Coverage in Visual Sensor Networks with Pan-Tilt-Zoom Cameras: the MaxFoV Problem

Vikram P. Munishwar
State University of New York at Binghamton
Email: vmunish1@cs.binghamton.edu

Vinay Kolar
IBM Research, India
Email: vinkolar@in.ibm.com

Nael B. Abu-Ghazaleh
State University of New York at Binghamton
Email: nael@cs.binghamton.edu

***Abstract*—We consider the problem of target coverage in visual sensor networks with Pan-Tilt-Zoom (PTZ) cameras. The finely controllable movement in PTZ dimensions creates a large number of possible Field-of-View (FoV) settings, making it prohibitively expensive to consider them all in coverage algorithms. However, these FoVs are redundant as each group of targets is generally covered by many FoVs. Thus, an important problem is how to identify FoVs that cover all maximal subsets of targets (MaxFoV) efficiently. We show that MaxFoV is an instance of generating all maximal cliques, which is NP-hard in general but polynomial if the number of cliques is polynomial. We construct an optimal algorithm to solve the problem with a worst case complexity of $O(n^3)$. Simulation and testbed experiments show that the algorithm drastically reduces the number of FoVs allowing multi-camera coverage to scale without sacrificing coverage quality.**

I. Introduction

Smart camera networks, also called Visual Sensor Networks (VSNs) [19], are used in a range of applications such as monitoring and surveillance, traffic management and health care [3], [14]. Coverage is a key problem in VSNs: how can cameras select their Field-of-Views (FoVs) to increase the amount and quality of information collected by the network. Coverage algorithms require identification of a set of feasible FoVs for each camera. Out of the feasible FoVs, coverage algorithms assign one FoV to each camera. The complexity of coverage algorithms increases significantly as a function of the number of feasible FoVs. Thus, it is critical to reduce the number of feasible FoVs without affecting coverage quality.

We focus on the problem of target coverage in VSNs with Pan-Tilt-Zoom (PTZ) cameras [18]. PTZ cameras can change their Field-of-View (FoV) in three dimensions: horizontal (pan), vertical (tilt), and depth (zoom). In VSNs, coverage has typically been considered for directional sensors (or pan-only cameras) [7], [10], [21] where each sensor can orient (pan) to cover k different sectors. The coverage problem assigns each camera to one sector to meet the coverage objective; for instance, to maximize the total targets covered [7], [8], [16], or to schedule cameras to allocate their time among different targets [11]. While PTZ cameras are able to provide superior coverage to pan-only cameras (or directional sensors), they introduce new challenges.

In coverage algorithms for directional sensors [2], [7], [8], [10], [16], [21], all discrete FoVs (sectors corresponding to the pan directions) are considered by coverage algorithms. We call this algorithm for generating feasible FoVs per camera the *DISCRETE algorithm*. For PTZ cameras, DISCRETE leads to a prohibitively high number of FoVs. The Axis-213 camera [4] at the finest control steps can be configured into millions of different FoVs. If we reduce the number of FoVs by increasing the step with which the FoV is varied in the three axes, this leads to missing FoVs that lie between the chosen steps, causing suboptimal coverage.

Reducing the number of FoVs considered by the multi-camera coverage algorithms is critical to their efficiency and scalability. The multi-camera target coverage problem is known to be NP-hard [2]; it is an assignment problem of one FoV to each camera with the goal of maximizing the overall number of covered targets. A high number of feasible FoVs, representing the possible camera assignments in coverage algorithms, significantly increases the complexity of coverage algorithms. With millions of FoVs for each camera, coverage algorithms become computationally infeasible, especially in mobile scenarios where coverage decisions must be continuously made in real-time [17].

In this paper, we define a new problem of deriving the optimal set of FoVs to be considered by PTZ cameras. The performance of PTZ coverage algorithms can be substantially increased by reducing the number of candidate FoVs for each camera. Rather than focusing on camera parameters, we focus on the groups of targets covered; only a small subset of FoVs need to be considered as some FoVs cover the same group of targets or a subset of targets covered by some other FoVs. Supporting coverage optimality requires that every subset of targets that is jointly coverable appears in at least one FoV in the solution set. Thus, our goal is to identify a minimal set of FoVs while supporting coverage-optimality. We formally define this problem, which we call the *Maximal FoV Identification Problem (MaxFoV)*, in Section II. We believe that MaxFoV is a new and fundamental problem that is necessary for target coverage in PTZ cameras, allowing practical solutions to this important class of sensor networks.

We show that MaxFoV under some conditions is an instance of the maximal cliques generation problem, which is known to be NP-hard for general graphs [9]. However, the problem is known to have polynomial time solutions for special graphs, where the number of maximal cliques is polynomial [20]. We show that for coverage graphs that arise in our problem the number of maximal cliques is in fact polynomial due to the geometric properties of the problem. Furthermore, we develop a polynomial time algorithm, called the Geometric Extreme FoV algorithm (G-EFA), and prove its optimality. G-EFA has cubic runtime as a function of the number of targets in range of the camera in the worst case.

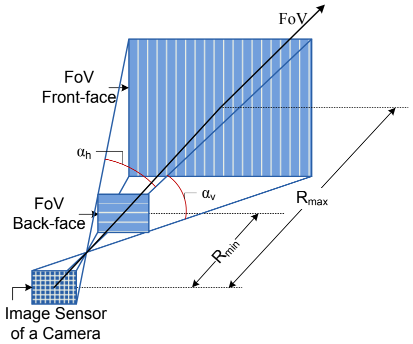


Fig. 1. Field-of-view of a camera.

We evaluate the algorithm both in simulation and on a small multi-camera testbed. Our results indicate that, compared to DISCRETE with 1° of step size, G-EFA generates 53x fewer FoVs per-camera in testbed scenarios with 20 targets, and up to 5000x fewer FoVs per-camera in simulated scenarios with 100 targets. Furthermore, DISCRETE fails to allow multi-camera coverage algorithms to reach optimality unless the finest camera control steps are used (proportionately increasing the number of candidate FoVs).

The primary contributions of the paper are:

1. We identify a new problem, *MaxFoV*, whose goal is to find unique maximal-FoVs for a PTZ camera. MaxFoV is an essential first step in solving target-oriented coverage problems for PTZ cameras; it provides coverage algorithms with a minimal but coverage-optimal set of FoVs, reducing their complexity while not limiting coverage of multi-camera algorithms.

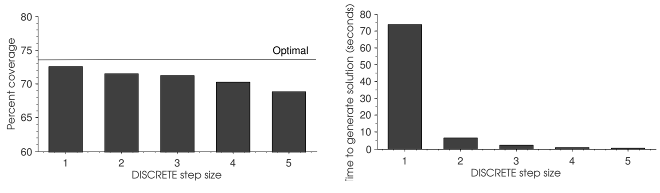
2. We prove (in Theorem 1) that MaxFoV under some conditions is an instance of the problem of finding all maximal cliques in a graph, which is known to be NP-hard for general graphs [9], [13]. However, the complexity can be polynomial if the number of maximal cliques is polynomial, which we show in the case for MaxFoV.

