On Target Coverage in Mobile Visual Sensor Networks

Azin Neishaboori[†‡], Ahmed Saeed[†], Khaled A. Harras[‡], Amr Mohamed[†]
[†]Department of Computer Science and Engineering, Qatar University
[‡]School of Computer Science, Carnegie Mellon University Qatar
azin.neishaboori@gmail.com, ahmed.saeed@qu.edu.qa, kharras@cs.cmu.edu, amrm@ieee.org

ABSTRACT

Recent advancements in manufacturing low-cost wireless battery operated cameras has made their application in Wireless Video Sensor Networks (WVSN) increasingly more feasible and affordable. The application of robotic sensing agents equipped with cameras in WVSNs, seems particularly promising in performing coverage tasks for ad hoc surveillance. Their application in this context can be specifically useful for surveying areas with little to no available or affordable infrastructure, or where quick deployment is necessary. In this paper, we address the target coverage problem for finding the minimum number of cameras, their placement, and orientation to cover arbitrarily located targets in an area of interest. We propose a computationally light-weight heuristic, where the number of used mobile cameras is close to those found by near-optimal algorithms. Specifically, we address this problem for non-uniform target distributions that naturally form clusters. Having light-weight heuristics will be particularly useful when the application is required to adapt to target mobility and/or is implemented in embedded systems. Our simulation study shows that when clusters are sufficiently separated, the required number of cameras found by our proposed method is very close to those acquired by the near-optimal algorithm, whereas the computational complexity of our algorithm is about ten times less. We also deploy our algorithm on a drone testbed using off-the-shelf components to verify its feasibility.

Categories and Subject Descriptors

C.2.1 [**Computer Systems Organization**]: Network Architecture and Design—*Network topology*; I.2.9 [**Computing Methodologies**]: Robotics—*Sensors*

Keywords

Coverage; Visual Sensors; Mobile Cameras; Clustering

MobiWac '2014 Montreal Canada

http://dx.doi.org/10.1145/2642668.2642671.

1. INTRODUCTION

Manufacturing low-cost wireless battery operated cameras has become increasingly cheaper in the recent years, making them more affordable and feasible to adopt in a wide range of applications including environment monitoring, smart surveillance, and health care [2, 3, 28]. Wireless cameras are additionally useful in Visual Sensor Networks (VSNs) for the purpose of ad-hoc surveillance, where they are to survey areas which have little available infrastructure, or when it is required to establish rapid and/or temporary deployments [24]. In the recent years, Micro Air Vehicles (MAVs) (a.k.a. small/micro UAVs or microdrones), typically equipped with a variety of sensors, including cameras, have been considered to be used as wireless mobile cameras [4, 26–28, 30]. MAVs offer easier maneuverability and, thanks to their small size, are able to be positioned in locations, inaccessible to other forms of robots, that achieve optimal sensing coverage in outdoor [27] as well as indoor settings [5].

While using wireless mobile cameras for the purpose of smart surveillance involves addressing several challenging problems including area [1, 7, 13, 16, 20, 21, 31, 35] and target coverage [1,6,14,16,19,24,29,34], activity detection [29], and target tracking [38], in this paper we focus on target coverage. The problem of optimal camera placement to maximize coverage has been proven to be NP-complete in various forms for both area and target coverage, and for both isotropic [19] and anisotropic sensors [16]. It has therefore been simplified in different ways both in the robotics and sensor networks fields [8,15,36]. Common simplifications include space and/or camera pan discretization, and/or fixing camera locations. Approximation algorithms for anisotropic sensors have been proposed in the past [16, 19]. Unfortunately, despite such efforts, finding computationally efficient near-optimal algorithms for an arbitrary number of targets and/or in randomly-sized areas still remains a challenge. Having a computationally efficient camera placement method becomes particularly crucial when the application is required to adapt to targets mobility and/or when it is to be deployed on embedded systems.

Considering the need for computationally efficient algorithms for autonomous control of the mobile visual sensors, we propose an efficient heuristic algorithm for finding the minimum number of cameras to cover targets. Our algorithm yields a near-optimal number of required cameras when targets naturally reside in clusters each with one dominant direction of stretch. Throughout this paper we refer to such clusters as *directional clusters*. Examples of direc-

tional clusters in real-world applications include people lines in amusement parks, people taking tours of historical sites or attractions, and some animal herds. Our contribution in this paper is the proposition of our Cluster First algorithm (CF), a low-complexity two-step iterative algorithm for target coverage. In each iteration, we first divide targets into a number of clusters, and then apply our devised method to find the camera location and orientation for each directional cluster. If any target is left uncovered, we increase the number of clusters and iterate again. We refer to this proposed method for finding the location and orientation of a single camera to cover a group of targets by *Simple Cover-Set Coverage*, SCSC. Although performing these actions requires knowledge of all targets' locations in the area of interest, the required computations are light-weight and may either be performed centrally on one machine, or distributed on multiple camera-equipped robots, if communication amongst them is viable.

We evaluate our proposed algorithm via simulations conducted in MATLAB, and verify its feasibility using a simple testbed [32]. The simulation results show that, when targets form directional clusters, our algorithm offers up to ten times less computational complexity compared to other comparable algorithms, while requiring a number of cameras similar to those required by the near-optimal methods, e.g. only 5% more cameras on average in less dispersed directional clusters. We evaluate different scenarios on our testbed that involve a mixture of static and mobile targets, where a drone was able to modify its location and direction according to the CF algorithm and cover targets, in real-time, as the location of the mobile targets varied.

