consists of a number of self-configurable visual sensors with adjustable spherical sectors of limited angle (often known as *Field of View*) that are meant to cover a number of targets randomly positioned over a deployment area. One of the fundamental problems of VSNs is to cover maximum number of targets using minimum number of sensors. This classical *Min–Max* problem is known to be NP-hard. The existing heuristics have a number of weaknesses that influence their coverage. Therefore, we present novel centralized heuristics that provide near-optimal coverage by prioritizing the targets that are coverable by *fewer* cameras. We also provide approximation bounds for both existing heuristics and the proposed heuristics and theoretically proved that in certain scenarios, the proposed heuristics outperform the existing ones. Finally, we provide performance comparison of the new heuristics with other heuristics through extensive simulation experiments using synthetic data set of successively increasing size.

1. Introduction

Visual sensor networks have drawn considerable attention of researchers as a branch of sensor networks with enormous applicability in real-world scenarios. As low cost video-transmitting sensors are becoming more and more available, the opportunity to deploy them in fields like surveillance system, environment monitoring, smart traffic controlling system etc. (Akyildiz et al., 2007; Soro and Heinzelman, 2009; Rashid and Rehmani, 2016) has tremendously been increased. Usually a visual sensor network, also known as a *Smart Camera Network* (SCN), consists of a set of targets to be monitored by a set of smart sensors capable of self-controlling their orientations. Although the primary goal of this network is to monitor as many targets as possible, the efficiency of such system depends on the extent of camera usage as fewer number of active sensors implies lower energy consumption and longer network life time.

Visual sensor networks can be broadly classified in two different categories: –(i) *over-provisioned* systems, and, (ii) *under-provisioned* systems. We call a VSN is *over-provisioned* if the number of sensors is sufficient to cover all the targets and *under-provisioned* otherwise. In under-provisioned systems, one needs to maximize the target coverage regardless of the number of cameras being used because the system does not have enough cameras to cover all the targets. On the contrary, the coverage task is more challenging in an over-provisioned system compared to the under-provisioned because for the former system one needs to minimize the number camera usage besides maximizing coverage. In this paper, we focus on over-provisioned systems and provide solutions to minimize the number of cameras and maximize the number of covered targets. The problem is eventually an instance of the classical set cover problem whose optimization version is known to be NP-hard (Ai and Abouzeid, 2006; Cai et al., 2007). Due to the hardness of the problem, many polynomial time heuristics (Ai and Abouzeid, 2006; Fusco and Gupta, 2009; Munishwar and Abu-Ghazaleh, 2011, 2013) have been proposed over the past few years. Also an Integer Linear Programming (ILP) formulation exists in the literature (Ai and Abouzeid, 2006) to solve the problem. However, it is not suitable for *large-scale* sensor networks due to its high computational complexity. Therefore, polynomial time heuristics are needed to tackle the issue.

One can envision two approaches to devise a heuristic: –(i) *sensor-oriented* approach, and (ii) *target-oriented* approach. In the first approach (Ai and Abouzeid, 2006; Fusco and Gupta, 2009; Munishwar and Abu-Ghazaleh, 2011, 2013), one can look into the cameras and determine the exact *coverage counts* in different spherical sectors (i.e., on different FoVs) of each camera. While counting the coverage, the overlapping regions of the neighboring cameras must also be considered to exclude the possibility of *redundant* coverage. Based on the coverage counts, one can iteratively select cameras that can cover maximal number of

* Corresponding author.
E-mail addresses: hafsa_zannat@yahoo.com (H. Zannat), tonni89@yahoo.com (T. Akter), mashrura_cse@yahoo.com (M. Tasnim), ashikur@cse.buet.ac.bd (A. Rahman).

http://dx.doi.org/10.1016/j.jnca.2016.08.015
1084-8045/

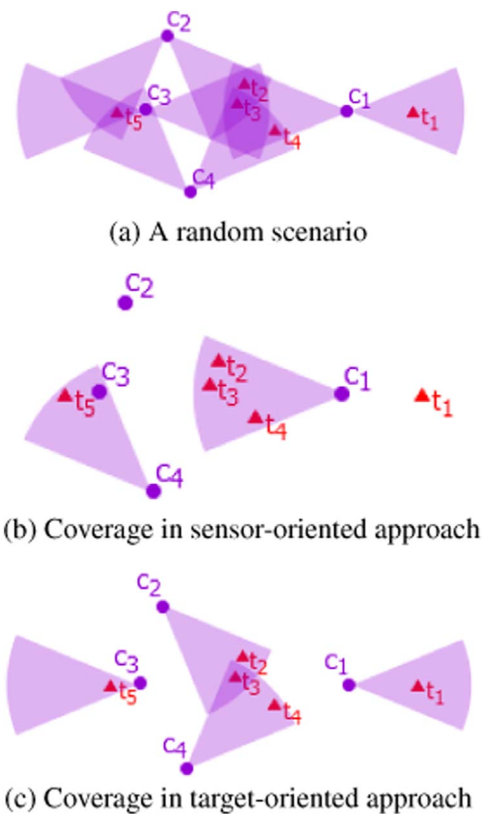


Fig. 1. Illustrating a drawback of sensor-oriented approach which is overcome by target-oriented approach.

(uncovered) targets. In the second approach, instead of looking into the cameras, one can look into the targets first. Some targets might be located in a difficult corner of the deployment area and could be covered only by a single or fewer number of cameras. In order to maximize the target coverage, those targets need to be covered first. Thus, one needs to *prioritize* the targets based on their *coverage vulnerability* and then select a minimal set of cameras that can cover targets in the order of their priorities. In this paper we adopt the target oriented approach and develop three heuristics to solve the problem.

To understand the need for a target-oriented approach we provide an example scenario in Fig. 1 where the sensor-oriented approach is unable to maximize the target coverage but the target-oriented approach overcomes the problem. In Fig. 1(a) we have shown a random deployment of five targets $\mathcal{T} = \{t_1, t_2, t_3, t_4, t_5\}$ and four available cameras $C = \{c_1, c_2, c_3, c_4\}$ to cover those five targets. All cameras are configurable directional sensors, i.e., have a fixed angle of view if set to a certain direction. Camera c_1 can cover the target $\{t_1\}$ in one direction and the targets $\{t_2, t_3, t_4\}$ in a different orientation. The camera c_2 and c_3 cover $\{t_2, t_3\}$ in one direction and the target $\{t_5\}$ in another direction. Finally, the coverage of c_4 are the set of targets $\{t_3, t_4\}$ and $\{t_5\}$ in two different directions. In sensor-oriented approach, coverage counts of all cameras in all possible directions are determined in this way and the camera which provides the most coverage in some pan is selected first. In this case, the camera c_1 will be chosen first as it covers three targets $\{t_2, t_3, t_4\}$ in one direction that no other camera can provide in a single orientation. Once c_1 is selected to cover $\{t_2, t_3, t_4\}$, the next camera to be selected is the one that provides the most coverage of the remaining uncovered targets. All of the remaining cameras c_2, c_3, and c_4 can cover only one uncovered target t_5. So it will select any of those three cameras next. Suppose it chooses c_4. After this selection the remaining cameras c_2 and c_3 can not cover any more uncovered targets and the algorithm stops with the configuration as shown in Fig. 1(b). Note that the target t_1 remains uncovered.

Now let us see how the target-oriented approach overcomes the problem. In this approach, the target covered by the fewest number of cameras is selected first. If we look into the targets, we can see that the target t_1 is covered by only $\{c_1\}$, t_2 is covered by $\{c_1, c_2, c_3\}$, t_3 can be covered by $\{c_1, c_2, c_3, c_4\}$, options for t_4 are $\{c_1, c_4\}$ and finally t_5 can be covered by $\{c_2, c_3, c_4\}$. Therefore t_1 is the most constrained target having lowest coverage count and must be covered first. c_1 will be selected first to cover t_1. After the direction of c_1 is set to cover t_1, the coverage of t_3 and t_4 is reduced to $\{c_2, c_3, c_4\}$ and $\{c_4\}$ respectively. Thus, among the remaining uncovered targets, t_4 has the least number of coverage and would be covered next by selecting c_4. When c_4 is oriented to cover t_4 the target t_3 get automatically covered. So only t_2 and t_5 remains uncovered after this step. t_5's coverage reduces to $\{c_2, c_3\}$ as c_4 has a different orientation. t_2's coverage set is also $\{c_2, c_3\}$. Therefore, both t_2 and t_5 have the same number of coverage and either of those can be picked up. If we select t_2 first then it can be covered by c_2 and t_5 can be covered by the other option c_3, although it could be other way around. After selecting and orienting the cameras in target-oriented approach the final configuration becomes the scenario shown in Fig. 1(c), where all targets are covered.

The summary of the major contributions of this paper is as follows:

(i) We propose a number of novel *target-oriented* heuristics for target coverage in smart camera networks. The proposed heuristics provide near-optimal coverage and outperform existing *sensor-oriented* heuristics addressing the same problem.

(ii) We provide approximation bounds on number of targets covered by the existing heuristics and the proposed heuristics.

(iii) We provide performance comparison of the new heuristics with the existing heuristics through extensive simulation experiments and show that the proposed ones outperform the existing ones.

The paper is organized as follows: This section introduces the problem we tackle in this paper and provides major motivation behind the work. Section 2 provides system model and definition of the required parameters. It also formally defines the problem that we solve in this paper. Section 3 mentions related research works and also separately discusses two existing sensor-oriented heuristics in details. Those two heuristics are used to compare the performance of the proposed target-oriented heuristics in later sections. In Section 4, we present target oriented heuristics (TOHs) as a plausible solution to the coverage problem of over-provisioned VSN and explain the working principle of the proposed heuristics with example scenario. Rigorous proof of performance lower-bounds of two existing heuristics and a greedy version of the target oriented heuristic are described in Section 5. Performance comparison of TOHs with existing heuristics with respect to several performance metrics under different network conditions in simulated environment is presented in Section 6. We draw a conclusion and discuss future works in Section 7.

2. System model and problem formulation

In this section, we formally introduce the VSN with relevant parameters. Then we terminologically describe the coverage problem that we solve in this paper. We also describe two well-known sensor-oriented existing heuristics that we use to compare the performance of the proposed target-oriented heuristics.

2.1. Visual sensor network description and parameters

The sensing region of a camera can be characterized by its Field of View (FoV) defined as follows.

Field of View (Fov): The FoV of a camera is the extent of the observable/sensing region that can be captured at any given direction. Some cameras come with fixed-FoV and for some, FoVs are adjustable. The smart cameras used in current VSNs are known as *Pan-Tilt-Zoom* (PTZ) cameras where FoV can be self-adjusted in three dimensions: (i) horizontal movement in pan, (ii) vertical movement or tilt, and (iii) change in depth-of-field by changing zoom. In this paper, we limit ourselves to pan-only cameras, i.e., we assume that a camera can move only in horizontal direction and its FoV is only described by its pan.

The pan of a camera is formally defined using the following two parameters:

(1) R_s: Maximum coverage *range* of the camera beyond which a target can not be detected with acceptable accuracy in a binary detection test.
(2) θ: The maximum sensing/coverage *angle* of a camera on a certain direction. This angle is also known as Angle of View (AoV).