3. We present a polynomial time algorithm to solve MaxFoV, *Geometric Extreme FoV Algorithm (G-EFA)* that has $O(n^3)$ complexity in the worst case. G-EFA is initially formulated based on the process used to derive the number of maximal cliques in step 2, which is then optimized using geometric constraints to further improve performance. G-EFA is presented in Section V.

4. We demonstrate – both in simulation and testbed – that G-EFA substantially improves the efficiency of multi-camera coverage algorithms. G-EFA efficiently generates a *minimal* coverage-optimal set of FoVs whose cardinality is orders of magnitude smaller than that produced by DISCRETE. MaxFoV enables dramatic reduction in the complexity of multi-camera coverage algorithms while not limiting the quality of the coverage. Evaluation results are presented in Section VI.

II. Motivation and Problem Definition

In this section, we first describe the FoV of a PTZ camera, and then motivate and formally present the MaxFoV problem.



(a)

(b)

Fig. 2. (a) Impact of varying step sizes of DISCRETE on maximum coverage; (b) Time required to generate the multi-camera coverage solution.

A. Camera FoV and PTZ Configurations

A Field-of-View (FoV) of a camera represents the extent of its visible field. Typically, an FoV is represented by its *angle-of-view* (α) and *depth-of-field* (R). The Angle-of-View (AoV) is represented in terms of the horizontal AoV, α_h, and the vertical AoV, α_v, as shown in the Figure 1. The *depth-of-field* (DoF) represents the area of the visual scene that is acceptably sharp, bounded by R_{min} and R_{max}.

For PTZ cameras, *panning* and *tilting* allow rotating the camera around its axis to adjust FoV horizontally (pan) and vertically (tilt). Camera zoom level can be controlled to change the camera DoF: zooming-in results in an increase in R_{min} and R_{max}, at a cost of a decrease in α, and vice versa. This change in FoV shape with respect to zoom complicates the problem of determining all maximal FoVs.

B. Motivation

Recall that the DISCRETE algorithm is typically used to enumerate the set of FoVs a camera can use [7], [8], [10], [16], [21]. In this section, we highlight limitations of using DISCRETE for PTZ cameras. Although current commercial cameras enable finely controllable movements in PTZ dimensions, DISCRETE allows only a finite set of FoVs per camera (e.g. 8 pans separated by 45° [2], [16]). To show the limitations of DISCRETE we conduct an experiment with 10 cameras and 100 targets placed uniformly on a $100 \times 100 \times 100m^3$ terrain. Each camera first generates all discrete candidate FoVs, which are passed to a coverage maximization algorithm [2], [16] to generate camera-FoV assignments that cover the maximum number of unique targets. *Percent Coverage* is used to evaluate coverage; it measures the percentage of targets covered out of all the targets that could be covered with some PTZ setting. Percent coverage is used simply to ignore non-coverable targets.

Figure 2(a) shows the impact of using various DISCRETE step-sizes on coverage. For example, step-size of 1 means the pan and the tilt angles are considered at intervals of 1° and the focal-length (zoom) at intervals of 1 mm. To limit the run-time, all cameras are constrained with 90° movement in pan, 40° movement in tilt, and $3\times$ zoom. Even with such constrained movement, DISCRETE with step-size 1 generates over 2.5 million FoVs for 10 cameras. Yet DISCRETE misses some optimal FoVs that would have maximized the coverage, as indicated by the gap between DISCRETE step-1 and Optimal. As the step-size is increased, more intermediate FoVs that cover the most number of targets are missed, resulting in reduced coverage. Figure 2(b) shows that the time to find optimal camera-FoV assignments is significantly high for smaller step-sizes. Such delay is especially harmful in dynamic settings,

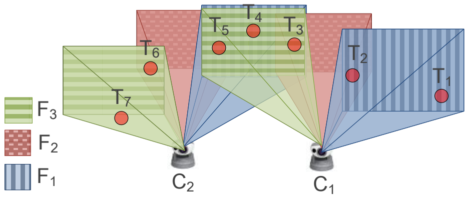


Fig. 3. Illustration of MaxFoVs.

where the problem has to be solved repeatedly as the targets or cameras move [17].

C. MaxFoV Problem Statement

Motivated by the goal of minimizing the number of FoVs per camera, we first define the *Maximal FoV Identification* (MaxFoV) problem and propose a solution to solve it efficiently. MaxFoV is a per-camera problem of finding a minimum set of non-redundant maximal FoVs. We first develop the following definitions.

Definition 1: A *Coverable Set* (Cover_C) is a set of targets that can be covered by one or more feasible FoVs of a camera C; a *Coverage Set* (CS_F) is a set of targets that are covered by a particular FoV, F, of a camera.

Definition 2: A *Maximal Coverage Set* (MCS) is a coverage set that is not a proper subset of another coverage set.

The MaxFoV problem aims to obtain all MCS's for a camera. The reasoning is that any coverable set of targets is coverable by one or more MCS. Thus, retaining only MCS' in the candidate FoV set for each camera is guaranteed not to restrict the coverage that can be obtained by the multi-camera coverage algorithm.

Figure 3 shows a scenario with two cameras and 7 targets. Each camera can cover one or more targets in its FoV. For C_1, the coverable set Cover_C is $\{T_1, T_2, T_3, T_4, T_5\}$, and the possible coverage sets are: $\{T_1\}$, $\{T_2\}$, $\{T_1, T_2\}$, $\{T_2, T_3\}$, $\{T_3, T_4\}$, $\{T_4, T_5\}$, $\{T_3, T_4, T_5\}$. Out of these coverage sets, only three are maximal coverage sets, $\{T_1, T_2\}$, $\{T_2, T_3\}$, and $\{T_3, T_4, T_5\}$, formed by maximal FoVs F_1, F_2, F_3 of C_1, respectively. Note that the FoV F_1 of C_1 uses a lower zoom to form a wider FoV that can jointly cover $\{T_1, T_2\}$. Similarly, the maximal coverage sets for C_2, $\{T_3, T_4, T_5\}$, $\{T_5, T_6\}$, $\{T_6, T_7\}$, are formed by maximal FoVs F_1, F_2, F_3 of C_2, respectively. We have omitted R_{min} in the figure for clarity.

The maximal FoVs serve as an input to all target-oriented coverage algorithms for PTZ cameras. For instance, the optimal camera-FoV assignment to enable covering the maximum number of targets is $C_1 - F_1, C_2 - F_1$ or $C_1 - F_3, C_2 - F_3$, which results in covering 5 targets. On the other hand, the greedy assignment, where each camera locally selects an FoV that covers the maximum targets, would be $C_1 - F_3, C_2 - F_1$, resulting in covering only 3 targets.