The rest of this paper is organized as follows. The related work is presented in Section 2. Section 3 is dedicated to describing the assumptions, notations, and definition of our problem. We present the details of our proposed algorithm in Section 4. In Section 5 we go over our performance evaluation. Finally, the paper is concluded in Section 6 where future work is also outlined.

2. RELATED WORK

While different applications require varying criteria for target coverage, a typical criteria is that at least one camera has the target in its view with acceptable quality. The definition of acceptable quality depends on the purpose of the application, i.e. detection, identification, or recognition [11]. This is the common assumption made in many coverage studies [1, 6, 14, 24, 34]. Some applications instead require a target to be fully angularly covered [36]. Herein, we assume that, if a target is seen by at least one camera from any angle, it is covered. While target coverage maximization has been shown to be NP-complete for most variants [8,37], many studies address this intractability by adding simplifying assumptions. The following two categories are typically considered for coverage problems:

(1) **Area Coverage Maximization:** This category is related to the classic Art Gallery problem [35]. The objective in this problem is to find the location of the minimum possible number of guards in an art gallery such that every point is seen by at least one guard. The Continuous-space Art Gallery problem has been proved both NP-Complete and APX-hard [31]. However, when space is discretized, polynomial time log-approximation solutions exist [7,13]. While guards can be considered as isotropic visual sensors, cameras have both a limited coverage range and Angle of View (AOV), i.e. are anisotropic, and thus make the coverage problem with cameras more complex. Even after discretization of space and camera pans, the problem solution remains unscalable [20]. Authors of [20], also propose a few heuristics which provide a coverage performance close to optimal, but with high computational complexity. A random set of deployed cameras have been used in [1] followed by posing the problem as finding the minimum subset of such cameras to be activated for attaining maximal coverage. This problem was also proven to be NP-Complete.

In addition to the studies conducted by the sensor networks community, coverage maximization using anisotropic and isotropic sensors has also been studied in the robotics community [9, 15, 17, 33] and [10]. In this community, it is typically assumed that the number of sensors are known, and the problem becomes how to move and direct sensors to optimize a coverage objective, such as an accumulation function. In contrast to visual sensor network applications, the coverage objective is typically different from area or target coverage. It is often the case that the location of targets are defined based on an assumed distribution density function. This function reflects a measure of probability that an interesting event takes place, or an object exists, at each location. Such function is either somehow known [22] or continuously learned using sensor readings [33]. Furthermore, in many of these studies, the sensor's sensing capability is assumed to follow a monotonically decreasing, continuous, and differentiable function both in radius and angle [18].

Our work is different from this category in the coverage objective. Our attempt is to provide maximum coverage for a given set of targets, rather than an area or an event/target distribution function. In our work, we consider dynamic camera deployments and attempt to find low complexity solutions for such scenarios.

(2) **Target Coverage Maximization:** Finding the minimal number of cameras and their optimal placement and orientation is a special case of this problem using any directional sensor. Using simplifications, solutions have been proposed in studies such as [1,6,14,24,34]. Fixed camera locations and discrete camera pans are assumed in [24], and a heuristic is proposed. A set of directional sensors, randomly scattered, are used in [6], a subset of which are selected one at a time. In [1], a similar problem is considered where an active set of camera sensors, and their directions, are selected from a lager pool of initially placed cameras. Using neighbor cooperation, fixed camera positions with variable pan/tilt are assumed in [34] to follow mobile targets. Rotating Directional Sensors (RDS) are utilized in [14] to maximize various coverage objectives. The optimal solution to these variants are shown to also be NP-complete, and therefore the authors propose heuristics.

Overall, the studies in this category assume either fixed camera locations or have them selected from a pool of previously selected random positions. These assumptions sets our study apart from the rest in this category. We introduced our first attempt to solve the problem in [25].

Note on Coverage versus Tracking: Coverage may be distinguished from tracking in that, the goal of coverage is to make sure a certain area or group of targets are covered by at least one camera at all times [38]. In that sense, the activities of each particular target over time is out of the scope of coverage.

3. OVERVIEW

In this section, we first state our assumptions regarding the cameras, targets, and the overall environment we consider in this paper. Afterwards, we provide a concise problem statement for our work.

3.1 Assumptions

Cameras and Camera Coverage: We assume the use of horizontal cameras with circular sector coverage areas as shown (highlighted in orange) in Figure 1. The radius of this sector indicates the maximum range of view of a camera, $R_{\max}$. The camera's Angle of View (AOV) is the angular width of this circular sector. We assume mobile cameras are mounted on agile ground or flying robots and are therefore able to move and redirect themselves to a certain location and orientation when asked.

Targets: We assume that all targets are located on the ground, or any 2D plane, as it is customary in target coverage problems [1, 24]. We also assume that objects or targets can be represented by points. To make coverage solutions more realistic, multiple points can be used to represent bigger sized targets. Also, we assume knowledge of the location of all targets [24]. This information may be acquired via e.g. using RFIDs as it is done in [38]. Alternatively, in the context of multi-layer sensor networks, this knowledge may be acquired via a higher tier camera providing low granularity coverage sufficient for detection and localization, but insufficient for identification, recognition, or activity monitoring [23]. Target T_j is covered by camera C_i if (1) its distance with the camera is less than or equal to $R_{\max}$, and (2) the line connecting C_i and T_j makes an angle less than half the camera's AOV with the direction of C_i. We assume that targets are either fully covered, or fully uncovered, i.e. no partial coverage. We do not consider occlusions in this paper and leave its consideration for future work.