Thus, when a camera is oriented towards a particular direction, it can cover a circular sector (called a pan) defined by R_s and θ. We assume that every camera possesses a specific number of non-overlapping pans, of which, only one can be selected in a particular deployment. For example: a camera with FoV defined by $\theta = \frac{\pi}{4}$ can pick any one of eight disjoint orientations. Fig. 2 depicts these parameters of camera coverage. Here, two cameras c_1 and c_2 have eight pans each and can be oriented towards any of these eight pans. We assume that cameras are homogeneous in terms of parameters. Position of a target and a sensor are expressed through Cartesian coordinates (x, y) in a two-dimensional plane. $\overrightarrow{d_{ik}}$ is a unit vector which cuts each pan (i.e., the sensing sector) into half representing the orientation of camera c_i towards pan p_k. $\overrightarrow{v_{ij}}$ is a vector in the direction from camera c_i to target t_j.

Target in Sector (TIS) Test : With TIS test (Ai and Abouzeid, 2006) one can verify whether a target t_j is coverable by a given sensor c_i. To conduct this test, at first we calculate the angle ϕ_{ij} between camera orientation $\overrightarrow{d_{ik}}$ of pan p_k and the target vector $\overrightarrow{v_{ij}}$.

$$\phi_{ij} = cos^{-1}\left(\frac{\overrightarrow{d_{ik}}\cdot\overrightarrow{v_{ij}}}{|\overrightarrow{d_{ik}}||\overrightarrow{v_{ij}}|}\right) \tag{1}$$

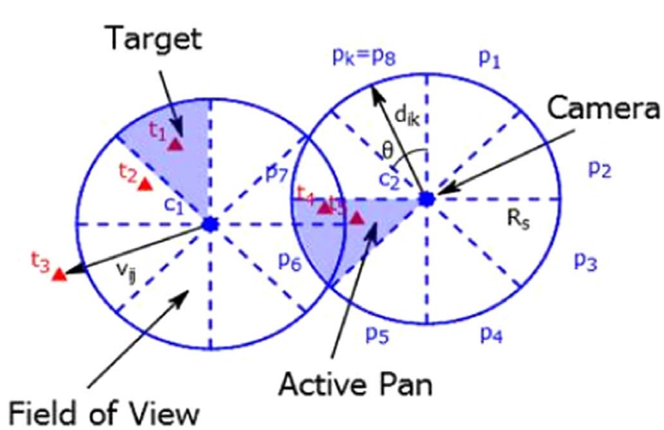


Fig. 2. Camera coverage parameters.

A target is coverable by a camera's FoV if the span of its FoV contains the target and the target is located within the sensing range of the camera. Geometrically, $\overrightarrow{d_{ik}}$ divides the pan p_k into two equal halves and if a target is located in either of them, it is coverable by that camera on the pan p_k. Thus, the angle ϕ_{ij} needs to be less than half of the AoV, i.e., $\phi_{ij} \le \frac{\theta}{2}$. The other condition requires that the target has to be inside the maximum sensing range of the camera, i.e.,: $|v_{ij}| \le R_s$.

Conducting TIS tests over every pan p_k of camera c_i and every target t_j, we can build a *binary coverage matrix* $A^M_{N\times Q}$ of the network comprising of M targets and N cameras with Q pans where an entry in the matrix can be calculated as:

$$a^j_{ik} = \begin{cases} 1 & \text{if target } t_j \text{ is covered by camera } c_i \text{ at pan } p_k \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

2.2. Problem formulation

In this section we provide a formal description of the problem. Suppose we have a set of M randomly deployed targets $\mathcal{T} = \{t_1, t_2, \ldots, t_M\}$ and a set of N homogeneous directional cameras $C = \{c_1, c_2, \ldots, c_N\}$, each of which can be (self) orientated towards one of the Q possible pans $\mathcal{P} = \{p_1, p_2, \ldots, p_Q\}$. Let us define a tuple $\langle c_i, p_k\rangle$ which is a camera-pan pair denoting a camera c_i oriented towards pan p_k for any $1 \le i \le N$ and $1 \le k \le Q$. Define a function $F(\langle c_i, p_k\rangle)$ which returns the set of targets covered by the tuple $\langle c_i, p_k\rangle$.

Suppose δ is the set of tuples that can cover at least one target, i.e.,

$$\delta = \{\langle c_i, p_k\rangle: |F(\langle c_i, p_k\rangle)| > 0\} \quad i \in [1, N];\ k \in [1, Q];$$

The goal is to find a subset $\delta' \subseteq \delta$ such that $|\delta'|$ is the minimum among all cardinalities of possible subsets of δ, and the quantity $\left|\bigcup_{\langle c_i, p_k\rangle \in \delta'} F(\langle c_i, p_k\rangle)\right|$ gets maximized.

In other words, we seek to find orientations of *minimum* number of cameras to cover *maximum* number of targets which is also traditionally known as *Maximum Coverage with Minimum Sensors* (MCMS) problem. We assume that the optimal solution applied to the same scenario covers all the targets utilizing a subset of available sensors, i.e. no target is left *uncovered* in the optimal solution for the given network topology.

3. Related works

A notable number of research activities have been done in the area of visual sensor networks during the past decade. The major goals of these research works are (Yap and Yen, 2014) design issues related to the optimal placement and configuration of the visual sensors, energy efficient scheduling methodologies to extend network lifetime, cost-effective transmission system for captured sensor data, and fault tolerant network design to obtain secured connectivity of the network. A comprehensive survey on different design aspects of VSN can be found in (Liu et al., 2016).

Appropriate placement and configuration of the sensors to achieve a predefined goal is an important design aspect for visual sensor networks. There are few works along this direction. Chakrabarty et al. (2002) worked on optimal placement problems while assuming the sensor field as a grid of points (coordinates). The visual sensors can be placed only in one of the gird points. They proposed Integer Linear Programming (ILP) model to solve small instances of the problem and a *divide and conquer* approach to solve large instances. Hörster and Lienhart (2006) proposed a rank based greedy heuristic for sensor placement, where *rank* of sensor depends on its cost and accrued target coverage. Although sensor placement is an important problem for VSNs, in this paper we limit ourselves only to the selection and orientation of the *randomly deployed* sensors.

For randomly deployed sensors the number of active sensors should be kept to a minimum due to energy limitations. Therefore, one important research question is what should be the most appropriate sensor selection scheme? There is a research thread devising several sensor selection schemes that are based on *sensor-oriented* approach. The objective of these works is to maximize target coverage using minimum number of sensors. Munishwar et al. suggested such a sensor-oriented approach (Munishwar and Abu-Ghazaleh, 2011) where they devised a distributed algorithm to achieve coverage maximization. In their approach, at first all sensors assign a unique priority value by themselves. Then each sensor detects the total number of targets it can cover in each direction and automatically orients itself in the direction of maximal target coverage. The orientation information is then communicated with the sensors located within twice of the sensing range. In case of two or more sensors covering overlapping regions, the orientation of the highest priority sensor is retained. For priority assignment, they proposed two approaches: – (i) an area based approach and (ii) a target based approach. In the area based approach higher priority is assigned to a camera having lower degrees of coverage overlaps with other cameras. On the other hand, in target based approach, a camera's priority is higher if it has lower number of useful pans. Johnson and Bar-Noy (2011) studied pan and zoom adjustable camera sensor networks. They provided polynomial time *dynamic programming* solution besides greedy approaches in both centralized and distributed manner for configuring sensors in order to achieve maximum coverage. The coverage issue in case of moving targets was studied by Adriaens et al. (2006), who provided a polynomial time algorithm for *breach* calculation. For a given deployment, their algorithm can determine the closest observable distance to a sensor, termed as *breach distance*, that any moving target must have at least once when traveling between two given regions to be covered. A game theoretic approach was adapted by Hatanaka et al. (2016). The authors discussed their pay-off based learning algorithm to choose an action to maximize accuracy. Dieber et al. (2011) jointly consider the coverage, QoS and resource allocation in VSNs. In other words, they consider usage of power, compression and transmission processing hardware as optimizable parameters without compromising Quality of Service (QoS). They provide evolutionary algorithm assuming the parameters as genes and through cross-over and mutation the algorithm obtains a set of Pareto-optimal configurations. The main difference between all of these works and our work is their approach is *sensor-oriented* but our approach is *target-oriented*. Also no theoretical performance bounds on sensor usage are known for these works but we provide such bounds for our work.

A significant challenge in designing sensor networks is to extend life-time of power-constrained wireless sensors. This issue has been studied widely. One of the common approaches to solve this problem is to divide the sensor set into disjoint subsets each of which independently ensures network coverage. The generated subsets can then be scheduled efficiently in a round robin fashion to monitor the targets. Cardei and Du (2005) developed max-cover heuristic using mixed integer programming to maximize coverage for each disjoint set of sensors. Efficient scheduling techniques of these disjoint sets were also described in (Tian and Georganas, 2002; Wang et al., 2011; Ye et al., 2003). An entropy based camera selection algorithm was proposed by Dai and Akyildiz (2009) to reduce the number of active sensors exploiting spatial correlation of over-provisioned system and to increase the network life-time. Addressing coverage problem for over-provisioned system, Slijepcevic and Potkonjakthe proposed an energy efficient incremental node deployment heuristic (Slijepcevic and Potkonjak, 2001). Although our proposed techniques are able to construct a single subset of available sensors covering all the targets, they can be executed iteratively to generate multiple disjoint subsets. In each iteration when a single subset is generated, all of its sensors can be removed from the available sensor set and the algorithm can be run again to generate another subset covering all the targets. The process may continue until either all the sensors are used up or no further subset covering all the targets is possible to generate.

Another major challenge in VSN is to ensure maintenance of connectivity and communication among sensor nodes in energy efficient manner. Shakkottai et al. (2005) presented calculation of several boundary conditions for various network parameters including sensing range, node density etc. ensuring coverage and connectivity. Wang et al. considered effect of attenuation and distortion on visual quality and proposed co-operative swarm optimization to obtain reliable transmission path. Gupta et al. designed and analyzed centralized and distributed algorithms for self-organization of connection topology in a sensor network to attain near optimal connected sensor coverage (Gupta et al., 2006). The connectivity issue of visual sensors is orthogonal to our problem and we do not explicitly address this issue in our work.

There are few other works which consider some variation of the basic camera selection problem. Such variations include mobility of cameras, presence of obstacles between camera and targets etc. Neishaboori et al. (2014) worked on covering a set of stationary targets located in a two-dimensional plane using homogeneous mobile pan-only cameras with known maximum AoV and coverage ranges. They proposed an algorithm to find the minimum number of cameras, their locations, and directions, such that each target is visible by at least one camera. In (Yap and Yen, 2016), the authors tackled occlusion caused by fixed and moving obstacles and presented Greedy Algorithm covering maximum number of *non-occluded* targets in each iteration. They exploited historical data to adjust random variables to solve the moving obstacles problem. In our work we do not consider the presence of obstacles although the heuristics that we develop can be augmented to consider the issue.