Assumptions: We assume that the locations of targets are known to the cameras. This assumption is consistent with the prior works, where targets' locations are determined using a complementary mechanism such as using a supporting network of low-resolution wide-angle cameras [12], [15], or camera calibration [5]. To illustrate the feasibility of this assumption, in our testbed experiments, we implement calibration-based localization and evaluate its accuracy. We also assume that the targets are stationary and a target is covered when some point, say its centroid, is visible in the FoV. Target mobility can be supported by re-invoking the coverage algorithm [17].

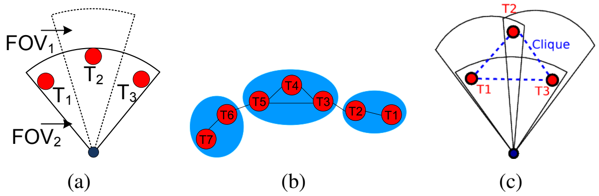


Fig. 4. Fig. (a) $FOV2$ is set at Min-Zoom to cover target T_2. $FOV2$ covers T_1 and T_3, which would be missed by $FOV1$ with its longer zoom; Fig. (b) shows coverage-graph for scenario in Fig. 3; In Fig. (c) Clique (T_1, T_2, T_3) does not belong to the coverage set $\{T_1, T_2, T_3\}$ when zoom is considered.

III. MaxFoV: Complexity Analysis

In this section, we show that solving MaxFoV is at least as hard as solving the problem of generating all maximal cliques.

Considering first the FoV shape to be fixed, a group of targets is coverable in an FoV if the maximum horizontal and vertical angles made by the targets with the camera are less than the FoV angles (α_h and α_v, respectively (please see Figure 1), and the distance of each target to the camera is between R_{min} and R_{max}. We call this test the *Fixed-FoV Coverage Test*.

However, since the FoV shape changes with zoom, it is difficult to determine joint-coverability of a group of targets. To address this difficulty, we first introduce the *Minimum Zoom Level (MZL)* property, which defines an order for considering targets for establishing joint target coverability.

A. Minimum Zoom Level (MZL) Property

The Minimum Zoom Level (MZL) property is based on the observation that the width of the FoV monotonically decreases as the zoom level increases. Thus, to maximize coverage, while covering a given farthest target, MZL implies that we should consider FoVs with the *minimum zoom* that covers that farthest target, as shown in Figure 4(a). Minimum Zoom can be formally defined as:

Definition 3: **Minimum Zoom (Min-Zoom):** Let a target T be in the coverable range from camera C. Then T is said to bound an FoV with Min-Zoom, if the the zoom can not be reduced further without losing T.

MZL allows us to use *one* zoom value that maximizes the chance to cover the group of targets together. Once the zoom value is fixed to the minimum zoom that will cover the farthest target, the FoV shape is fixed allowing direct application of the *Fixed-FoV Coverage Test* to check if the group of targets is jointly coverable.

B. MaxFoV and Generating All Maximal Cliques

We now show that solving MaxFoV for PTZ cameras is at least as hard as solving the problem of Generating All Maximal Cliques (GAMC), which is known to be NP-hard [9], [13] for general graphs. We show this by showing that solving MaxFoV for Pan-Tilt (PT) cameras, with fixed zoom = 1×, requires solving GAMC.

A *coverage graph* $G = (V, E)$, can be constructed using the targets coverable by a camera C, Cover_C. In G, each target is a vertex, and there exists an edge between two vertices if those targets are jointly coverable by the camera. For instance, Figure 4(b) shows a coverage graph generated for the scenario in Figure 3. Shaded regions show maximal-cliques, which correspond to MCS' or maximal-fovs. Now we show that finding all coverage sets for a camera requires finding all cliques in G.

First, we show that each coverage set (CS_F) can be represented by a clique, and vice versa. A *clique* is a subset of vertices of a graph where each vertex is connected to each other vertex. A *maximal clique* is one that can not be extended by including one more vertex.

Theorem 1: For Pan-Tilt camera, a clique, $\mathbb{C}$, is present in G iff the union of targets $\in \mathbb{C}$ is a coverage set.

Proof: First, we show that if there is a coverage set, there must be a clique in the graph, G. A coverage set is a set where there exists an FoV that covers all the targets jointly. Thus, each pair of nodes in the coverage set are pairwise coverable by the same FoV, and the coverage set forms a clique in the graph G.

Next, we prove that the opposite is true for PT cameras: that if there exists a clique, $\mathbb{C}$, in G, then all the targets $\in \mathbb{C}$ are jointly coverable, forming a coverage set. We will prove this statement by induction.

Base case: consider a clique of one target. This clique is also a cover set since the target is coverable by the camera.

Inductive step: we show that if we have a clique of n vertices in the graph that forms a coverage set of targets $T_1 \dots T_n$, say CS_n, then a new clique of size $n+1$ is formed by adding a new target T_{n+1} to the original clique if and only if T_{n+1} is jointly coverable with CS_n. If the inductive step is true, then all cliques are also coverage sets.

We prove this step by showing that T_{n+1} is jointly coverable in the pan dimension first, and then the tilt dimension. Without loss of generality, consider that T_1 is the leftmost and T_n is the rightmost target in the CS_n. If T_{n+1} is to the left of T_1 then it is jointly coverable with all the nodes in the pan dimension since it is pairwise coverable with the rightmost node T_n. The same reasoning applies if T_{n+1} is to the right of T_n. If T_{n+1} lies between T_1 and T_n, then it is easy to see that it is jointly coverable in the pan dimension (e.g., by the same setting of the FoV covering CS_n). After fixing the pan to cover all $T_1 \dots T_{n+1}$ targets, the same reasoning can be applied for the tilt dimension to show joint coverability in both dimensions. Thus, the inductive step is true. ■

Thus, all maximal coverage sets (MCS's) of Cover_C correspond to all maximal cliques in G. That is, solving MaxFoV for PT cameras requires solving GAMC on the coverage graph.

MaxFoV for PTZ cameras: The proof above relies on a fixed FoV shape (which is true when the zoom is fixed) for the inductive step to be true. Since the shape of the FoV changes with the zoom value, not every clique in the graph is guaranteed to be a jointly coverable set. We show this by example in Figure 4(c) where three targets are each pairwise coverable with each of the other two, but are not jointly coverable since no single FoV exists that covers all three. We call such cliques non-coverable cliques.

In order to solve MaxFoV for PTZ cameras, one can solve GMAC to identify all maximal cliques. If all targets in a maximal clique are jointly coverable, they represent an MCS. If they are not jointly coverable, then sub-cliques of smaller cardinality have to be generated from them and checked for coverability.