Camera Configuration System: A wireless communication channel is required between the mobile cameras and the computational entities. If computations are performed on the camera-equipped robots, these robots are considered the computational entities. To avoid the necessity of multihop communication, and maintain simplicity, we assume that the coverage area is smaller than the wireless communication range when running experiments in this paper.

Remark on Centralized vs Distributed Implementation: If only one computationally capable entity acquires the location of all targets, it centrally calculates the location and orientation of all mobile cameras and informs them accordingly. Alternatively, if all mobile cameras are informed of the location of all targets, they each run the proposed algorithm individually and relocate and reorient themselves according to the outcome of the algorithm. If each mobile camera is statically assigned an ID, a mapping between these IDs and index in the generated camera configuration can be used to determine the new locations of all cameras. Optimizing this scheme in terms of energy and time includes making decision as to which mobile camera is to relocate to which one of the decided locations. Making such decisions necessitates path-planning, which we leave for future work.

3.2 Problem Statement

A set of targets reside in a two-dimensional plane. Using homogeneous horizontal cameras with a given maximum AOV and maximum coverage range $R_{\max}$, we want to find the minimum number of cameras, their location, and direction, such that each target is visible by at least one camera.

The above problem has been proven to be NP-Complete [19]. In this paper, we attempt to find a low-complexity heuristic solution for the case of non-uniform target distributions wherein targets naturally reside in a number of directional clusters.

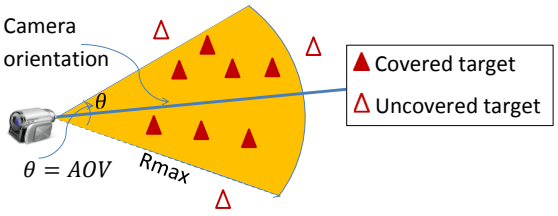


Figure 1: Camera coverage area

4. PROPOSED SOLUTION

In this section we describe the two steps of our iterative algorithm, Cluster-First (CF). The summary of this algorithm is depicted in Figure 2. In each iteration, we first divide the targets into a number, k, of clusters as described in subsection 4.1. We then use our proposed Simple Cover-Set Coverage method, SCSC, proposed in subsection 4.2 to find the location and orientation of the single camera that is to cover all targets in each cluster. As this task may not be possible, at the end of these steps, we calculate the number of targets that have been covered so far. If any target is left uncovered, we increase the number of clusters, k, using uniform steps, binary search, or a multiplicative-increase/decrease scheme, and re-iterate until the coverage criterion is met, i.e. all targets are covered. We may relax the coverage criterion to only require a minimum fraction of targets to be covered. Such criterion will result in a fewer number of required cameras which may be important, for instance when camera cost is an issue.

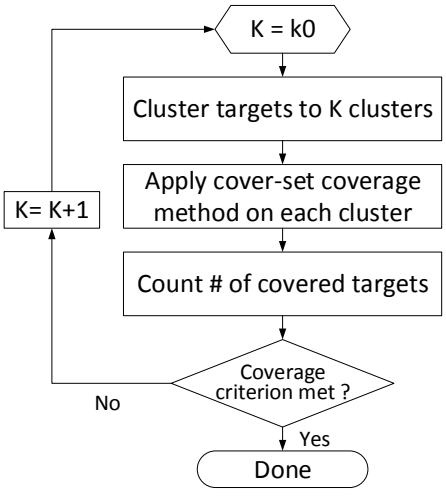


Figure 2: Our Cluster-First (CF) iterative method for finding the number, location, and orientation of cameras to cover targets

4.1 Step 1: Clustering

At each iteration of CF, the number of clusters, k, (and hence cameras) is given as input to an off-the-shelf unsupervised clustering algorithm. This algorithm divides the targets into k clusters. Subsequently the method described in subsection 4.2 is used on each such cluster to find the location and orientation of the camera to cover targets in this cluster.

4.2 Step 2: Covering targets in one cluster

Before delving into details of finding the location and orientation of the single camera allocated to each cluster, it is helpful to look at its input domain

Problem Space: An arbitrary co-planar set of targets is distributed in a fashion that falls somewhere between the following two extreme and degenerate cases: (1) All targets reside on one line. (2) Targets are evenly distributed in a circle. Now, let us consider a set of targets with a constellation that is somewhere between the two described above. We call the smallest (in the sense of area) ellipse containing all the targets in this set as ξ_{opt} with parameters a and b.

We now assume that targets' locations are randomly drawn from a $2D$ uniform distribution over this ellipse. Without loss of generality, we also assume that the average point coincides with the origin, and that the ellipse's major and minor axes are aligned with horizontal and vertical axis respectively, as shown in Figure 3.

4.2.1 Intuition on optimal camera direction:

Problem: Find the radial sector, with minimum possible radius R, its location and orientation to contain an ellipse originating at $(0, 0)$ with parameters a and b.

Solution: The two points P_1^1 and P_2^2 indicated in Figure 3 are both local optima. This can be seen easily by trying to deviate the radial sectors originating at these points, and directed at the major and minor axis respectively, either in location or direction, and observe that either will result in part of the ellipse being left uncovered.