There also exists a few works (Fusco and Gupta, 2009; Lu and Yu, 2014; Costa et al., 2014) aiming at k-coverage problem in directional sensor networks. These works allow redundancy in sensor activation to ensure multiple coverage of all or prioritized/sensitive targets. Costa et al. (2014) proposed an algorithm to achieve a high level of monitoring redundancy for some critical targets by concurrently viewing them using more than one visual sensor. The optimal solution to k-coverage problem has been proved to be NP-hard by Fusco and Gupta (2009). They provided a centralized greedy solution to the problem assuming that each sensor has overlapping pans instead of discrete pans. Malek et al. (2016) identified a novel issue called *coverage balancing* in k-coverage problem and provided centralized solutions to jointly solve k-coverage and coverage balancing. The solutions to k-coverage problems address fault tolerance issue which is beyond the scope of this paper and we limit ourselves only to *single* coverage.

3.1. Detailed description of two existing sensor-oriented heuristics

The MCMS problem can be solved optimally using the Integer Linear Program (ILP) formulation described in (Ai and Abouzeid, 2006). However the solution to ILP formulation is not suitable for large-scale smart camera networks due to its high computational complexity and low computing capabilities of the embedded sensors. As a remedy the following two centralized heuristics were proposed:

Algorithm 1. Greedy Target Oriented Heuristic (GTOH).

Input: $\mathcal{C}$ = {set of unused cameras}; $\mathcal{T}$ = {set of uncovered targets}; $\mathcal{P}$ = {set of discrete pans}
Output:
$\mathcal{Z}$ = {Camera–pan pairs given by GTOH}
repeat
  Compute w_j for each $t_j \in \mathcal{T}$, and $\mathcal{R}_{ik}$ for each $p_k \in \mathcal{P}$ of $c_i \in \mathcal{C}$
  Construct a set $\mathcal{T}_1$ containing only the lonely targets of $\mathcal{T}$
  if $\mathcal{T}_1 \neq \phi$ **then**
    $S = \varnothing$
    for all $c_j \in \mathcal{C}$ **do**
      for all $p_l \in \mathcal{P}$ **do**
        $\mathcal{T}_{jl} = F(\langle c_j, p_l \rangle) \cap \mathcal{T}_1$
        $S = S \cup \{\mathcal{T}_{jl}\}$
      end for
    end for
    Find the set $\mathcal{T}_{ik}$ whose size is maximum in the set S.
    $\mathcal{Z} = \mathcal{Z} \cup \{\langle c_i, p_k \rangle\}$
    $\mathcal{C} = \mathcal{C} \backslash \{c_i\}$
    $\mathcal{T} = \mathcal{T} \backslash \mathcal{T}_{ik}$
  else
    Find the camera-pan pair $\langle c_m, p_n \rangle$ that has maximum rank $\mathcal{R}_{mn}$ among all available camera-pan pairs
    Suppose $\mathcal{T}_3$ is the set of targets covered by $\langle c_m, p_n \rangle$
    $\mathcal{Z} = \mathcal{Z} \cup \{\langle c_m, p_n \rangle\}$
    $\mathcal{C} = \mathcal{C} \backslash \{c_m\}$
    $\mathcal{T} = \mathcal{T} \backslash \mathcal{T}_3$
  end if
until $\mathcal{C} = \phi$ or $\mathcal{T} = \phi$ or no new $\langle c_i, p_k \rangle$ pair is found to be included in $\mathcal{Z}$

Centralized Greedy Algorithm (CGA). In CGA (Ai and Abouzeid, 2006), initially all the cameras remain *inactive* and all the targets are *uncovered*. At each iteration, the camera pan covering maximum number of uncovered targets (greedily) gets activated. The camera and the covered targets are removed from the set of *available* cameras and the set of *yet-to-be-covered* targets respectively. The process continues until either of the sets becomes empty.

CGA has a shortcoming at the "tie-breaking" situation when more than one cameras cover the same (maximal) number of targets (Munishwar and Abu-Ghazaleh, 2013) at a certain iteration. In tie-breaking situations, arbitrary choice of camera selection may degrade coverage.

Centralized Force-directed Algorithm (CFA). In (Munishwar and Abu-Ghazaleh, 2013), the authors presented CFA to overcome the shortcomings of CGA. They proposed a modification of CGA by assigning priorities of each camera pans. Instead of activating a camera having maximum number of uncovered targets to some pan, they proposed to activate the cameras having targets only in a singular pan first and cover those targets. In order to determine such cameras, they introduced a concept called *force* which is a measurement attached to each pan. The force of a camera-pan pair is the ratio of coverable targets in that camera pan and the total number of uncovered targets available in all possible pans. Mathematically, the force F_{ik} of camera c_i at pan p_k is defined as:

$$F_{ik} = \frac{|M_{ik}|}{|M_i|} \tag{3}$$

where $|M_{ik}|$ is the number of targets covered by camera c_i on pan p_k, and $|M_i|$ is the number of coverable targets by camera c_i in all possible pans. The pan with the highest *force* is always selected at each iteration. The selected camera and the covered targets are removed (like CGA) and the process continues until no more cameras can be selected or all the targets are covered.

4. Target oriented heuristics

In this section we propose several target oriented heuristics to solve the coverage problem of VSNs. In particular, we propose three heuristics: (i) Generalized Target Oriented Heuristic (GTOH), (ii) Purely Target Oriented Heuristic (PTOH), and (ii) Hybrid Target Oriented Heuristic (HTOH).

4.1. Greedy Target Oriented Heuristic (GTOH)

GTOH introduces a target oriented modification over CGA. Unlike CGA which assigns equal priority to each of the targets, the GTOH assigns higher priority to the targets coverable by only one camera (dubbed as *lonely targets*) compared to the targets coverable by more than one camera. However, the targets coverable by multiple cameras are assigned equal priority regardless of the number of cameras covering them.

Algorithm 2. Purely Target Oriented Heuristic (PTOH).

Input: $\mathcal{C}$ = {set of unused cameras}; $\mathcal{T}$ = {set of uncovered targets}; $\mathcal{P}$ = {set of discrete pans}
Output: $\mathcal{Z}$ = {Camera–pan pairs given by PTOH}

repeat
  Construct bipartition **B** and compute w_j for each $t_j \in \mathcal{T}$, and $\mathcal{R}_{ik}$ for each $p_k \in \mathcal{P}$ of $c_i \in \mathcal{C}$
  Find $t_j \in \mathcal{T}$ with minimum weight among all uncovered targets
  Find $\langle c_m, p_n \rangle$ having the maximum rank $\mathcal{R}_{mn}$ among all the camera-pan pairs covering target t_j
  Suppose $\mathcal{T}'$ is the set of targets covered by $\langle c_m, p_n \rangle$
  $\mathcal{Z} = \mathcal{Z} \cup \{\langle c_m, p_n \rangle\}$
  $\mathcal{C} = \mathcal{C} \setminus \{c_m\}$
  $\mathcal{T} = \mathcal{T} \setminus \mathcal{T}'$
until $\mathcal{C} = \phi$ or $\mathcal{T} = \phi$ or no new $\langle c_i, p_k \rangle$ pair is found to be included in $\mathcal{Z}$

The priority assignment of GTOH is achieved by assigning a weight w_j to each target $t_j \in \mathcal{T}$ based on the number of cameras covering them, i.e.:

$$w_j = \begin{cases} 1 & \text{if target } t_j \text{ is coverable by only one camera} \\ N & \text{otherwise} \end{cases} \tag{4}$$

where N is the number of available cameras. A target may be covered by at most N cameras, so the worst possible weight is N. All the targets that are coverable by multiple cameras are treated equally and assigned the same weight N (i.e., the worst possible weight) regardless of the number of cameras covering them. After assigning weights to all targets, the camera-pans covering lonely targets are selected and oriented at once towards the pans covering the corresponding lonely targets. As soon as a camera is activated at a specific pan, that camera and the covered targets are removed from the corresponding sets. Before covering rest of the targets, all camera-pan pairs are ranked first. The rank of each pan p_k of camera c_i is denoted by $\mathcal{R}_{ik}$ and calculated using the following equation:

$$\mathcal{R}_{ik} = \sum_{j=1}^{M} \left\{ \frac{1}{w_j} : t_j \in F(\langle c_i, p_k \rangle) \right\} \quad i \in [1, N];\ k \in [1, Q]; \tag{5}$$

Recall that, $F(\langle c_i, p_k \rangle)$ is the set of targets located at pan p_k of camera c_i. After assigning the ranks to all camera-pan pairs, the highest-ranked pair $\langle c_m, p_n \rangle$ is selected and oriented. This process continues until all the targets are covered or all the cameras are activated and oriented at particular pans. Algorithm 1 summarizes the GTOH.

4.2. Purely Target Oriented Heuristic (PTOH)

The Purely Target Oriented Heuristic (PTOH) differs from GTOH in the weight assignment of targets and the camera selection criteria at each iteration. To understand the selection procedure, let us consider the camera-pan pair $\langle c_i, p_k \rangle$ for any $1 \le i \le N$ and $1 \le k \le Q$. PTOH starts by constructing a bipartition, $\mathbf{B}: \delta \to \mathcal{T}$, where δ is the set of camera-pan pair tuples that can cover at least one target and $\mathcal{T}$ is the set of targets to be covered. For any tuple $\langle c_i, p_k \rangle \in \delta$ and target $t_j \in \mathcal{T}$ there exists an edge e_{ijk} in the bipartition if t_j is coverable by camera c_i at pan p_k.

Next, we assign weight w_j to each target $t_j \in \mathcal{T}$ by counting the number of cameras that can cover t_j. This can be done by using TIS testing described in Section 2. Mathematically:

$$w_j = \sum_{i=1}^{N} \sum_{k=1}^{Q} a_{ik}^{j} \quad a_{ik}^{j} \in A_{N \times Q}^{M} \tag{6}$$

Recall that $A_{N \times Q}^{M}$ is a *binary coverage matrix* as defined in Section 2.

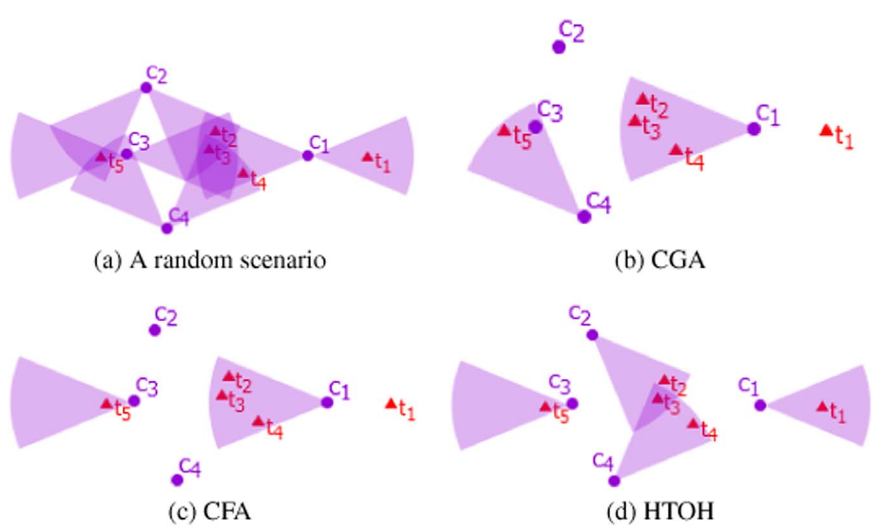


Fig. 3. Coverage comparison between CGA, CFA and HTOH.