Solving MaxFoV for PTZ cameras is at least as hard as solving MaxFoV for PT cameras. In other words, if there existed a solution with lesser complexity for PTZ camera, it must be applicable for the special case of PT cameras (PT cameras with zoom = 1×).

The problem of GAMC for general graphs is NP-hard [9], [13]. However, Tsukiyama et al [20] showed that polynomial time solutions are possible when the number of maximal cliques in the graph is polynomial. As a result, the problem has polynomial time complexity for special graphs such as chordal graphs and planar graphs. In the next section, we show that the number of maximal cliques (and maximal coverage sets) in the coverage graph is in fact polynomial due to the geometric constraints of the FoV shape.

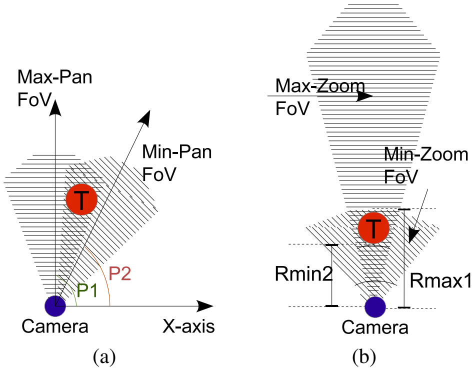
IV. EFoVs: Polynomial Time MCS Generation

In this section, we define the notion of Extreme-FoVs (EFoVs) and use it to show that the number of maximal coverage sets (and also the number of maximal cliques in the coverage graph) is polynomial. In the case of PT cameras, this result establishes that a polynomial time algorithm for generating all maximal cliques exists for coverage graphs since the problem is an instance of GMAC [20]. In the next section, we develop algorithms to optimally solve general MaxFoV in polynomial time; i.e., the problem has polynomial complexity even for PTZ cameras.

EFoVs are the FoVs that have at least one target at the extreme edges of each of the pan, tilt, and zoom dimensions of the FoV. The targets on the edges of an FoV are called bounding targets. Informally, the key property of EFoVs is that they enable covering the *maximum* number of other targets in addition to the bounding targets. We first formally define EFoVs, and then prove that every MCS must be covered by at least one EFoV. Finally, we show that with respect to the number of coverable targets, only a polynomial number of EFoVs exists.

Definition 4: **Extreme FoV (EFoV)** Let F be an FoV of a camera C. F is an Extreme FoV if it is bounded by a target in each of the Pan, Tilt, and Zoom dimensions. An FoV, F, is said to be bounded by a target T_B in a dimension (pan, tilt or zoom) if a new FoV, F', obtained by moving F by an arbitrarily small distance in the bounding dimension away from the target (i.e., towards the opposite boundary in the same dimension) results in an FoV that does not cover the target.

Each of the n targets may serve as the bound for an EFoV in each of the Pan, Tilt, and Zoom dimensions; note that one target can also be a bound in multiple dimensions



(a)

(b)

Fig. 5. Fig. (a) P1 > P2. Thus, FoV creating angle P1 with X-axis is Max-Pan FoV w.r.t. target T. Both Min-Pan and Max-Pan FoVs are EFoVs in pan, bounded by T. Fig. (b) shows Min-Zoom and Max-Zoom FoVs to cover T.

simultaneously. An FoV has two bounding edges in each pan, tilt, and zoom dimensions. Depending on which edge is bounded by a target, an EFoV can be classified as Min/Max-Pan/Tilt/Zoom EFoV. For instance, when a target bounds an EFoV in Min-Zoom, which was defined Section III-A, it is called as a *Min-Zoom EFoV*. Definitions for *Max-Zoom*, *Min-Pan*, *Max-Pan*, *Min-Tilt*, and *Max-Tilt* can be constructed in a similar way.

For example, Figure 5(a) shows how an FoV can created to bound target T with two edges of pan, creating two EFoVs. Depending on the angle they create with the X-axis, we classify them as either Min-Pan or Max-Pan FoVs. Min-Tilt and Max-Tilt FoVs can also be shown in similar way. Figure 5(b) shows two FoVs bounding T with minimum and maximum zoom values, creating Min-Zoom and Max-Zoom FoVs. Essentially, Max-Zoom FoV bounds the FoV with R_{min2} and Min-Zoom FoV bounds the FoV with R_{max1}

Note that cameras may not be able to bound a target in the pan, tilt or zoom dimensions due to the camera's mechanical limits. In such cases, the bound is obtained as the nearest legal camera setting.

Theorem 2: Each MCS is covered by at least one EFoV.

Proof: By definition, every MCS S can be covered by at least one FoV F. To prove the theorem, we must show that S can also be covered by an EFoV. We start from the covering FoV F and show that there also exists an EFoV EF that covers S by construction.

We start by reducing the zoom of F to the point where it may no longer be decreased without losing a target, or the minimum zoom limit is reached, whichever comes first. Reducing the zoom expands the pan and tilt angles, and therefore does not remove coverage of any existing targets. If zoom reduction was stopped to avoid a target loss, the target will be at the edge of the FoV in the zoom direction; such a target is a bounding target, defining the Min-Zoom bound of an EFoV.

Bounding targets (or limits on Pan and/or Tilt movements) in the remaining two dimensions can also be obtained in a similar fashion. For example panning the FoV to the left (or right) until it is bounded by a target, will not lose coverage of any targets. It will also not remove the zoom bound since the two directions are orthogonal. Similarly, tilting the FoV up (or down) until it is bounded by a target, will not affect the pan or zoom bound, resulting in an FoV that is bounded in all three axes – this is an EFoV that covers S. With this procedure, we show that an EFoV exists that covers each maximal subset of coverable targets. In the special case that F cannot be moved in one or more dimensions without losing coverage of a target, then it is already bounded in those dimensions and need only be moved in the unbounded dimensions, if any, to construct an EFoV. ■

Polynomial Number of EFoVs: Thus far, we have shown that every MCS can be covered by an EFoV. Next we show that there is only a polynomial number of EFoVs, and consequently, a polynomial number of MCS'.

Each EFoV is bound by 3 (or less if one target is the bound in more than one direction) targets in the three dimensions. If there are n coverable targets, a maximum of n^3 combinations of three targets can be generated. Given a group of three targets, they may each bound the EFoV in any of the three dimensions. Moreover, in each dimension, the bound can be in two directions (for example, Min-Pan or Max-Pan). Thus, a group of 3 targets, can provide the bounds for up to 8 different EFoVs. Thus, the total number of EFoVs can be no more than $8n^3$: it is a polynomial function of the number of targets.