For a given camera with sectoral coverage and a maximum AOV of θ, depending on the relationship between a, b, AOV, and $R_{\max}$, the following two camera configurations may be used to cover all targets: Place the center of a circular sector of angle θ on the (1) major, and (2) minor, axis of ξ_{opt}, such that the upper and lower boundaries are tangent to it. If the resulting location of the sector vertex is within 0 and $R_{\max}$ to all targets, this sector can cover them all. It can be shown that the maximum distance between the camera and the farthest possible target, $d^1_{\max}$ and $d^2_{\max}$ for the first and second configuration are as follows:

$$d^1_{\max} = a + \sqrt{a^2 + \frac{b^2}{\tan^2(\theta/2)}} \tag{1}$$

$$d^2_{\max} = b + \sqrt{b^2 + \frac{a^2}{\tan^2(\theta/2)}}$$

If both $d^1_{\max} \leq R_{\max}$ and $d^2_{\max} \leq R_{\max}$, either method could be used. Otherwise, method i with $d^i_{\max} \leq R_{\max}$ will be selected. Herein, the tie is broken in favor of the first configuration. The camera configuration decision for degenerate cases (1) and (2) are obtained using $a = b$ and $b = 0$ respectively. Note that for $a = b$, all camera directions are also equally suitable.

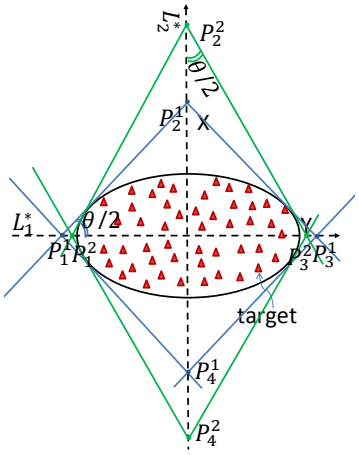


Figure 3: Finding camera location for a set of targets

Procedure for approximating a and b: Assume a given set of targets Λ with targets $T_k, k \in \{1, .., N\}$, and cameras with coverage range $R_{\max}$ and AOV $= \theta$. We find lines L_1^* and L_2^* for set Λ as follows. L_1^* is the line best describing the target points using the Mean Square Error (MSE) criterion. This line may also be found using Principal Component Analysis (PCA) [12] and finding the eigen vector, $u_{\max}$, corresponding to the largest eigen value, $\lambda_{\max}$. We denote the line perpendicular to L_1^* by L_2^*. Since there are a maximum of two eigen values in a 2D plane, this line corresponds to the other eigen value, $\lambda_{\min}$. We denote the standard deviation of target locations projected on L_1^* and L_2^* by σ_1 and σ_1 respectively. Next, we interpret $\frac{\lambda_{\max}}{\lambda_{\min}} \approx \frac{a}{b}$. We set $a := 2\sigma_1$ and find $b = 2\sigma_1 \times \frac{\lambda_2}{\lambda_1}$.

Plugging the values of a and b obtained from the above procedure in equation 1, we decide between L_1^* and L_2^* for the camera orientation. We denote the decided camera orientation by L^*.

4.2.2 Simple Cover-Set Coverage Method (SCSC)

Assume a given set of targets Λ with targets $T_k, k \in \{1, .., N\}$, and cameras with coverage range $R_{\max}$ and AOV $= \theta$. We denote the slope of line L^*, obtained in 4.2.1, relative to the x-axis by α. In our proposed method, SCSC, the location of the camera is obtained as shown in Figure 4 and is explained as follows.

(i) Through each point T_k, we pass two lines: (1) line L_1^k making angle $\alpha + \theta/2$ with respect to the x-axis, (2) line L_2^k making angle $\alpha - \theta/2$ with respect to the x-axis.

(ii) We select the outermost line in the resulting grid, i.e. we pick the lines where all target points reside only on one side of them. There will be 4 such lines: (1) $L_1^{\min} = \min L_1^k, \forall k \in \{1, .., N\}$, (2) $L_1^{\max} = \max L_1^k, \forall k \in \{1, .., N\}$, (3) $L_2^{\min} = \min L_2^k, \forall k \in \{1, .., N\}$, and (4) $L_2^{\max} = \min L_2^k, \forall k \in \{1, .., N\}$. (iii) Two intersection points $P_2 = L_1^{\max} \times L_2^{\min}$ and $P_4 = L_1^{\min} \times L_2^{\max}$ both meet the AOV requirements. Between them, we select the one that covers most targets, and break ties randomly. Note that for $\theta = \pi/2$, points P_1 and P_3 also meet the AOV requirements. Also note that this takes $O(N)$ operations for N targets.

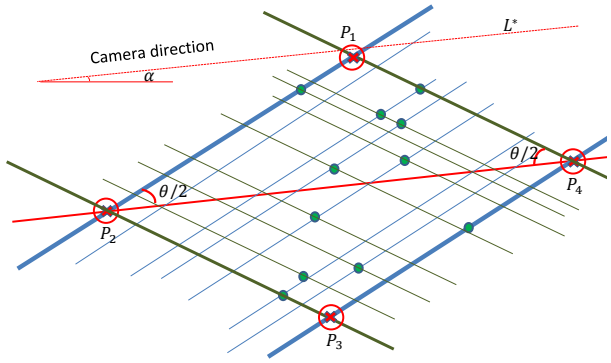


Figure 4: SCSC method: Targets are the green dots. The decided camera direction is α (the red line). Two lines which make $\theta/2$ (green lines) and $-\theta/2$ (blue lines) with the camera direction are passed through each target. These "outer-most" lines (bolded) intersect at four points P_1 to P_4.

Complexity of CF: The computational complexity of this algorithm depends on the complexity of the clustering algorithm. If, for instance, k-means clustering is used, which is linear in both dimensions and number of targets [12], CF will also be an $O(N)$ method.

Remarks on Target Mobility:
Periodic or event-based invocations of the coverage algorithms can be used to cover mobile targets. However, the time it takes for recalculating the location and orientation of the cameras, in addition to relocating and repositioning the cameras, must be less than the time interval between consecutive invocations. This necessitates low-complexity and fast coverage algorithms.