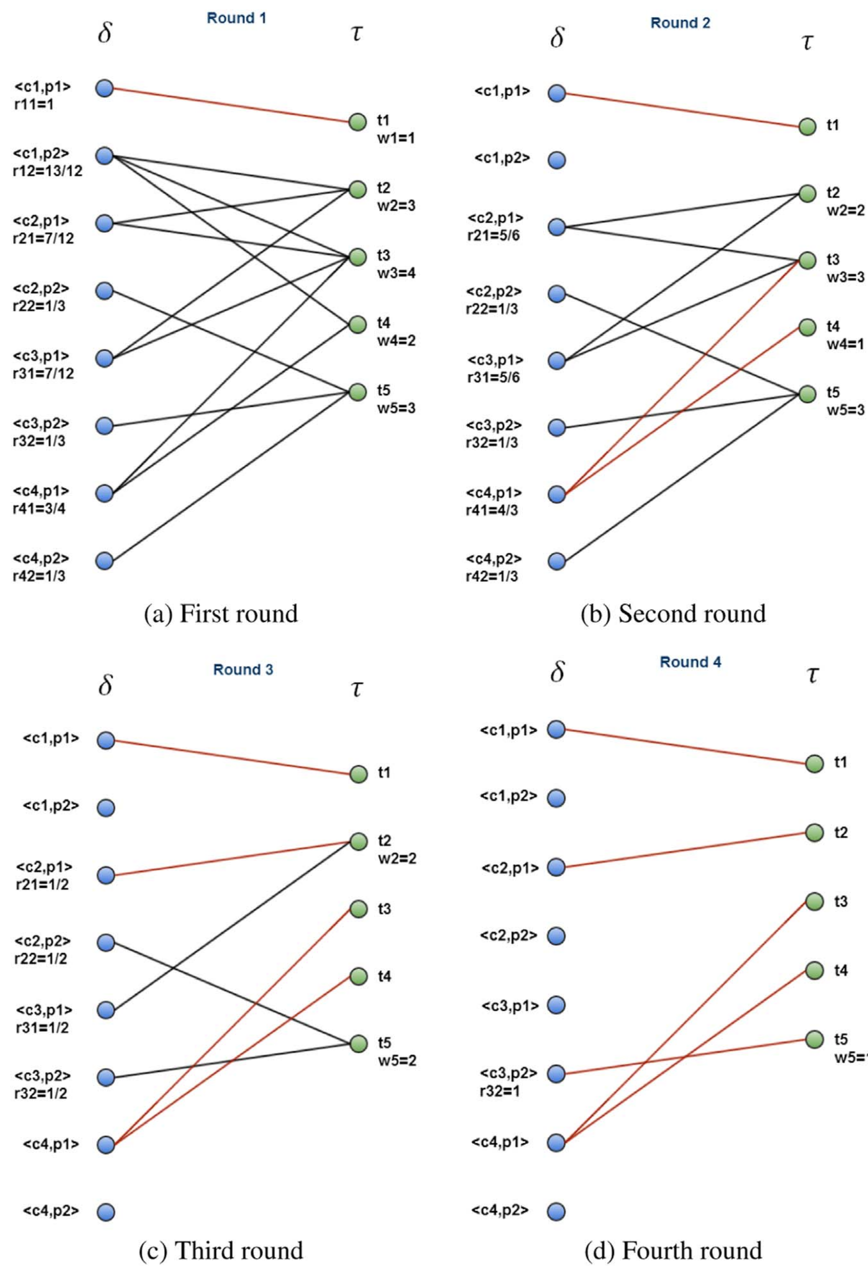


Fig. 4. HTOH algorithm snapshot and bipartite Graph Representation for Fig. 3(a).

After bipartition and weight assignment the actual camera selection tasks begin. At each iteration, we pick a target $t_{min} \in \mathcal{T}$ with minimum weight w_{min}. Then we look into camera-pan pair covering t_{min}. If there is only one such pair then we immediately select that camera and orient it to cover t_{min}. If t_{min} is coverable by more than one pairs, then we calculate a rank $\mathcal{R}_{ik}$ for each pair $\langle c_i, p_k \rangle$ covering t_{min}. The rank is calculated by taking the summation of inverse weight terms of each targets falling under its FoV, i.e.:

$$\mathcal{R}_{ik} = \sum_{j=1}^{M} \left\{ \frac{1}{w_j} : t_j \in F(\langle c_i, p_k \rangle) \right\} \quad i \in [1, N]; \; k \in [1, Q]; \tag{7}$$

where $F(\langle c_i, p_k \rangle)$ is the set of targets located at pan p_k of camera c_i. Note that while calculating ranks, targets covered by fewer cameras have lesser weights and contribute more as we take the inverse of weights. In this way, we cover *vulnerable* targets in earlier iterations with high probability.

After computing the ranks to all camera-pan pairs, the highest-ranked pair $\langle c_m, p_n \rangle$ is selected and oriented. Then we remove it from the bipartition **B** together with the corresponding edges and the other end of the edges in the bipartition (i.e., all covered targets). All weights and ranks are recomputed and the highest ranked pan-pair selection continues until all the targets are covered or all the cameras are activated and oriented at particular pans. Algorithm 2 summarizes the overall procedure.

4.3. Hybrid Target Oriented Heuristic (HTOH)

HTOH combines the ideas of both GTOH and PTOH. The weight assignment of targets and the rank calculation of camera-pan pairs are similar to PTOH. But the rule for selecting camera-pan pairs follows GTOH. More specifically, camera-pan pairs covering lonely targets are selected and oriented first. When no more lonely targets are found, camera-pan pair with the highest rank is chosen. The terminating condition is also the same. For weight assignments of the targets, Eq. (6) is applied instead of Eq. (4) of GTOH. Other than these two modifications, the algorithm of HTOH remains essentially the same as Algorithm 1 of GTOH.

4.4. Example scenario

Let us walk through the same example scenario of Fig. 1(a) to see how the target oriented approach HTOH provides more coverage with regard to other two heuristics CGA and CFA. For the convenience of the readers, Fig. 1(a) is repeated in Fig. 3(a). Recall that $F(\langle c_i, p_k\rangle)$ is a function that returns the set of targets that are coverable by camera c_i at pan p_k. Using this notation we may conclude the followings about Fig. 3(a):

$$F(\langle c_1, p_1\rangle) = \{t_1\};\ F(\langle c_1, p_2\rangle) = \{t_2, t_3, t_4\};F(\langle c_2, p_1\rangle) = \{t_2, t_3\};\ F(\langle c_2, p_2\rangle) = \{t_5\};F(\langle c_3, p_1\rangle) = \{t_2, t_3\};\ F(\langle c_3, p_2\rangle) = \{t_5\};F(\langle c_4, p_1\rangle) = \{t_3, t_4\};\ F(\langle c_4, p_2\rangle) = \{t_5\};$$

Fig. 3(b) shows the selection and orientation of cameras when we run CGA. Being greedy, CGA picks up camera c_1 and orients it towards pan p_2 because this camera-pan pair can provide highest number of target coverage, i.e., $|F(\langle c_1, p_2\rangle)|$ is maximum. By doing so, the target t_1 is ultimately left uncovered.

Fig. 3(c) shows the coverage by CFA. In the first iteration, the *force* of camera c_1 at pan p_2 is found to be maximum (=3/4) among all possible forces of camera-pan pairs (the force is calculated based on Eq. (3)). Again, by doing so, the target t_1 left uncovered at the end.

Finally, Fig. 3(d) shows the coverage of HTOH. Both CGA and CFA prioritizes the cameras first and thus CGA picks up a camera covering highest number of targets and CFA picks up a camera providing highest force. Using this camera-oriented strategy they leave the target t_1 uncovered. On the contrary, HTOH prioritizes the targets and covers the most vulnerable targets first. At the first iteration, HTOH covers all targets that are coverable by a single camera, i.e., the so-called *lonely targets*. By doing so the target t_1 readily gets covered at the first iteration whereas the other two heuristics failed to cover t_1. Thus HTOH covers more targets compared to CGA and CFA using the same number of cameras.

Fig. 4 explains the working mechanism of HTOH. At first we create a bipartition $\mathbf{B}$: $\delta \to \mathcal{T}$, where δ is the set of camera-pan pair tuples that can cover at least one target and $\mathcal{T}$ is the target set. An edge e_{ijk} exists in the graph if t_j is coverable by camera c_i at pan p_k.

Fig. 4(a) shows the first iteration of HTOH. The weights of all targets and the rank of each camera-pan pairs are calculated and shown in the figure. For instance, t_2 has a weight $w_3 = 3$ because it is coverable by three camera-pan pairs namely $\langle c_1, p_2\rangle$, $\langle c_2, p_1\rangle$, and $\langle c_3, p_1\rangle$. The rank of $\langle c_2, p_1\rangle$ was 13/12 because it covers three targets t_2, t_3, and t_4 with the weights, $w_2 = 3$, $w_3 = 4$, $w_4 = 2$ respectively and based on Eq. (7) its rank becomes:

$$\mathcal{R}_{21} = \frac{1}{3} + \frac{1}{4} + \frac{1}{2} = \frac{13}{12}$$

In the first round, target t_1 is covered as it is coverable by a single camera c_1 (at pan p_1), i.e., it is a lonely target. There is no more lonely targets. Thus the camera c_1 is selected and oriented towards pan p_1 in the first round. $\langle c_1, p_1\rangle$, t_1 and the edge e_{111} are removed and a new bipartition is created as shown in Fig. 4(b). In the second round, t_4 is the only lonely target coverable by c_4 (at pan p_1). Therefore, $\langle c_4, p_1\rangle$ is selected. $\langle c_4, p_1\rangle$, $\{t_3, t_4\}$ and the edges $\{e_{431}, e_{441}\}$ are removed. Note that t_4 gets removed because it is also covered by $\langle c_4, p_1\rangle$. In the third round, we are left with two targets t_2 and t_5 with equal weights $w_2 = w_5 = 2$. So we can randomly choose either of them. Suppose t_2 is chosen. Now we find camera-pan pairs that can cover t_2. There are two such pairs $\langle c_2, p_1\rangle$ and $\langle c_3, p_1\rangle$. Both has the same rank which is $\mathcal{R}_{21} = \mathcal{R}_{31} = 1$. Again we can randomly choose one of them. Suppose we select $\langle c_2, p_1\rangle$. So, $\langle c_2, p_1\rangle$, t_2 and the edge e_{221} are removed. Finally, in the fourth round we have a lonely target t_5 which can be covered by $\langle c_3, p_2\rangle$. So, it is selected at the final step. At this stage the algorithm terminates and the solution becomes $\delta' = \{\langle c_1, p_1\rangle, \langle c_4, p_1\rangle, \langle c_2, p_1\rangle, \langle c_3, p_2\rangle\}$. Thus, HTOH reaches at optimal solution in this example.

5. Performance bounds

In this section, we provide lower bounds of target coverage for the three heuristics CGA, CFA, and GTOH.