V. MaxFoV Algorithms

We now present two polynomial time algorithms for solving MaxFoV. The first algorithm (*Basic Extreme FoV Algorithm (B-EFA)*, generates all EFoVs and filters out infeasible or redundant ones. The second algorithm, *Geometric Extreme FoV Algorithm (G-EFA)*, exploits the geometric structure of the problem to generate only feasible EFoVs and to find maximal EFoVs. Finally, we present an FoV centering mechanism to provide tolerance against target localization errors, since the targets on the edges of EFoVs are most vulnerable to localization errors.

A. Basic Extreme FoV Algorithm (B-EFA)

B-EFA simply generates all possible EFoVs. by considering *all* possible combinations of three targets to bound an FoV in all three dimensions. Thus, B-EFA generates $8n^3$ EFoVs.

Filtering non-Maximal FoVs: Not all groups of three targets are jointly coverable. For instance, two targets may be in the front, while the third target may be behind the camera. We remove such infeasible EFoVs by discarding any FoV that exceeds the camera's FoV parameters.

Furthermore, some EFoVs may be redundant (e.g. a group of three targets can be bounded by both Min-Pan and Max-Pan), and non-maximal (e.g. CS $\{T_4, T_5\}$ in Figure 3, which is a subset of MCS $\{T_3, T_4, T_5\}$). To remove such EFoVs, B-EFA performs pairwise matching of each of their CS's with the other CS's, resulting in an $O(m^2)$ comparisons, where m is the input number of EFoVs. We implement more efficient filtering that exploits the geometry of the FoVs in G-EFA.

B. Geometric Extreme FoV Algorithm (G-EFA)

G-EFA avoids creating unnecessary EFoVs by using the following geometric observations: (1) EFoVs identified by B-EFA can be infeasible, since many triplets of targets are not jointly coverable; and (2) For each triplet of targets, it is sufficient to consider only one EFoV, instead of 8 possible EFoVs. We now illustrate how G-EFA exploits the two observations (Algorithm 1 shows the full implementation).

Optimization 1 - Finding only feasible target triplets: Since an FoV has a limited angle-of-view or AoV (45° for Axis 213 PTZ camera at minimum zoom), all the pairs of targets whose angle with the camera is larger than the camera's maximum AoV are not jointly coverable. Thus, to obtain only jointly coverable triplets of targets, G-EFA considers targets $\in Cover_C$ ordered in zoom, pan, and tilt dimensions (line 1). For the first target in decreasing zoom-order (i.e. the farthest target from camera), $T_{R_{max}}$, G-EFA sets the Min-Zoom, e_{zoom} to cover it (line 2,3). Next it refines the coverable targets set, say $Cover_{e_{zoom}}$, to consider the only targets that are jointly coverable with $T_{R_{max}}$ (line 4).

G-EFA repeats this process for pan and tilt bounding targets (lines 5-7): From $Cover_{e_{zoom}}$, it selects a target in pan order, $T_{e_{pan}}$, for which, e_{pan} is the Min-Pan. G-EFA refines the coverable targets set, $Cover_{e_{zoom},e_{pan}}$, to consider the only targets that are jointly coverable with $T_{R_{max}}$ and $T_{e_{pan}}$. Now, each target in $Cover_{e_{zoom},e_{pan}}$ acts as a bounding target in tilt, and thus forms a triplet of jointly coverable targets with $T_{R_{max}}$ and $T_{e_{pan}}$. For each such triplet, its corresponding EFoV and coverage set are constructed (lines 9-11).

Once all triplets of jointly coverable targets with $T_{R_{max}}$ and $T_{e_{pan}}$ are formed, the next target in pan order is selected. At this point, $T_{e_{pan}}$ can be safely ignored from $Cover_{e_{zoom}}$ (not from $Cover_C$ though), since all triplets with it are already formed (line 12). We call this process *progressive elimination*, which is also applied for each tilt, and zoom bounding targets.

Correctness: To generate feasible target triplets, G-EFA differs from B-EFA in its use of a progressive elimination technique. Thus, we show that progressive elimination does not result in missing any feasible target triplets. Consider progressive elimination for zoom, where targets are considered in the *decreasing* zoom order, and are bounded by Min-Zoom. In such cases, after generating all EFoVs specific to the current zoom-bounding target, that target can never be a part of EFoVs generated using the remaining zoom-bounding targets (the ones that require lower Min-Zoom). The correctness of it can be trivially verified based on the definition of Min-Zoom. Similarly, the correctness of progressive elimination for pan and tilt dimensions can be shown.

Optimization 2 - Selecting only one out of 8 EFoVs: Note that G-EFA constructs only one EFoV out of the 8 possible EFoVs for each triplet of jointly coverable targets (line 9). G-EFA considers the EFoV bounded by ⟨ Min-Zoom,Min-Pan,Min-Tilt ⟩, but we can choose to use either Min-Pan or Max-Pan, and either Min-Tilt or Max-Tilt. However, for zoom, only Min-Zoom can be used.

Correctness: The min-bound or max-bound in the pan and tilt dimensions can be alternatively used because when the zoom is fixed, the FoV size remains constant. Thus, the FoV can be moved in pan and tilt dimensions as long as the coverage set remains the same; thus, we can move the FoV so that it is bounded in our choice of min-bound or max-bound and still cover the MCS. However, Max-Zoom bound can not be used, because as the zoom is increased, the FoV narrows and some targets may become uncovered.

Algorithm 1: Geometric Extreme-FoV Algorithm

```
input  : $Cover_C$ = Set of coverable targets of $C$
output: $\mathbb{E}$ = Set of all EFoVs for $C$
1  Sort $Cover_C$ three times once for each dimension, pan, tilt and zoom
2  for $T_{r_{max}} \in Cover_C$, the farthest remaining target do
3      $e_{zoom}$ = the minimum zoom necessary to cover $T_{r_{max}}$.
4      $Cover_{e_{zoom}}$ = the set of targets pairwise coverable with $T_{r_{max}}$ at zoom $e_{zoom}$; the targets are sorted in pan-dimension.
5      for $T_{e_{pan}} \in Cover_{e_{zoom}}$, the bounding target in the pan dimension do
6          $Cover_{e_{zoom},e_{pan}}$ = the set of targets jointly coverable with both $T_{r_{max}}$ and $T_{e_{pan}}$; the targets are sorted in tilt-dimension.
7          for $T_{e_{tilt}} \in Cover_{e_{zoom},e_{pan}}$, the bounding target in tilt dimension do
8              $Cover_{e_{zoom},e_{pan},e_{tilt}}$ = Coverable targets in addition to $T_{rmax}$, $T_{epan}$, and $T_{etilt}$.
9              $EFOV = \langle e_{zoom}, e_{pan}, e_{tilt} \rangle$.
10             $\mathbb{E} = \mathbb{E} \cup \langle EFOV, Cover_{e_{zoom},e_{pan},e_{tilt}} \rangle$
11             $Cover_{e_{zoom},e_{pan}} = Cover_{e_{zoom},e_{pan}} \setminus T_{e_{tilt}}$
12         $Cover_{e_{zoom}} = Cover_{e_{zoom}} \setminus T_{e_{pan}}$
13     $Cover_{e_{zoom}} = Cover_{e_{zoom}} \setminus T_{r_{max}}$
```

Complexity: While G-EFA has a worst case run-time of $O(n^3)$, geometric bounds help reduce the sizes of $Cover_{e_{zoom}}$ and $Cover_{e_{zoom},e_{pan}}$ significantly. Note that the targets can be pre-sorted in pan, tilt, and zoom dimensions. In addition, coverable set construction (line 8) can be performed in $O(1)$, since the targets are already sorted in the tilt-dimension.