5. EVALUATION

In this section, we first present the evaluation we conducted via simulation on Matlab. Afterwards, we share the experience gained as a result of our preliminary experimental evaluation on our testbed using UAV quadcopter drones.

5.1 Simulation

5.1.1 Simulation Set-Up

We use MATLAB simulations to compare the performance of our proposed algorithm against those of methods previously proposed in the literature. We generate 8 directional clusters of targets. In each of these clusters, the y-coordinates of target locations are dispersed along the defined cluster direction using a normal distribution with standard deviation σ. Low values of σ result in well-defined directional clusters, while large values of it have the opposite effect, as depicted in Figure 5 for 50 targets. For each value of σ, we generate 10 random scenarios in this manner and average the acquired performance metrics over all scenarios.

We used two heuristic algorithms to compare our proposed algorithm against. Both of these algorithms were selected from a few proposed in [20] and were modified to solve target coverage instead of area coverage. The first one is *greedy search*, which is the closest to optimal in terms of the number of cameras it finds covering all targets, but has the highest computational complexity. The second algorithm is *dual-sampling*, which is the most computationally efficient algorithm proposed in [20]. To the best of our knowledge, there are no other heuristic algorithms that use similar assumptions and would allow us to compare our methods against. Other heuristics to the area and target coverage make additional assumptions about the location of cameras, and/or location of targets, which makes it difficult to conduct a meaningful comparison between those method and ours.

Parameter	Range	Nominal value
Dim	$50m \times 50m$	$50m \times 50m$
AOV	$\in \{45° - 150°\}$	$90°$
Target count	$50 - 200$	50
$R_{\max}$	$6m - 20m$	$15m$
σ	$0.5 - 4.5$	1

Table 1: Range and nominal values for simulation parameters

In the following, we explain how these two heuristics operate. In greedy search, cameras are positioned and oriented one at a time. The location and direction of each additional camera is determined by calculating the rank of all possible pairs of camera location-direction. The rank of each location-direction pair is calculated by the number of remaining targets it is able to cover. Hence, this algorithm needs $O(D^2 \frac{2\pi}{\Delta\phi} N^2 \log N)$ calculations and $O(D^2 \frac{2\pi}{\Delta\phi})$ stored elements, where D, N and $\Delta\phi$ are the dimensions of the square-shaped area, the number of targets, and the pan step, respectively. In dual-sampling, each time, one target is randomly selected. Then the camera location is selected from the area in the $R_{\max}$ neighborhood of this target. Afterwards, the location and direction pair which has the highest rank that can also cover the chosen target is picked. Therefore, this algorithm needs $O(R_{\max}^2 \frac{2\pi}{\Delta\phi} N^2 \log N)$ calculations and $O(R_{\max}^2 \frac{2\pi}{\Delta\phi})$ stored elements.

Dual-sampling and greedy search require very similar number of cameras to cover all targets in all our scenarios, which we attribute to the open square-shaped are of interest defined with no walls or hallways. However, greedy search consumes a lot more execution time due to its higher complexity.*In the figures displaying execution time throughout this section, we exclude the greedy search results since its values are very large ($\frac{D^2}{R_{\max}^2}$ times more than dual-sampling) and including them would skew the scale of the figure and mask the difference in the performance of other algorithms.*

We apply two specific clustering algorithms as representatives of centroid and non-centroid based clustering algorithms: kmeans, and kd-tree [12], and denote them by CF-kmeans and CF-tree. We also denote the greedy and dual-sampling methods by Gd and Dual-Smp respectively.

Parameters and Metrics: The parameter values are summarized in Table 1. We consider the following metrics: (1) Number of required cameras, and (2) Execution time.

5.1.2 Simulation Results

Impact of cluster dispersiveness: We vary the dispersiveness of directional clusters, depicted in Figure 5, by changing the value of parameter σ, as described earlier in this section. The performance of the different methods is compared in Figure 6. From Figure 6(a), we notice that

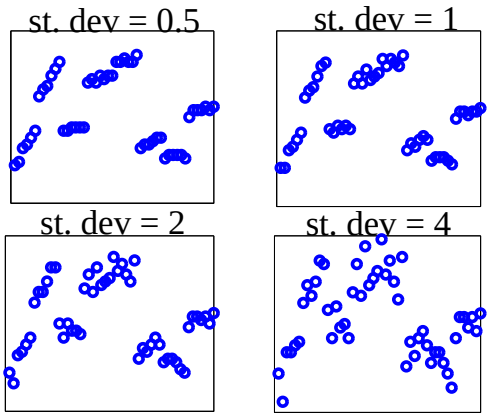
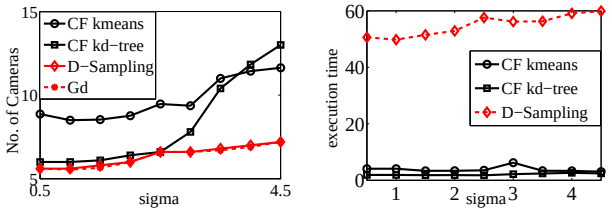


Figure 5: Target constellation samples for 4 values of σ: 0.5, 1, 2, 4



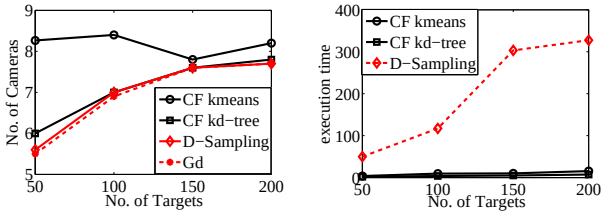
(a) Camera Count vs. σ (b) Complexity vs. σ