Suppose, we have M stationary targets that needs to be covered by N fixed cameras—all targets and cameras are located at positions known apriori. Suppose the target set is $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ where $|\mathcal{T}| = M$. Imagine an over-provisioned system where all targets are coverable by some camera(s), i.e., the Optimal Algorithm (OA) covers all the targets. Hence, the total number of targets covered by OA is $|\mathcal{T}| = M$. In the following theorem, we claim that the Centralized Greedy Algorithm (CGA) covers at least $M/2$ targets.

Theorem 5.1. *For the over - provisioned system where OA is able to cover all M targets, CGA can cover at least* $\frac{M}{2}$ *targets.*

Proof. Let n_g be the number of cameras selected and oriented by CGA. Let us number those cameras by $1, 2, 3, \ldots, n_g$ in the order of their selection and denote the set of (selected) cameras by $C_{CGA} = \{c_1, c_2, \ldots, c_{n_g}\}$.

Consider the i-th stage of CGA wherein CGA has already selected and oriented $i - 1$ cameras. Note that, the camera selected at the i-th stage of CGA is c_i. The *benefit* of selecting the camera c_i, denoted by $B(c_i)$, is the *additional* number of target points that c_i can cover once oriented by CGA. Thus, when we add the benefits of each stage of camera selection, we get the quantity, $\sum_{i=1}^{n_g} B(c_i)$, which is the number of target points covered by CGA when it finishes its execution. Therefore, it suffices to prove that:

$$\sum_{i=1}^{n_g} B(c_i) \geq \frac{M}{2}$$

Now suppose because of the selected orientation of c_i at i-th stage of CGA, a certain number of uncovered target points could not be covered at all by the selected (and oriented) cameras at the subsequent stages of CGA. Let us call this *lost* number of target points as *deficit* of c_i and denote it by $D(c_i)$. Thus, when we add the deficits of all stages of camera selections, we get the quantity, $\sum_{i=1}^{n_g} D(c_i)$, which is the total number of target points left uncovered when CGA finishes its execution. As the Optimal Algorithm (OA) can cover all the target points in $\mathcal{T}$, where $|\mathcal{T}| = M$, it is quite obvious that adding the *cumulative* benefits to the *cumulative* deficits yields the total number of target points M, i.e.,:

$$\sum_{i=1}^{n_g} B(c_i) + \sum_{i=1}^{n_g} D(c_i) = M \tag{8}$$

Let us consider the set of cameras selected by OA and denote it by C_{OA}. For the selected and oriented camera c_i at the i-th stage of CGA, the following three cases may exactly happen:

Case 1. $c_i \in C_{OA}$ *and the selected pan is the same in both CGA and OA.* If this is the case then CGA has made the right choice about c_i and no target points could be lost due to this camera selection. Thus, for this selection: $D(c_i) = 0$.

Case 2. $c_i \in C_{OA}$ *but the selected pan is different in CGA and OA.* Suppose, CGA has selected pan p_j and OA has selected pan p_k of the same camera c_i, where $p_j \neq p_k$. In such a case CGA covers $B(c_i)$

Table 1
Acronyms used in the plots.

Acronym	*Actual term*	*Reference to work*
CFA	Centralized Force-Directed Algorithm	(Munishwar and Abu-Ghazaleh, 2013)
CGA	Centralized greedy algorithm	(Ai and Abouzeid, 2006)
GTOH	Greedy Target Oriented Heuristic	This paper
HTOH	Hybrid Target Oriented Heuristic	This paper
OA	Optimum algorithm	(Ai and Abouzeid, 2006)
PTOH	Purely Target Oriented Heuristic	This paper

targets on pan p_j of c_i. The number of uncovered targets in pan p_k can not be more than $B(c_i)$, otherwise greedy would switch to this alternative pan p_k as (being greedy) it tries to cover maximum number of uncovered targets at each step. The worst case scenario happens when all the target points in pan p_j selected by CGA are covered by some other cameras in OA and none of the targets in pan p_k are covered by any other camera in OA. Thus, the wrong pan selection of c_i by CGA may lead to a loss of at most $B(c_i)$ targets relative to OA at this step. Therefore, for this selection:

$$D(c_i) \le B(c_i)$$

Case 3. $c_i \notin C_{OA}$. This may happen when the use of camera c_i becomes redundant for OA, i.e., all the targets that could be covered by c_i are already covered by some other camera(s) of OA. In that case no target points could be lost due to this decision of CGA as OA is not using c_i at all to cover any target point. So, for this selection:

$$D(c_i) = 0$$

For all of the above three cases we can see that, $0 \le D(c_i) \le B(c_i)$ holds for $\forall\ i = 1, 2, 3, \ldots n_g$. Therefore,

$$\sum_{i=1}^{n_g} D(c_i) \le \sum_{i=1}^{n_g} B(c_i) \tag{9}$$

Eq. (8) can be rearranged as follows:

$$\sum_{i=1}^{n_g} D(c_i) = M - \sum_{i=1}^{n_g} B(c_i) \tag{10}$$

Combining Eqs. (9) and (10), we get:

$$\sum_{i=1}^{n_g} B(c_i) \ge \frac{M}{2} \tag{11}$$

□

Next, we focus on Centralized Force-directed Algorithm (CFA) and see how it performs under over-provisioned systems. Let us define few terms first. For any camera, let us call a pan is a *feasible* pan if it contains at least one uncovered targets. A camera is called a camera with *lonely pan* if it has only one feasible pan. Now suppose, for a scenario, the set of cameras having lonely pan is the set $C_{LP} = \{c_1, c_2 \ldots c_{n_{lp}}\}$ and the total number of (unique) targets covered by selecting those cameras in C_{LP} is M', i.e.:

$$\sum_{i=1}^{n_{lp}} B(c_i) = M'$$

Then we can claim the following theorem to be true for over-provisioned systems:

Theorem 5.2. *For the over - provisioned system where OA is able to cover all M targets, the CFA can cover at least* $\frac{M+M'}{2}$ *targets.*

Proof. CFA differs with CGA primarily in the camera selection orders. Unlike CGA, which selects a camera that covers the most targets, CFA selects a camera that has maximum force towards a pan. For any camera, the force of a pan is the ratio of the number of uncovered targets located in that pan and the total number of uncovered targets in all the pans. Thus, CFA selects a pan (of the selected camera) that has the most number of uncovered targets compared to the other pans. Any camera having lonely pan always has the maximum force of 1 towards that pan and thus always gets selected by CFA. So, CFA selects all cameras with lonely pans first and guarantees to cover M' targets at first place (i.e., all cameras in $C_{\mathcal{LP}}$).

Once the cameras in $C_{\mathcal{LP}}$ are selected, CFA removes those cameras and the corresponding covered targets. After removal, we get a new scenario with $M - M'$ targets that needs to be covered by the rest of the cameras without having lonely pans. For the remaining cameras and targets, CFA selects the camera with maximum force first and configures it in a greedy way, i.e., selects a pan that covers maximum number of targets. Thus, from Theorem 5.1, we can claim that for the remaining targets, (being greedy) CFA will cover at least half of the $M - M'$ targets. Thus, when CFA finishes, the total number of covered targets becomes at least:

$$M' + \frac{M - M'}{2} = \frac{M + M'}{2}$$

□

Finally, we move on to target oriented heuristics (TOHs). Among three variations of TOHs we provide the lower bound of GTOH and left the other two as open problems. Let us define few terms first. A target is called a *lonely* target if it can be covered by only a *single* camera. Suppose, for a scenario, the set of lonely targets is the set $\mathcal{T}_{\mathcal{L}} = \{t_1, t_2 \ldots t_{n_l}\}$ where $|\mathcal{T}_{\mathcal{L}}| = M''$. Then we claim the following theorem:

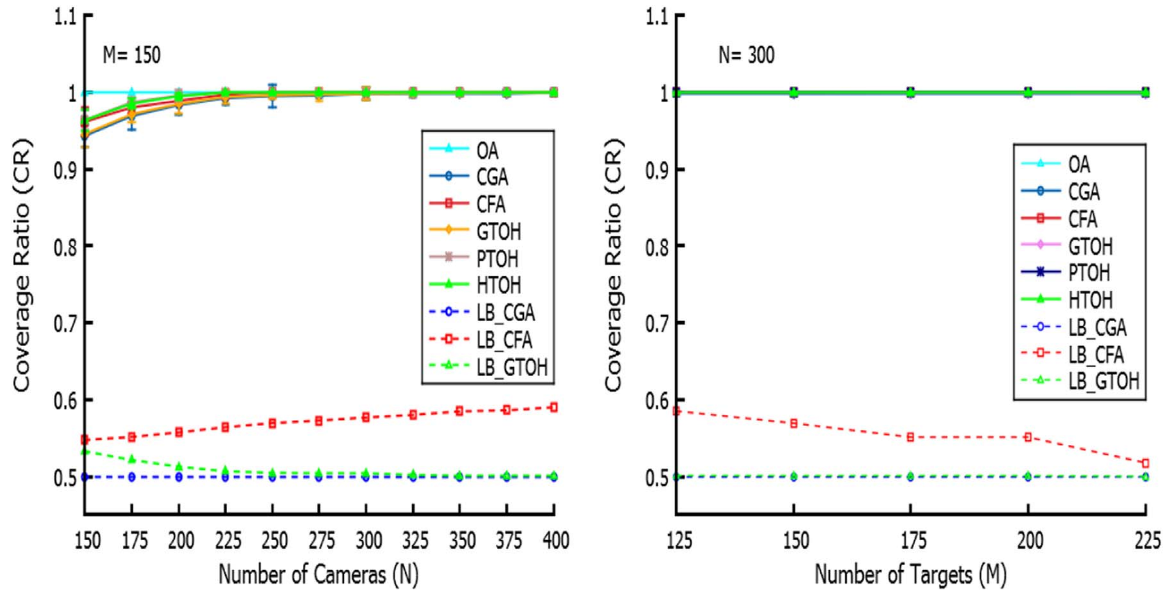
Theorem 5.3. *For the over - provisioned system where OA is able to cover all M targets, the Greedy Target Oriented Heuristic (GTOH) can cover at least* $\frac{M+M''}{2}$ *targets.*

Proof. The GTOH assigns a weight w_j=1 to each lonely target t_j and $w_j = N$ for every other target t_j that is covered by at least two cameras. Here N is the total number of available cameras. After weight assignment, the camera-pan pairs covering lonely targets are activated first and all lonely targets along with the cameras covering those are removed from the scenario. So after this step, M'' targets are guaranteed to be covered and we get a new scenario with $M - M''$ uncovered targets. For covering the rest of the targets, at first the rank $\mathcal{R}_{ik}$ of each camera-pan pair $\langle c_i, p_k \rangle$ is determined which is the summation of weights of all targets falling on pan p_k of camera c_i. The camera-pan pair $\langle c_i^*, p_k^* \rangle$ with the highest rank value is then selected. Let us assume that the highest rank value at this stage is $\mathcal{R}_{ik}^*$ and there are x_{ik} uncovered targets located on this (selected) camera-pan pair. At first we show that CGA will also choose the same camera-pan pair at this stage of GTOH. According to GTOH, each of the x_{ik} targets will have an equal weight of $\frac{1}{N}$. So the rank $\mathcal{R}_{ik}^* = \frac{x_{ik}}{N}$. If the rank of some other camera-pan pair covering y_{ik} uncovered targets is r_{ik}, then obviously $\mathcal{R}_{ik}^* > \mathcal{R}_{ik}$. Therefore,

$$\mathcal{R}_{ik}^* > \mathcal{R}_{ik} \Rightarrow \frac{x_{ik}}{N} > \frac{y_{ik}}{N} \Rightarrow x_{ik} > y_{ik}$$

.