Obtaining Maximal FoVs: G-EFA applies geometric optimizations to avoid performing m^2 pair-wise matching of CS's to remove non-maximal and redundant CS's (recall that m is the number of generated FoVs):

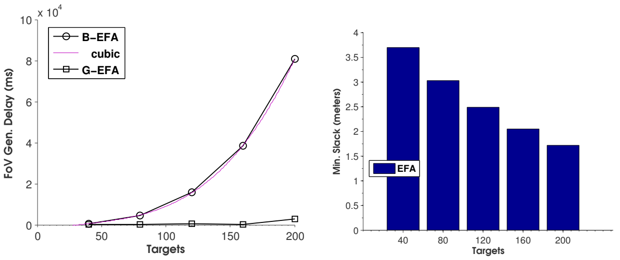
Optimization 1: If targets $T_i \in CS_F$ and $T_j \in CS_{F'}$ are not jointly coverable, then their coverage sets, CS_F and $CS_{F'}$, will never have one of them as a proper subset of the other.

Optimization 2: Once a set is found to be redundant, or once it is compared against all other subsets, it is removed from further matching consideration. Thus, to facilitate quick detection and elimination of redundant subsets, we compare the largest cardinality subsets first, which increases the likelihood of finding and eliminating redundant subsets quickly.

While complexity of finding maximal FoVs is $O(m^2)$, geometric optimizations help ignore matching many coverage sets, resulting in a significantly lower runtime for most target distributions.

C. FoV Refinement using Centering

MaxFoV uses EFoVs that are bounded by targets; this leaves little tolerance for localization errors for the targets on



(a)

(b)

Fig. 6. (a): FoV generation delay (milliseconds); (b): tolerance of localization error, represented as a slack (in meters.)

Targets	G-EFA	Discr_1	Discr_2	Discr_3	Discr_4	Discr_5
40	0.151	47.360	3.432	0.967	0.402	0.294
80	0.424	68.838	5.553	1.961	0.754	0.517
120	1.314	88.097	8.383	2.879	1.361	0.808

TABLE I. TIME (IN SECONDS) TO COMPUTE OPTIMAL CAMERA-FOV ASSIGNMENTS.

the FoV edges. To increase the tolerance, we apply centering on the resulting EFoVs. The goal of FoV-centering is to bring the group of covered targets at the center of the FoV. This is achieved by adjusting the FoV to maximize the distance (or slack) of bounding targets from the edges of the FoV.

Specifically, FoV centering is a two step process. First, we compute a geometric center, G_c, of the FoV coverage set (covered targets), and adjust pan and tilt to bring G_c at the center of the FoV. Next, we increase zoom in small steps, which reduces the slack in pan and tilt dimensions, until the slacks in all pan, tilt, and zoom dimensions are equal. FoV centering is applied after the execution of G-EFA.

VI. PERFORMANCE EVALUATION

We evaluate the MaxFoV algorithms using simulation and a small testbed. The camera testbed consists of Axis 213 cameras [4] integrated with the OpenCV library which provides implementations of common vision algorithms [1]. The testbed size and location limits us to experiments with limited numbers of targets. Thus, to study more general scenarios, we developed 3D camera coverage models within the QualNet simulator. We carefully modeled the camera parameters in terms of its FoV and PTZ ranges to match the Axis 213 cameras used in the testbed. Finally, we present a use case, in which we integrate G-EFA with coverage maximization algorithms [2], [16] to study the impact of FoV selection on multi-camera coverage.

A. Simulation: Target Scale and Distribution

In simulation, We study the impact of different scales and distributions of targets on the number of FoVs generated, the FoV generation time, and the tolerance for localization errors.

1) MCS Generation: For a single camera, we evaluate B-EFA and G-EFA to study the EFoV generation overhead and localization error tolerance for varying number of targets. The camera and targets are deployed on a $100 \times 100 \times 100$ m^3 terrain. The PTZ ranges used for the camera are similar to that of the Axis-213 camera: Pan: $+180°$ to $-180°$, Tilt: $+90°$ to $-90°$, and $26\times$ zoom (i.e. focal-length range of 3.5 mm to 91 mm).

Uniform Distribution: For the first set of experiments, the targets and cameras are placed using a uniform distribution in the simulation area. Results for each setting are obtained by averaging 20 different runs. Figure 6(a) plots the FoV generation delay for B-EFA and G-EFA. As expected, the delay for B-EFA is cubic in nature, as shown in the graph. However, the FoV generation time for G-EFA is reduced by an average of 96% due to the geometric optimizations reducing the number of generated FoVs.

Next, to obtain the localization error tolerance, we measure *minimum slack* observed after centering MaxFoVs. The minimum slack is essentially the minimum distance between the covered targets and the FoV edges. As shown in the Figure 6(b), minimum slack decreases with the increasing target density, since more and more targets can fit within an FoV. However, even for 200 targets, the minimum slack is about 1.75 m. This shows that G-EFA with FoV-centering tolerates localization errors even in case of high target density.

2) G-EFA Vs. DISCRETE: In this experiment, we use G-EFA and DISCRETE as FoV generation approaches for each camera in a multi-camera setting. The goal of this experiment is to study the difference in coverage benefit and overhead using DISCRETE with varying step-size and G-EFA.

For the experiment discussed in Section II-B, G-EFA not only provides the optimal solution, but also solves the multi-camera problem in 0.83 seconds. G-EFA generates only 544 FoVs. For varying number of targets, Table I compares the delay to generate optimal camera-FoV assignments, when G-EFA and DISCRETE are used. As the target density increases, G-EFA takes relatively more time as more EFoVs must be considered. Nonetheless, the delay required by G-EFA is drastically less than that of DISCRETE with fine step-sizes, while maintaining optimality.

Figure 7 and 8 represent the percentage of targets covered and the total number of FoVs generated by all cameras, respectively. For this experiment, we consider uniform camera and target deployment in a 3D space of $100 \times 100 \times 100 m^3$. For the discrete case, x in *Discrete_x* represents the step size. The results show that irrespective of the number of targets, G-EFA provides optimal coverage with significantly less overhead (number of FoVs).