Figure 6: Comparison between camera placement algorithms for different cluster dispersiveness levels

when clusters are less dispersed, i.e. σ is not large, the number of cameras obtained by CF-tree is very close to those obtained via Dual-Smp and Gd. As the dispersiveness increases, the clusters become less defined, and therefore the underlying logic for our proposed SCSC method becomes less effective. This causes CF-tree to perform worse as dispersiveness increases. We also notice that CF-tree requires a fewer number of cameras compared to CF-kcam when σ is small. This is because we set our clusters to be stretched mainly in one direction, which makes centroid-based clustering methods such as k-means less effective. On the contrary, kd-tree is capable of identifying non-circular shaped clusters effectively. However, when σ is large and therefore clusters are less defined, CF-kmeans performs better than CF-tree. In terms of computational complexity, as can be seen in Figure 6(b), both CF variants perform much better than Dual-Smp and Gd.

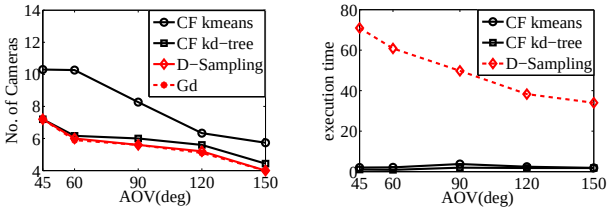
Impact of target density: Figure 7 compares the performance of the various methods for different number of targets (and hence density, since the area is fixed). From Figure 7(a), we notice that again the number of cameras required to cover targets by CF-tree is very similar to those obtained by Gd and Dual-Smp. We also observe that this measure converges for all methods as the number of targets increases. On the contrary, from Figure 7(b), we see that while the execution time for CF-variants only slowly grows, it experiences a large increase for Dual-Smp. This is in agreement with the computational complexities calculated.

Impact of AOV: The performance of the algorithms were compared for a number of different values for AOV:



(a) Camera Count vs. target density (b) Complexity vs. target density

Figure 7: Comparing camera placement algorithms with different target densities



(a) Camera Count vs. AOV (b) Complexity vs. AOV

Figure 8: Comparing camera placement algorithms with different AOVs

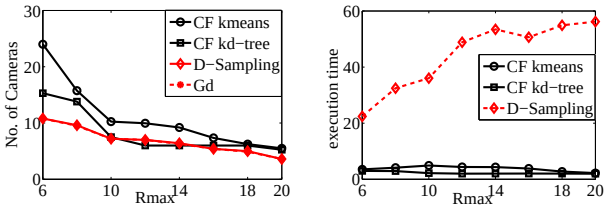
45° (unzoomed webcam), 60° (vertical camera of Parrot AR.Drone), 90° (frontal camera of Parrot AR Drone), 120 and 150 (fish-eyed lenses). Figure 8 depicts the results. As expected, we notice from Figure 8(a) that the number of required cameras for all four algorithms drops (and converges) for wider AOV values. Also, the execution time of the different algorithms decreases as can be seen in Figure 8(b). Again, FC variants offer much smaller computational complexity than Dual-Smp and Gd (Gd eliminated again for easier comparison).

Impact of $R_{\max}$: We would like to make the following note on the reasonable range of choices for $R_{\max}$ before presenting the result of its impact on the performance of the different algorithms:

Note 1: For some range of values of $R_{\max}$ the target coverage problem may be trivially solved. Let us consider the quantity *covering-density* for K cameras with range $R_{\max}$ and $AOV = \theta$ as defined in [16] as:

$$covering\ density = \frac{K \times \pi R_{\max}^2 (\frac{\theta}{2\pi})}{D^2}$$

in which the nominator quantifies the direct sum of all sub-areas covered by all cameras, and the denominator quantifies the total surface area. As this quantity does not account for overlapping covered areas across cameras, it is a conservative measure of coverage. Nevertheless, if this quantity adds up to one, all the given area is covered, in which case, the target coverage problem is trivial. For example, assume that we are to cover a 50m×50m square area using cameras that have an $AOV = \pi/2$, and $R_{\max} = 40$m. Positioning one camera at every corner of this square and directing it according to the square's diagonals towards the center covers the whole area, and all the targets that are contained in it.



(a) Camera Count vs. $R_{\max}$ (b) Complexity vs. $R_{\max}$

Figure 9: Comparing camera placement algorithms with different coverage ranges

Considering the above note, we vary the value of $R_{\max}$ between 6m and 20m, to ensure a nontrivial coverage problem. Note that for a given camera type, this quantity also captures the quality images a specific application requires. The results are shown in Figure 9. As expected, it can be seen in Figure 9(a) that the number of required cameras decreases (and converges) for all four algorithms when $R_{\max}$ increased. As seen in Figure 9(b), while the FC variants' execution time decreases as $R_{\max}$ increases, D-Smp (and Gd whose result is not shown to make easier comparison) exhibit second-order polynomial increase in execution time.

Note 2: We can scale the $R_{\max}$ and area dimensions based on the application and the required camera coverage quality. For instance, the simulation results presented for our 50m ×50m simulation area and $R_{max} = 5$ may be down-converted to a 10m ×10m area at $R_{max} = 1$m, or up-converted to a 100m ×100m area and $R_{\max} = 10$.