Thus, at this stage GTOH eventually selects a camera-pan pair that covers maximum number of uncovered targets and its behavior becomes exactly similar to CGA. Therefore, according to



(a) (b)

Fig. 5. Coverage performance comparison of OA, CGA, CFA, GTOH, PTOH, and HTOH heuristics with 95% confidence interval and their associated achievable lower bounds (denoted by "LB_" in front).

Theorem 5.1, we can claim that for the remaining targets, (being greedy) GTOH will cover at least half of the $M - M''$ targets. Thus, when GTOH finishes, the total number of covered targets becomes at least:

$$M'' + \frac{M - M''}{2} = \frac{M + M''}{2}$$

□

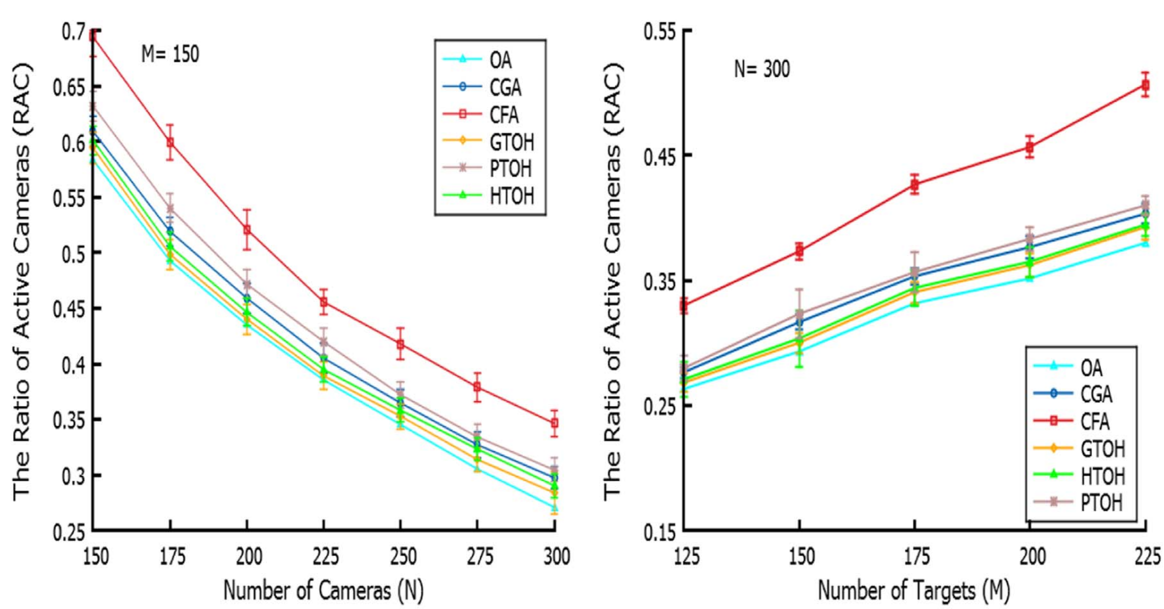
6. Performance evaluation

In this section we compare the performance of the target oriented heuristics (TOHs) with other approaches through extensive simulations.

6.1. Simulation environment

We consider a fixed area of $1000\,\text{m} \times 1000\,\text{m}$ unit square, where N cameras and M targets were randomly deployed. Position of cameras and targets are defined as 2-D points (x,y) in Cartesian co-ordinate system. The sensing range (R_s) is assumed to be 100 unit, and the Angle Of View (AoV) is set to $\theta = \frac{\pi}{4}$. With an AoV of $\frac{\pi}{4}$, the number of discrete pans (Q) becomes $Q = \frac{2\pi}{\frac{\pi}{4}} = 8$.

Scenario generation. We generate two types of static scenarios: (i) *camera-only* scenarios, and (ii) *target-only* scenarios. In order to generate camera-only scenarios, we vary the number of cameras over a fixed deployment area of $1000\,\text{m} \times 1000\,\text{m}$ square region. Number of cameras ranges from $N=150$ to $N=400$ in different scenarios with an increment of 25 cameras from one



(a) (b)

Fig. 6. Sensor Usage comparison for OA, CGA, CFA, GTOH, PTOH, and HTOH with 95% confidence interval.

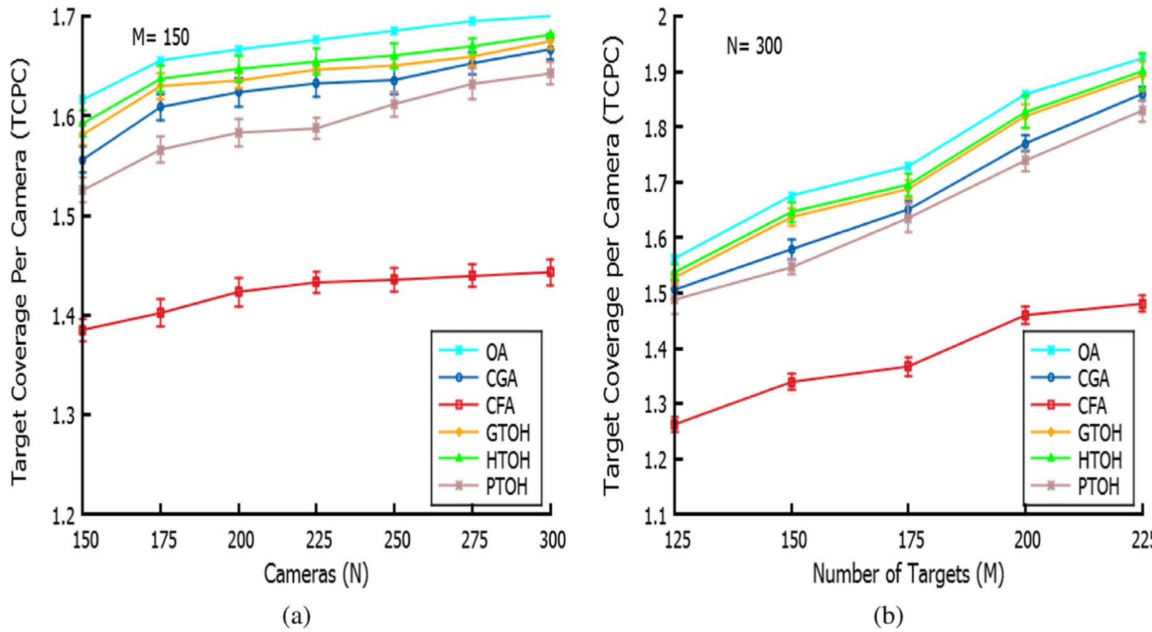


Fig. 7. Target Covered Per Camera (TCPC) comparison for OA, CGA, CFA, GTOH, PTOH, and HTOH with 95% confidence interval.

scenario to the other. As we increase the number of cameras, the larger set includes the cameras from the previous scenario. This ensures a consistent evaluation of the impact of the enlarged population of cameras by retaining all the “features” of the previous environment and simply making it better.

In order to generate target-only scenarios, 125–225 targets were randomly spread over a 1000 m $\times$ 1000 m square region with an increase of 25 targets at each step. Similar strategy of including targets from the smaller sets to the larger sets was adopted while generating the target-only scenarios.

Finally, we combine one camera-only scenario (for a certain N) with one target-only scenario (for a certain M) to create a smart camera network (SCN). Then the proposed heuristics along with the existing algorithms are simulated over these SCNs. For each SCN size (i.e., for a certain N and M), we have generated 4 instances. Performance measures are reported as an average of these 4 random SCNs (unless explicitly stated otherwise).

We have adopted two different approaches to capture the performance during simulation:–(i) by varying the number of cameras N while keeping the number of targets M fixed, and (ii) by varying the number of targets M while keeping the number of cameras N fixed.

6.2. Performance metrics

We analyze the performance of the different SCNs using five performance criteria: *coverage ratio* (CR), *ratio of active cameras* (RAC), *target coverage per camera* (TCPC), *power consumption* (PC), and *Number of Rounds* (NR). These performance metrics are defined as follows:

6.2.1. Coverage ratio (CR)

CR is the fraction of targets covered by the selected cameras after running an algorithm, i.e.:

$$CR = \frac{\text{Number of targets covered}}{\text{Total number of targets}} \tag{12}$$

As we are assuming that the Optimal Algorithm (OA) covers all the targets, the metric CR indicates the proximity of performance of an algorithm with respect to the optimal solution. The CR of the Optimal Algorithm is always 1. For any other algorithm, the lower value of CR indicates poorer performance w.r.t. OA in terms of target coverage.

6.2.2. Ratio of active cameras (RAC)

RAC is the fraction of cameras selected by an algorithm:

$$RAC = \frac{\text{Number of cameras activated}}{\text{Total number of cameras}} \tag{13}$$

While CR is a measure of target coverage, RAC is a measure of camera usage. The *smaller* the ratio, the better is the performance.

6.2.3. Target coverage per camera (TCPC)

TCPC is the ratio of the number of covered targets and the number of cameras activated, i.e.:

$$TCPC = \frac{\text{Number of targets covered}}{\text{Number of cameras used}} \tag{14}$$

Notably, TCPC provides average number of targets covered per camera. This is an important metric because a particular algorithm may achieve higher RAC value activating a large number of cameras or lower CR value without covering all the targets. Thus, the better performance of one metric may not reflect the poorer performance of the other metric. Considering this fact, TCPC is introduced which combines the effect of both camera usage and target coverage (i.e., considers CR and RAC simultaneously).

6.2.4. Number of rounds (NR)

NR is the number of iterations taken by an algorithm to cover all the targets or to reach the state when no more targets can be covered due to unavailability of either feasible cameras or coverable targets.

6.3. Results

In this section we provide the performance comparison graphs for each of the proposed target oriented heuristics along with the optimal algorithm (OA) solving the ILP formulation developed in (Ai and Abouzeid, 2006) and two other sensor-oriented algorithms

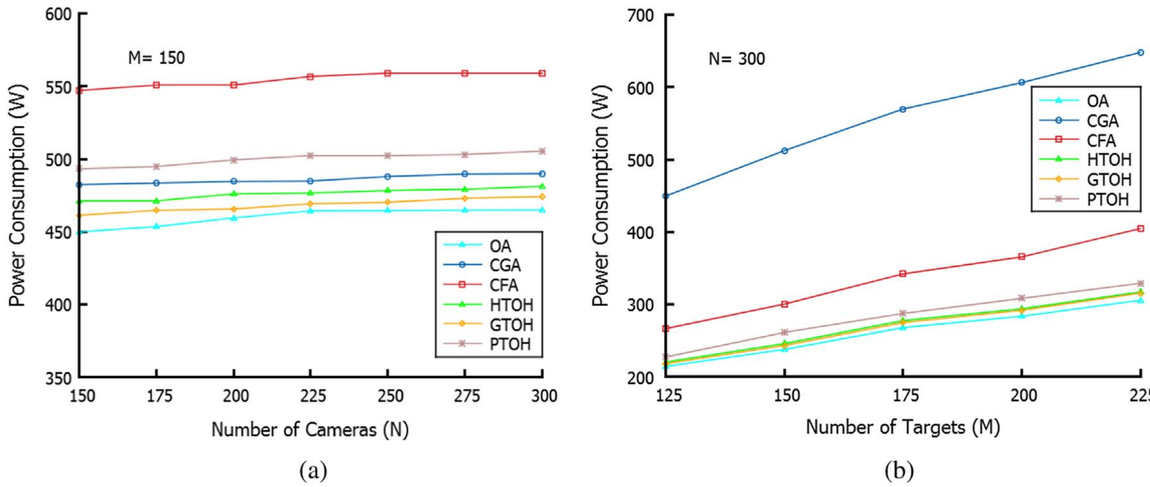


Fig. 8. Power Consumption comparison for OA, CGA, CFA, GTOH, PTOH, and HTOH.

from the literature namely *Centralized Greedy Algorithm* (CGA) (Ai and Abouzeid, 2006) and *Centralized Force-directed Algorithm* (CFA) (Munishwar and Abu-Ghazaleh, 2013). Table 1 summarizes all the acronyms used in the plots.