B. Testbed Validation

In this section, we first discuss the camera calibration and target localization methods used in our testbed experiment, followed by the evaluation of G-EFA in realistic settings.

1) Calibration and Target-localization: For the miniaturized testbed, we perform calibration and target localization to automatically obtain world coordinates of the targets. The coordinates of cameras are assumed to be known. The cameras first map their PTZ parameters to the world coordinates; this process is called calibration. We use *landmark-based calibration* using *anchor targets* whose world coordinates are known in advance. The cameras view the anchors in various FoVs by slightly altering their PTZ parameters, allowing them to localize using known approximation techniques. The cameras then detect the targets (circles), and localize them by transforming the local PTZ parameters to the world coordinates.

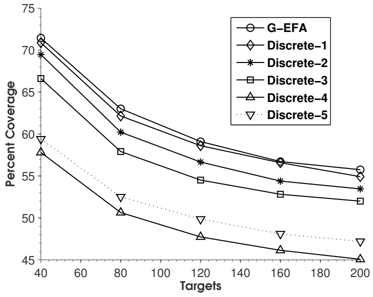


Fig. 7. G-EFA vs. DISCRETE (in Simulation): Coverage.

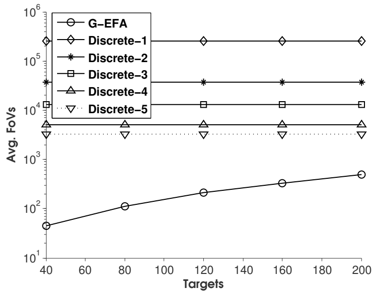


Fig. 8. G-EFA vs. DISCRETE (in Simulation): Number of FoVs.

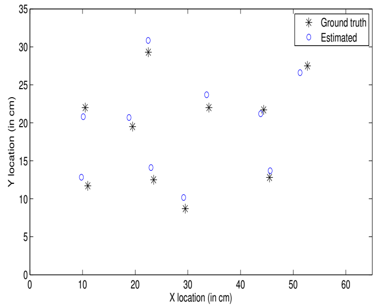


Fig. 9. Calibration and Localization Accuracy

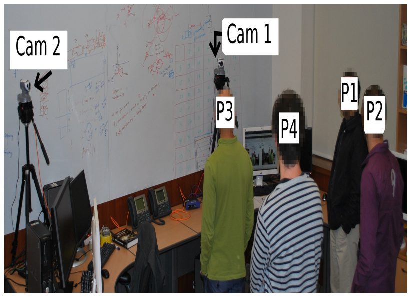


Fig. 10. People monitoring scenario

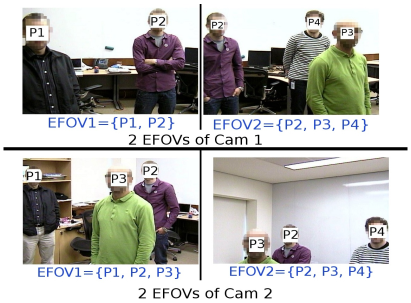


Fig. 11. Maximal EFoVs

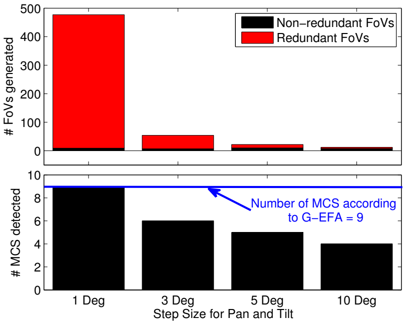


Fig. 12. G-EFA vs. DISCRETE (in testbed): Number of FoVs and MCS's.

We observed an average accuracy error in localization is around 1.3 cm, which translates to around 2% error in the testbed area. Figure 9 shows localization errors in a random scenario with 10 targets. More advanced location estimation techniques, such as SLAM and SURF [6], can also be used.

2) MCS Generation: We now demonstrate the applicability of MCS generation process of G-EFA in realistic settings. Our experiments are based on a people monitoring application. We begin with a small scale experiment, where two cameras are trying to monitor four persons standing in a room. For our subsequent experiments, we abstract a person's face into a circle (or dot) primarily for three reasons: (1) to work with medium-scale scenarios of targets; (2) to maintain privacy of people; and (3) to miniaturize the testbed space to efficiently perform the testbed specific operations.

Camera Testbed: Figure 10 shows the topology of two cameras (Cam1 and Cam2) monitoring four persons (P1 to P4). Coordinates of the center of a person's face is considered as a target's coordinate. EFoVs cover the whole face by adjusting the e_{pan} and e_{tilt} angles by the angle constituting half side of a face. Each camera locally executes G-EFA and generates MCS's (EFoVs) as shown in the figure.

Recall that G-EFA only needs to consider either Min-Pan and Min-Tilt, or Max-Pan and Max-Tilt out of the all possible EFoVs that it can generate by keeping bounding targets on the edges. We show both the types of EFoVs–all EFoVs except $EFoV2$ for $Cam2$ are Min-Pan and Min-Tilt based. $EFoV2$ for $Cam2$ has its Max-Pan bounded by $P4$ and Max-Tilt bounded by $P3$. In contrast, a Min-Pan and Min-Tilt based version of $EFoV2$ for $Cam2$ would have bounded $P3$ in Pan and $P4$ in Tilt dimensions.

3) G-EFA Vs. DISCRETE: We compare G-EFA and DISCRETE in the miniature testbed by placing 20 targets randomly in an area of $50 \times 16\ cm^2$. In this area, the camera has an overall pan and tilt range of around $48°$ and $10°$, respectively. G-EFA provides 9 MCS for this scenario, all of which were detected in the testbed. In the DISCRETE experiment, we vary the step size of pan and tilt by $1°$, $3°$, $5°$ and $10°$. Figure 12 shows the number of FoVs generated, and the number of MCS detected under each step size. At smaller step-sizes, the number of FoVs generated is large. However, most of the FoVs generated are redundant; redundant FoVs cover a sub-set of the targets that are already covered in a non-redundant FoV. For example, at a step-size of $1°$, there are 477 FoVs that are generated, out of which only 9 FoVs are non-redundant. All the targets tracked in other 468 are already tracked in at least one of the 9 non-redundant FoVs.

The number of MCS detected also varies depending on the step-size. Smaller step size aids in detecting all the MCS, which are calculated by the G-EFA algorithm. Larger step-size misses a majority of the MCS. For example, with a step-size of $10°$, only 4 out of 9 MCS are detected. Thus, DISCRETE algorithm either results in large number of redundant FoVs or detects only a small fraction of available MCS; both hinder the coverage algorithm to track maximum number of targets.