5.2 Preliminary Experimental Evaluation

Testbed: We show the feasibility of the proposed algorithm using simple experiments. In our experiments, micro Unmanned Air Vehicles (UAVs), mounted with a front camera, move around the area of interest to cover toy targets within an indoor setup. Figure 10 shows the different components of the testbed.

The testbed setup covers an area of $1.5m$ by $2.25m$. To localize the different moving objects in the area of interest (i.e. targets and UAVs), each object is labelled with a certain color and an Axis 213 PTZ Network Camera was mounted at $3.8m$ height to locate the different objects. The locations of all objects are reported to the central node, a laptop, to calculate the placement strategy of the UAVs.

We use the Parrot AR Drone 2.0 quadcopters as UAVs. These drones are equipped with two cameras: a front 720p camera with a 93° lens and a vertical QVGA camera with a 64° lens. Those drones are controlled through an external computer through WiFi, and are also equipped with an on board ARM Cortex $A8$ processor with 1 Gbit RAM that runs a Linux 2.6.32 Busybox. The main purpose of the on board computing power is to report the state of the drone and collected video to the external computer and provide assistance on basic manoeuvres. The testbed is shown in Figure 11. More details on this testbed may be found in [32].

Experiment: Our deployed scenario includes one drone and four targets, of which two are static, and the other two are remote controlled mobile cars. The algorithm runs on a laptop, and the drone is commanded to adjust its location and orientation periodically (maximum of 3 times per second) to maximize coverage. Initially, all four targets are

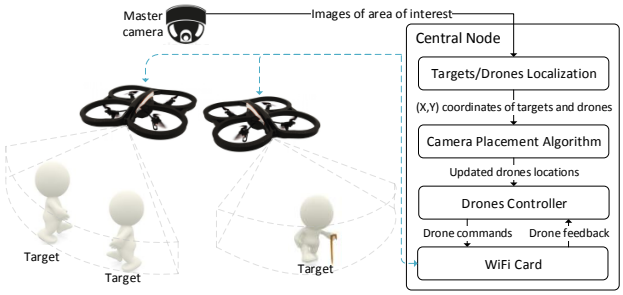


Figure 10: The experimental testbed architecture

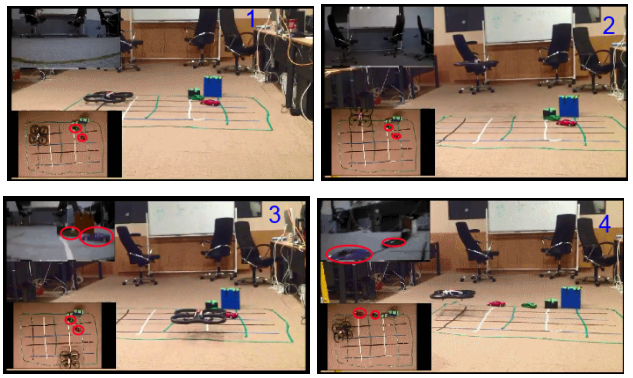


Figure 11: A UAV coverage scenario using the AR Parrot Drones. The view from the UAV is in the top left corner of each snapshot, and the view from the master camera is in the bottom left corner of each snapshot. The general view is that taken from our camera of the whole environment. Targets that become visible by either UAV cameras, or master camera are circled in red.

static, and the drone, located in an arbitrary spot, takes off, moves to the location prescribed by the CF algorithm, adjusts its direction and consequently covers all targets. Then the two remote-controlled cars move one at a time, and the drone adjusts its location/direction to maximize coverage. In the demo, we change the locations of the cars such that one camera is sufficient to cover all four targets.

6. CONCLUSION AND FUTURE WORK

In this paper, we studied the problem of finding the number of cameras, their location, and orientation such that all targets are covered. We proposed a computationally efficient heuristic algorithm based on iteratively clustering targets, which finds the location/direction of a single camera to cover targets in each individual cluster. We used simulation to evaluate and compare our proposed method's performance against those of existing work. Our simulation results showed that, when targets are naturally spread in directional clusters, the number of cameras our method finds are very close to those obtained by near-optimal methods, while requiring much less computational complexity. To show the feasibility of our algorithm, we also deployed a simple testbed and used our proposed algorithm to au-

tonomously position and orient Parrot AR.Drone MAVs, in real-time, to cover a set of static and mobile targets.

In the future, we will address occlusions, distributed heuristics, and path planning for mobile cameras relocation considering energy consumption. A more comprehensive testbed will also be used with more complex scenarios.

7. ACKNOWLEDGEMENT

This work was made possible by NPRP grant $\#4-463-2-172$ from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

8. REFERENCES

[1] J. Ai and A. Abouzeid. Coverage by directional sensors in randomly deployed wireless sensor networks. *Combin. Optim.*, 11(1):21–41, June 2006.

[2] I. Akyildiz, T. Melodia, and K. Chowdhury. A survey on wireless multimedia sensor networks. *Computer Networks*, 51(4):921 – 960, Dec 2007.

[3] H. Alemdar, Y. Durmus, and C. Ersoy. Wireless healthcare monitoring with RFID-enhanced video sensor networks. *Distributed Sensor Networks*, 10(1):290–300.

[4] J. Allred, A. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, and K. Mohseni. An airborne wireless sensor network of micro-air vehicles. pages 117–129. ACM SenSys, Nov 2007.

[5] R. W. Beard and T. W. McLain. Multiple uav cooperative search under collision avoidance and limited range communication constraints. pages 25–30. IEEE CDC, Dec 2003.

[6] Y. Cai, W. Lou, M. Li, and X.-Y. Li. Target-oriented scheduling in directional sensor networks. pages 1550–1558. IEEE Infocom, May 2007.