6.3.1. Target coverage

At first, we measure the coverage ratio by keeping the number of targets fixed at $M=150$ and varying the number of cameras N from 150 *to* 400. The result is shown in Fig. 5(a). All the algorithms converge to the Optimal Algorithm (OA) when number of cameras becomes high enough to cover all 150 targets, i.e., the system moves into an over-provisioned system. This happens when the number of cameras crosses 200.

Next we plot CR in Fig. 5(b) by keeping the number of cameras fixed at $N=300$ and varying the number of targets M from 125 to 225. As the number of cameras is lot more than the number of targets, clearly the system is an over-provisioned system and all approaches have enough camera to cover all the targets and thereby nearly converge to the optimal coverage.

For each of the curves, we also show the 95% confidence interval as small bars in the plots. In Fig. 5(a) and (b) the confidence interval decreases to zero as soon as all of the algorithms converge to the Optimal Algorithm (OA).

In both figures, we have also added the theoretical lower bounds of CGA (Ai and Abouzeid, 2006), CFA (Munishwar and Abu-Ghazaleh, 2013), and GTOH derived in Section 5. For greedy algorithm the bound is $\frac{1}{2}$. The lower bound of GTOH is higher for fewer number of cameras because the number of lonely targets is more with fewer cameras and the theoretical lower bound of GTOH, as derived, linearly increases with the increase in number of lonely targets (see Theorem 5.3).

For large number of cameras the number of lonely targets becomes very small (near to zero) and the lower bound of GTOH converges to CGA (Ai and Abouzeid, 2006). As opposed to CGA (Ai and Abouzeid, 2006) and GTOH, the theoretical lower bound of CFA (Munishwar and Abu-Ghazaleh, 2013) is always better. The reason is, in over-provisioned system, a rich proportion of cameras are cameras with lonely pan and the theoretical lower bound of

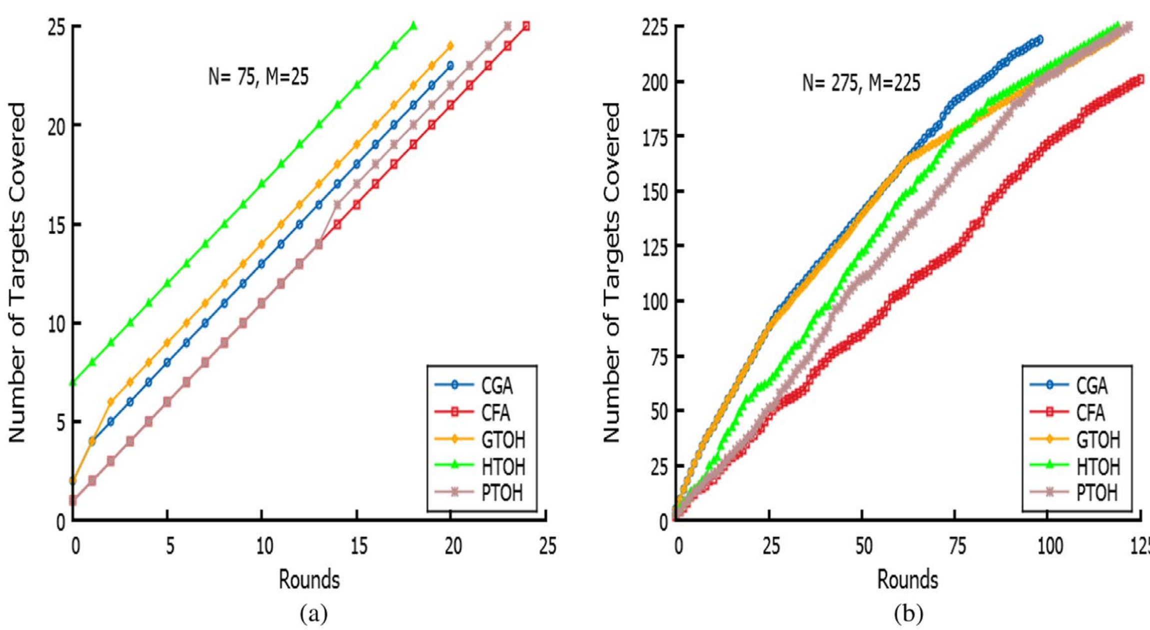


Fig. 9. Number of rounds comparison for CGA, CFA, GTOH, PTOH, and HTOH.

CFA linearly increases with the number of cameras having lonely pans (cf. Theorem 5.2).

In most of the cases, the target oriented approach PTOH covers more targets compared to CGA (Ai and Abouzeid, 2006) or CFA (Munishwar and Abu-Ghazaleh, 2013) because in an over-provisioned system the number of cameras is sufficient to cover all the targets. When enough cameras are available it is always beneficial to cover the targets coverable by fewer cameras like PTOH, otherwise there is a chance that the cameras that could cover them might get used up in the earlier iterations and these targets are left uncovered.

6.3.2. Camera usage

The ratio of active camera (RAC) values are shown in Fig. 6 where we vary both the number of cameras and the number of targets. In Fig. 6(a) the number of targets M is kept fixed at 150 and the number of cameras N varies from 150 to 300. As we increase N the system approaches to an over-provisioned system. In terms of camera usage, GTOH performs close to optimal as it greedily activates cameras but does not cover all of the targets. Next better performance is given by HTOH. PTOH uses more cameras than HTOH as it covers all the targets by activating more cameras. Between CGA (Ai and Abouzeid, 2006) and CFA (Munishwar and Abu-Ghazaleh, 2013), CGA uses less cameras than CFA. CFA's camera usage is notably highest among all heuristics.

Fig. 6(b) shows the RAC values when we keep the number of camera N fixed at 300 and vary the number of targets from 125 to 225. When we increase the number of targets while keeping the number of cameras fixed the system becomes less over-provisioned and the ratio of active cameras linearly increases. The performance comparison of different heuristics follow the similar trend of Fig. 6(a).

Similar to CR figures, we include 95% confidence interval for sensor usage curves (i.e., Fig. 6(a) and (b)) as well. These figures also show the similar trends that the confidence interval band of each of the curves decreases with the gradual increase of either sensors or targets.

6.3.3. Target coverage per camera usage

Ultimately, the target oriented heuristics (HTOH and GTOH) turn up the clear winners when we look at Fig. 7(a) and (b). From both of the figures, it is clearly evident that for different number of cameras, TCPC of HTOH is convergent to the optimal as it provides higher coverage using fewer cameras. Between GTOH and HTOH, GTOH activates less cameras as its RAC value is better as shown in Fig. 6(a) but it has poorer coverage w.r.t. HTOH as shown in Fig. 5(a). When we scale the coverage measure to camera usage measure, i.e., consider *coverage/camera usage* as the metric of interest (i.e., the TCPC), the increase in coverage per camera of HTOH over GTOH is about 7%. Both HTOH and GTOH outperform CGA (Ai and Abouzeid, 2006). Performance of PTOH is not as good as HTOH and GTOH but it is better than CFA (Munishwar and Abu-Ghazaleh, 2013). From the figure we can see that CFA's performance is worst among all heuristics as it uses more cameras only to cover fewer targets.

The plots also show 95% confidence interval bands on each curve.

6.3.4. Power consumption

Next, we consider the power consumption of each algorithm. For measuring power consumptions we use the same power consumption model proposed by Farzana et al. (2016). According to this model, a visual sensor can be in any of the three states: *active*, *idle*, and *sleep*.

(1) **Active:** In this state, the visual sensor is powered up and it either monitors a target or transmits the signals/gathered information to some desired receiver. This is the most power consuming state.
(2) **Idle:** In this state, though the sensor is powered up, it is not participating in the monitoring or transmitting operations. Still it requires some power to run the background OS processes.
(3) **Sleep:** In this state, the sensor is powered down and deactivates all of its hardware component. This results in the least power consuming state.

In our proposed heuristics (Malek et al., 2016), we assume that all the active sensors continue to monitor targets until their battery is completely drained off. All other remaining sensors are in sleep state which might become active on demand to replace any of the drained sensor(s). Considering these two factors, the total power consumption P_t is calculated as follows:

$$P_t = (\text{Number of active sensors} \times P_a) + (\text{Number of inactive sensors} \times P_s)$$

where, P_a is the power consumption of a sensor in active state and P_s is the power consumption of the sensor in sleep state. For a Panoptes video sensor (Feng et al., 2005), the value of P_a and P_s is 5.268 W and 0.058 W respectively. We use these values for plotting the power consumption curves of all algorithms.

The power consumption for all the considered algorithms are shown in Fig. 8(a), where we keep the number of targets fixed and vary the number of sensors and in Fig. 8(b), where we keep the number of sensors fixed and vary the number of targets. In Fig. 8(a), we have kept number of targets fixed at 150 and the sensors are gradually increased from 150 to 300. Thus the system moves from under-provisioned system to over-provisioned because initially there do not have enough cameras to cover all the targets. With the addition of new cameras, more cameras become activated because all algorithms try to maximize the target coverage. Therefore, the power consumption naturally increases with the addition of new cameras. After some point when the system moves to fully over-provisioned system, the power consumption remains almost constant because at that point all targets are already covered and no more cameras get activated as all of the algorithms (except CFA) also try to minimize camera usage. So, the newly added cameras remain unutilized at sleep state incurring a little increase in the total power consumption due to their small power consumptions at the sleep state.

In Fig. 8(b), we keep the number of cameras fixed at 300 and the number of targets are gradually increased from 125 to 225. As the number of targets are increasing the system approaches from over-provisioned to under-provisioned system. Initially there are plenty of cameras to cover the targets. Gradually the system adds more targets in the network, and more and more inactive cameras become activated which results in the linear increase in the power consumption curves. Also the power consumption curves follow the similar trend of the sensor usage curves shown in Fig. 6(b).

Notably, for both of the power consumption plots in Fig. 8(a) and (b), the proposed target oriented heuristics (GTOH and HTOH) outperform the existing algorithms CGA and CFA.