C. Use-case: Target Coverage Maximization

We now present a use-case for a multi-camera coverage maximization approach. Our approach is two-fold: (1) G-EFA is applied to generate MCS's for each camera; (2) Using the MCS's for all cameras as an input, existing target coverage maximization approaches [2], [16] are run in testbed and in

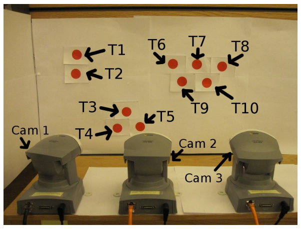


Fig. 13. Clustered Placement of Targets.

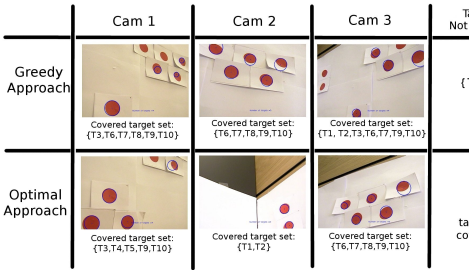

Cam 1
Cam 2
Cam 3
Targets Not Covered
Greedy Approach
Covered target set: {T3,T6,T7,T8,T9,T10}
Covered target set: {T6,T7,T8,T9,T10}
Covered target set: {T1, T2,T3,T6,T7,T9,T10}
{ T4, T5 }
Optimal Approach
Covered target set: {T3,T4,T5,T9,T10}
Covered target set: {T1,T2}
Covered target set: {T6,T7,T8,T9,T10}
All targets covered

Fig. 14. G-EFA Use-case: Optimal vs Greedy

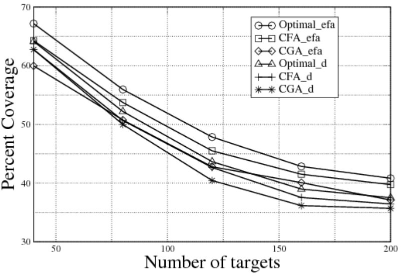


Fig. 15. Simulation: coverage maximization

simulation to generate best camera-FoV assignments to cover *maximum unique targets*. We modify the existing approaches, developed for 2D coverage using Pan-only cameras, to incorporate the 3D coverage model for PTZ cameras.

In testbed, we evaluate two multi-camera algorithms, *greedy* and the centralized *optimal*. In the greedy approach, each camera locally configures the EFoV that maximizes the number of targets covered by that camera. In the centralized optimal approach, EFoVs of all cameras are provided to a central solver that generates optimal camera-FoV assignments to cover maximum unique targets [16]. To compare the two strategies, we deploy cameras and targets as shown in Figure 13. Figure 14 shows that greedy coverage fails to track two targets whereas the optimal coverage tracks all the targets.

In simulation, we study the existing centralized and distributed heuristics: Centralized Greedy Approach (CGA), Centralized Force-based Approach (CFA), and Distributed Greedy Approach (DGA) [2], [16]. We do not describe these approaches in the interest of space. We deploy 10 cameras with varying target density uniformly on a $100 \times 100 \times 100m^3$ terrain. The algorithms use G-EFA (denoted as *_efa) and Discrete_1 (denoted as *_d) FoV generation approaches. In Figure 15, it is interesting to note that not only the Optimal_efa, but also the centralized heuristic, CFA_efa, provides better coverage than the Optimal_d approach.

VII. Concluding Remarks

We considered the coverage problem for PTZ smart camera networks. The degrees of freedom in the selection of FoVs for PTZ cameras make it prohibitively expensive to consider all possible FoVs. We formulated a MaxFoV problem whose goal is to identify FoVs representing the maximal subsets of targets. MaxFoV is a new problem fundamental to target coverage in sensor networks with PTZ cameras. We showed that MaxFoV is at least as difficult as generating all maximal cliques in a graph, which, for general graphs, is NP-hard. For some special graphs, number of maximal cliques is known to be polynomial. The geometric nature of PTZ coverage also makes MaxFoV polynomial, which we demonstrated by constructing an efficient algorithm to optimally solve the problem with a worst case complexity of $O(n^3)$. We showed the correctness of the algorithm and studied its performance and run-time both in a testbed and in a simulation environment. The proposed algorithm substantially outperforms the existing schemes by providing only the essential FoVs that are required for optimal coverage, thus significantly reducing the time required to solve coverage problem in multi-camera networks.

References

[1] Opencv (open source computer vision). (http://opencv.willowgarage.com/wiki/).

[2] J. Ai and A. Abouzeid. Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization*, 2006.

[3] I. Akyildiz, T. Melodia, and K. Chowdhury. A survey on wireless multimedia sensor networks. *Computer Networks*, 2007.

[4] Axis 213 ptz network camera. Available at http://www.axis.com/products/cam_213/.

[5] A. Barton-Sweeney, D. Lymberopoulos, and A. Sawides. Sensor localization and camera calibration in distributed camera sensor networks. In *Proc. of IEEE BROADNETS*, 2006.

[6] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *Computer Vision , ECCV*, LNCS. Springer Berlin / Heidelberg, 2006.

[7] Y. Cai, W. Lou, M. Li, and X. Li. Target-Oriented Scheduling in Directional Sensor Networks. *IEEE Infocom*, 2007.

[8] G. Fusco and H. Gupta. Selection and orientation of directional sensors for coverage maximization. In *Proc. of IEEE SECON*, 2009.

[9] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. 1979.

[10] M. Johnson and A. Bar-Noy. Pan and scan: Configuring cameras for coverage. In *Proc. of IEEE INFOCOM*, 2011.

[11] N. Krahnstoever, T. Yu, S. Lim, K. Patwardhan, and P. Tu. 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*, 2008.

[12] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu. SensEye: a multi-tier camera sensor network. In *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005.

[13] E. LAWLER. Generating all maximal independent sets : Np-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 1980.

[14] F. Licandro and G. Schembra. Wireless mesh networks to support video surveillance: architecture, protocol, and implementation issues. *EURASIP J. Wirel. Commun. Netw.*, Jan. 2007.

[15] K. Miyamoto. Fish eye lens. *JOSA*, 1964.

[16] V. Munishwar and N. Abu-Ghazaleh. Coverage algorithms for visual sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 9(4), 2013.

[17] V. Munishwar, S. Tilak, and N. Abu-Ghazaleh. Coverage management for mobile targets in visual sensor networks. In *Proc. of MSWiM*, 2012.

[18] F. Z. Qureshi and D. Terzopoulos. Surveillance camera scheduling: a virtual vision approach. In *VSSN '05: Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, 2005.

[19] S. Soro and W. Heinzelman. A survey of visual sensor networks. *Advances in Multimedia*, 2009.

[20] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6:505, 1977.

[21] Y. Wang and G. Cao. On full-view coverage in camera sensor networks. In *Proc. of IEEE INFOCOM*, 2011.