[7] O. Cheong, A. Efrat, and S. Har-Peled. Finding a guard that sees most and a shop that sells most. *Discrete Comput. Geom.*, 37(4):545–563, June 2007.

[8] D. G. Costa and L. A. Guedes. The coverage problem in video-based wireless sensor networks: A survey. *Seonsors*, 10:8215–8247, Sept 2010.

[9] R. Dirza and A. Gusrialdi. Performance-guaranteed distributed coverage control for robotic visual sensor network with limited energy storage. pages 334–339. Intl Conf. on Instrumentation Control and Automation, Nov 2011.

[10] R. Dirza and A. Gusrialdi. Performance-guaranteed distributed coverage control for robotic visual sensor network with limited energy storage. In *ICA*, Nov 2011.

[11] J. Donohue. Introductory review of target discrimination criteria. Technical Report PT-TR-92-2129, US Airforce Systems Command, Phillips Laboratory, Dec 1991.

[12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification - second edition.* John Wiley and Sons Inc., NY, 2001.

[13] A. Efrat and S. Har-Peled. Guarding galleries and terrains. pages 181–192. Information Processing Letters, July 2006.

[14] G. Fusco and H. Gupta. Placement and orientation of rotating directional sensors. pages 1–9. IEEE SECON, Nov 2010.

[15] A. Gusrialdi, T. Hatanaka, and M. Fujita. Coverage control for mobile networks with limited-range anisotropic sensors. pages 4263–4268. IEEE CDC, Dec 2008.

[16] X. Han, X. Cao, E. L. Lloyd, and C. Shen. Deploying directional sensor networks with guaranteed connectivity and coverage. In *Proceedings of the IEEE SECON*, June 2008.

[17] B. Hexsel, N. Chakraborty, and K. Sycara. Coverage control for mobile anisotropic sensor networks. IEEE Intl Conf. on Robotics and Automation, May 2011.

[18] B. Hexsel, N. Chakraborty, and K. Sycara. Distributed coverage control for mobile anisotropic sensor networks. Technical Report CMU-RI-TR-13-01, Robotic Inst, Carnegie Mellon University, PA, Jan 2013.

[19] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *J. ACM*, 32:130–136, 1985.

[20] E. Horster and R. Lienhart. On the optimal placement of multiple visual sensors. pages 111–120. VSSN, Oct 2006.

[21] E. Horster and R. Lienhart. Approximating optimal visual sensor placement. pages 1257–1260. IEEE Intl Conf on Multimedia and Expo, Nov 2010.

[22] T. K. J. Cortes, S. Martinez and F. Bullo. Coverage control for mobile sensing networks. *IEEE Trans. on Robotics and Automation*, 20(2):243–255, Apr 2004.

[23] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu. Senseye: a multi-tier camera sensor network. Number 1, pages 71–77. ACM Multimedia, November 2005.

[24] V. P. Munishwar and N. B. Abu-Ghazaleh. Coverage algorithms for visual sensor networks. *ACM Trans on Sensor Networks*, 9(4):1–34, July 2013.

[25] A. Neishaboori, A. Saeed, A. Mohamed, and K. Harras. Target coverage heuristics using mobile cameras. *International Workshop on Robotic Sensor Networks*, 2014.

[26] J. Nemeroff, L. Garcia, D. Hampel, and S. DiPierro. Application of sensor network communications. pages 336–341. IEEE MILCOM, Oct 2001.

[27] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner. Networked UAVs as aerial sensor network for disaster management applications. *e & i Elektrotechnik und Informationstechnik*, 127(3):56–63, 2010.

[28] M. Quaritsch, E. Stojanovski, C. Bettstetter, G. Friedrich, H. Hellwagner, B. Rinner, M. Hofbaur, and M. Shah. Collaborative microdrones: applications and research challenges. page 38. Intl Conf on Autonomic Computing and Communication Systems, Sep 2008.

[29] M. J. P. S. R. Bodor, A. Drenner and N. Papanikolopoulos. Mobile camera positioning to optimize the observability of human activity recognition tasks. IROS, Aug 2005.

[30] P. Rudol, M. Wzorek, G. Conte, and P. Doherty. Micro unmanned aerial vehicle visual servoing for cooperative indoor exploration. pages 1–10. IEEE Aerospace Conference, Mar 2008.

[31] C. S. S. Eidenbenz and P.Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):181–192, Nov 2001.

[32] A. Saeed, A. Neishaboori, A. Mohamed, and K. Harras. Up and away: A cheap uav cyber-physical testbed, work in progress. *arXiv preprint arXiv:1405.1823*, 2014.

[33] M. Schwager, M. P. Vitus, D. Rus, and C. J. Tomlin. Robust adaptive coverage for robotic sensor networks. In *Proceedings of the International Symposium on Robotics Research (ISRR 11)*, August 2011.

[34] C. Soto, B. Song, and R. Chowdhury. Distributed multi-target tracking in a self-configuring camera network. page 1486âĂŞ1493. IEEE CVPR, Nov 2009.

[35] J. Urrutia. Art gallery and illumination problems. *Handbook of Computational Geometry*, 1(1):973–1027, 2000.

[36] E. Yildiz, K. Akkaya, E. Sisikoglu, and M. Y.Sir. Optimal camera placement for providing angular coverage in wireless video sensor networks.

[37] M. Younis and K. Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6(4):621–655, June 2008.

[38] X. Yu. *Hybrid radio frequency and video framework for identinity-aware augmented perception in disaster management and assistive technologies.* PhD thesis, UMass Amherst, MA, 2013.