6.3.5. Number of rounds

Finally, we plot the number of rounds (NR) for all the target oriented heuristics (TOHs) and compare their performance with CGA (Ai and Abouzeid, 2006) and CFA (Munishwar and Abu-Ghazaleh, 2013). In Fig. 9(a), we plot the number of rounds (NR) needed to complete the simulation on a particular scenario with

$N=25$ and $M=25$ for all three TOHs, CGA, and CFA. Next in Fig. 9 (b), we plot NR for another scenario where number of targets is =225 and number of cameras is =275.

In both figures, HTOH outperforms all others by achieving higher number of target coverage on average at each round and by converging to the Optimal Algorithm (OA). In the next, GTOH performs better than PTOH. PTOH takes higher NR values than both HTOH and GTOH. Between CGA and CFA, CFA always takes notably higher number of rounds than CGA and all other TOHs. In comparison of CGA with TOHs, CGA performs better than only PTOH by taking less number of rounds and covering higher number of targets on average in each round. So, in case of number of rounds, the target oriented heuristics HTOH and GTOH converge to the optimal solution faster than both CGA and CFA.

7. Conclusion

We have presented a target oriented approach to coverage problem of Visual Sensor Networks (VSNs) that provides near-optimal performance to achieve maximum target coverage using minimum number of sensors. We have provided elaborate description of the proposed heuristics and theoretically determined its performance bounds in terms of coverage. We have also presented similar bounds for two other exiting heuristics. From the theoretical bounds, it is clearly evident that the performance of target oriented heuristics improves when there exist a rich proportion of lonely targets in the network that are coverable by only a single camera. Then we have presented the performance evaluation of our heuristics varying the number of targets while keeping the number of sensors fixed and vice versa. We have simulated performance of TOHs, CGA, and CFA considering the optimal solution as benchmark and taking Coverage Ratio (CR), Ratio of Active Cameras (RAC), Target Covered Per Camera (TCPC), and finally, Number of Rounds (NR) as metrics. The results show that TOHs covers more targets attaining higher CR, activating less number of sensors scoring lower RAC, and converges to the Optimal Algorithm (OA) much faster in terms of NR. It is also noticeable that TOHs achieve higher TCPC ratio in over-provisioned systems which validates our claim that TOH is a better solution for such systems.

There still remain some open problems to answer. For this study, we limit ourselves to *pan-only* cameras i.e., we allowed only horizontal movements of the cameras. Orientations in other dimensions (like tilt and zoom) are immediate possibilities. We only consider uniform distribution of the cameras and targets. It will be interesting to see how the proposed heuristics perform under other kinds of node distributions.

Acknowledgment

This work was done as a part of B.Sc. thesis completed at the Department of Computer Science & Engineering (CSE), Bangladesh University of Engineering & Technology (BUET) (Zannat et al., 2014), and received *the best undergraduate thesis award*. The authors would like to thank anonymous reviewers for providing valuable feed backs at the time of thesis evaluation.

References

Adriaens, J., Megerian, S., Potkonjak, M., 2006. Optimal worst-case coverage of directional field-of-view sensor networks. In: Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 , Vol. 1, IEEE, pp. 336–345.

Ai, J., Abouzeid, A.A., 2006. Coverage by directional sensors in randomly deployed wireless sensor networks. J. Comb. Optim. 11 (1), 21–41.

Akyildiz, I.F., Melodia, T., Chowdhury, K.R., 2007. A survey on wireless multimedia sensor networks. Comput. Netw. 51 (4), 921–960.

Cai, Y., Lou, W., Li, M., Li, M., 2007. Target-oriented scheduling in directional sensor networks. In: IEEE INFOCOM, IEEE, pp. 1550–1558.

Cardei, M., Du, D.-Z., 2005. Improving wireless sensor network lifetime through power aware organization. Wirel. Netw. 11 (3), 333–340.

Chakrabarty, K., Iyengar, S.S., Qi, H., Cho, E., 2002. Grid coverage for surveillance and target location in distributed sensor networks. Comput. IEEE Trans. 51 (12), 1448–1453.

Costa, D.G., Silva, I., Guedes, L.A., Portugal, P., Vasques F., 2014. Enhancing redundancy in wireless visual sensor networks for target coverage. In: Proceedings of the 20th Brazilian Symposium on Multimedia and the Web, ACM, pp. 31–38.

Dai, R., Akyildiz, I.F., 2009. A spatial correlation model for visual information in wireless multimedia sensor networks. IEEE Trans. Multimed. 11 (6), 1148–1159.

Dieber, B., Micheloni, C., Rinner, B., 2011. Resource-aware coverage and task assignment in visual sensor networks. IEEE Trans. Circuits Syst. Video Technol. 21 (10), 1424–1437.

Farzana, S., Papry, K.A., Rahman, A., Rab, R., 2016. Maximally pair-wise disjoint set covers for directional sensors in visual sensor networks. In: 2016 Wireless Days (WD), IEEE, pp. 1–7.

Feng, W.C., Kaiser, E., Feng, W.C., Baillif, M.L., 2005. Panoptes: scalable low-power video sensor networking technologies. ACM Trans. Multimed. Comput. Commun. Appl. 1 (2), 151–167.

Fusco, G., Gupta, H., 2009. Selection and orientation of directional sensors for coverage maximization. In: Proceedings of the 6th Annual IEEE Communications Society Conference on, IEEE Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON'09, pp. 1–9.

Gupta, H., Zhou, Z., Das, S.R., Gu, Q., 2006. Connected sensor cover: self-organization of sensor networks for efficient query execution. Netw. IEEE/ACM Trans. 14 (1), 55–67.

Hörster, E., 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, pp. 111–120.

Hatanaka, T., Wasa, Y., Funada, R., Charalambides, A.G., Fujita, M., 2016. A payoff-based learning approach to cooperative environmental monitoring for ptz visual sensor networks. IEEE Trans. Autom. Control. 61 (3), 709–724. http://dx.doi.org/10.1109/TAC.2015.2450611.

Johnson, M.P., Bar-Noy, A., 2011. Pan and scan: configuring cameras for coverage. In: INFOCOM, 2011 Proceedings IEEE, IEEE, pp. 1071–1079.

Liu, J., Sridharan, S., Fookes, C., 2016. Recent advances in camera planning for large area surveillance: a comprehensive review. ACM Comput. Surv. 49 (1), 6:1–6:37. http://dx.doi.org/10.1145/2906148.

Lu, S., Yu, S., 2014. A fuzzy k-coverage approach for RFID network planning using plant growth simulation algorithm. J. Netw. Comput. Appl. 39, 280–291. http://dx.doi.org/10.1016/j.jnca.2013.07.015.

Malek, S.M.B., Sadik, M.M., Rahman, A., 2016. On balanced k-coverage in visual sensor networks. J. Netw. Comp. Appl. . http://dx.doi.org/10.1016/j.jnca.2016.06.011, ISSN: 1084-8045.

Munishwar, V.P., Abu-Ghazaleh, N.B., 2011. Target-oriented coverage maximization in visual sensor networks. In: Proceedings of the 9th ACM International Symposium on Mobility Management and Wireless Access, ACM, pp. 175–178.

Munishwar, V.P., Abu-Ghazaleh, N.B., 2013. Coverage algorithms for visual sensor networks. ACM Trans. Sens. Netw. 9 (4), 45.

Neishaboori, A., Saeed, A., Harras, K., Mohamed, A., Low complexity target coverage heuristics using mobile cameras. In: Proceedings of the IEEE 11th International Conference on, IEEE Mobile Ad Hoc and Sensor Systems (MASS), 2014, pp. 217–221.

Rashid, B., Rehmani, M.H., 2016. Applications of wireless sensor networks for urban areas: a survey. J. Netw. Comput. Appl. 60, 192–219.

Shakkottai, S., Srikant, R., Shroff, N.B., 2005. Unreliable sensor grids: coverage, connectivity and diameter. Ad Hoc Netw. 3 (6), 702–716.

Slijepcevic, S. Potkonjak, M., 2001. Power efficient organization of wireless sensor networks. In: Communications, 2001. ICC 2001. IEEE International Conference on, Vol. 2, IEEE, pp. 472–476.

Soro, S., Heinzelman, W., 2009. A survey of visual sensor networks. Adv. Multimed.

Tian, D., Georganas, N.D., 2002. A coverage-preserving node scheduling scheme for large wireless sensor networks. In: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, ACM, pp. 32–41.

Wang, P., Dai, R., Akyildiz, I.F., 2011. Visual correlation-based image gathering for wireless multimedia sensor networks. In: INFOCOM, 2011 Proceedings IEEE, IEEE, pp. 2489–2497.

Yap, F.G., Yen, H.-H. Novel visual sensor deployment algorithm in occluded wireless visual sensor networks. IEEE Systems Journal PP (99).

Yap, F.G., Yen, H.-H., 2014. A survey on sensor coverage and visual data capturing/processing/transmission in wireless visual sensor networks. Sensors 14 (2), 3506–3527.

Ye, F., Zhong, G., Cheng, J., Lu, S., 2003. L. Zhang, Peas: A robust energy conserving protocol for long-lived sensor networks. In: Proceedings of the 23rd International Conference on, IEEE Distributed Computing Systems, 2003, pp. 28–37.

Zannat, H., Akter, T., Tasnim, M., 2014. The Coverage Problem in Visual Sensor Networks: A Target Oriented Approach, a thesis submitted to the department of Computer Science & Engineering (CSE), Bangladesh University of Engineering & Technology (BUET), Dhaka, Bangladesh as a partial fulfillment of the B.Sc. Engg. degree, September, Received best undergraduate thesis award.

Hafsa Zannat received her B.Sc. Engg. degree in Computer Science and Engineering from the Department of Computer Science and Engineering (CSE), Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in 2014. She is currently working towards the M.Sc. Engg. degree in CSE at BUET. Her research interests include Wireless Sensor Networks, Mobile Computing and Cloud Computing.

Taslima Akter received her B.Sc. Engg. degree in Computer Science and Engineering from the Department of Computer Science and Engineering (CSE), Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in 2014. Her research interests include wireless and sensor networks, mobile computing, health informatics and human computer interaction. She is currently working as a Lecturer in the Department of Computer Science and Engineering (CSE), University of Asia Pacific, Dhaka, Bangladesh.

Mashrura Tasnim received her B.Sc. Engg. degree in Computer Science and Engineering from the Department of Computer Science and Engineering (CSE), Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in 2014. Her research interests include human computer interaction, data mining, artificial intelligence and visual sensor network. She is currently working as a Lecturer in the Department of Computer Science and Engineering (CSE), Eastern University, Dhaka, Bangladesh.

Ashikur Rahman received his B.Sc. and M.Sc. degrees in Computer Science and Engineering from the Department of Computer Science and Engineering (CSE), Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in 1998 and 2001, respectively. He received his Ph.D. in 2006 from the Department of Computing Science, University of Alberta, Canada. After finishing his PhD, he worked as a postdoctoral researcher at the Simon Fraser University (2007), University of Calgary (2010–2011), McGill University (2011), Canada and Binghamton University (2012), USA. His research interests include ad-hoc and sensor networks, peer-to-peer computing, swarm intelligence, data center networks, cyber physical systems, back-end compiler optimization and neural networks. He is currently working as a Professor in the Department of Computer Science and Engineering (CSE), BUET.