Target Coverage Heuristics Using Mobile Cameras

Azin Neishaboori[1] and Ahmed Saeed[2] and Amr Mohamed[3] and Khaled Harras[4]

***Abstract*—The availability of low-cost mobile robots with sensing, communication, and computational capabilities has made feasible a new class of Cyber-Physical Systems (CPS). These mini-CPSs may be used where quick, low-cost or non-lasting visual sensing solutions are required, e.g. in border protection and disaster recovery. In this paper, we take the first steps towards a fast and automated CPS. We consider the problem of low complexity placement and coordination of an unknown number of mobile cameras to cover arbitrary targets. We address this problem as an unsupervised clustering task. A set of proximal targets are clustered together, whereas the camera location/direction for each cluster are calculated/estimated individually. Our proposed solutions provide centralized computationally efficient heuristics using two clustering-based algorithms: k-camera clustering, and cluster-first algorithms. Our evaluation shows that the required number of cameras approach those obtained via near-optimal methods as the cameras' coverage range, angles of view, or the number of targets increase.**

I. INTRODUCTION

The availability of low-cost mobile robots with sensing, communication, and computational capabilities has made feasible a new class of mini-CPSs which can make their deployment simpler, faster, and more easily reconfigurable. Particularly, Micro Air Vehicles (MAVs) (a.k.a. microdrones or small/micro UAVs), typically equipped with cameras, have been proposed to be used as mobile cameras [1], [2] in Wireless Video Sensor Networks (WVSN). One of the main advantages of using MAVs is their maneuverability and small size which enables them to be placed in locations that achieve optimal sensing coverage in both indoor [3] and outdoor scenarios [1]. Among the many applications of such Wireless Mobile Video Sensor Networks (WMVSN) are environment monitoring, smart surveillance, traffic management and healthcare where quick, low-cost or non-lasting solutions are required [1], [4], [5]. While research in camera surveillance has tackled many challenges such as tracking and activity detection, it has largely focused on the more fundamental challenge of area or target coverage. The problem of optimal camera placement to maximize coverage has been shown to be NP-complete in many variations for both area and target coverage. Therefore, this problem has been simplified in many forms to address optimal sensor placement using isotropic and unisotropic sensors in the field of robotics and sensor networks, e.g. [6], [7]. With numerous solutions for area coverage [8]–[12] and target coverage [12]–[17], typical simplifications include fixing camera locations, or discretizing space and camera pan. Despite all these efforts, finding a fast and computationally efficient algorithm for arbitrary number of targets has remained a challenge.

In this paper we take the first steps towards developing an automated and fast surveillance system using mobile cameras. We propose computationally efficient heuristics for mobile camera placement and orientation for the coverage of a set of localized targets. Our objective is motivated by the need for efficient algorithms for autonomous control of the mobile cameras due to the limitations in their energy and computational capabilities (relative to static cameras). The most significant distinction between our methods and those of others in this area, is our statistical approach. That is, rather than attempting to find a solution for camera location/direction for each specific target constellation, we aim for a solution that fits a large number of scenarios.

We consider the problem of finding the minimum number of cameras to cover a high fraction of a set of targets as a clustering problem. We address this problem as an unsupervised clustering task and find simple and efficient solutions using classification techniques. We propose two algorithms. (1) K-Camera clustering: we iteratively cluster targets coverable by one camera, and then update the cameras locations until convergence is achieved. (2) cluster-first method: we first cluster proximal targets together, and then find the camera location/direction for each cluster.

We evaluate our proposed algorithms via simulation in MATLAB. We observe that our methods offer significantly lower computational complexity, up to 100 times. However, they require more cameras to cover a pre-determined fraction of targets, e.g. 0.9. As the number of targets, camera coverage ranges, and their maximum Angle Of View (AOV) increase, the required number of cameras in our methods becomes increasingly closer to those of near-optimal methods.

II. SYSTEM ASSUMPTIONS

Targets: We adopt the conventional simplifying assumption that targets reside on a 2D plane [12], [13]. We assume that target locations are known, which allows us to represent a bigger sized object of interest by multiple targets. This knowledge can be gained by higher tier cameras, common in multi-tier visual sensor networks, used only for detection and localization, and communicated to all lower tier cameras [18], or using RFIDs [19].

Cameras and Camera Coverage: Cameras are horizontal and their area of coverage are circular sectors, as shown (highlighted in orange) in Figure 1a. The radius of this sector is the depth of view of a camera, R, where $0 \leq R \leq R_{\max}$ within which the captured image is considered of acceptable sharpness and quality. The angular width of this sector is the Angle of View (AOV) of the camera which is approximately inversely proportional to the lens's focal length. Mathematically, if (1) a target is within $R_{\max}$ distance of the camera, and (2) the angle between camera C_i and target T_j relative to the orientation of the camera is within the AOV of the

*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.

[1]Azin Neishaboori is a post doctoral researcher at the ECE Department of Qatar University. azin.neishaboori@gmail.com

[2]Ahmed Saeed is a research engineer at the ECE Department of Qatar University. ahmed.saeed@qu.edu.qa

[3]Amr Mohamed is an assistant professor at the ECE Department of Qatar University. aamrm@ieee.org

[4]Khaled Harras is an associate professor at the School of CS in Carneige Mellon University Qatar. kharras@cs.cmu.edu

camera, target T_j is covered by camera C_i. We impose hard constraints on coverage of a target: either completely covered or not covered at all. We do not consider occlusions, as we assume the use of camera-equipped ground robots or MAVs as mobile cameras, with cameras above the targets $2D$ plane. We assume mobile cameras which are capable of following a command to position and orient themselves as prescribed. We assume a lower bound on the percentage of targets that are to be covered, and refer to this bound as Coverage Termination Criterion (CTC). We also pose an upper bound on the number of cameras.

Camera Configuration Commands and Medium: We assume that a centralized computational entity calculates the position and direction of all mobile cameras. This entity can be one of the mobile cameras or a separate entity. We also assume that, there is a wireless channel between this central entity and the mobile cameras. We assume coverage areas are smaller than the communication range. Commands are sent via the wireless medium to each mobile camera to inform it of the new location and orientation to move to.

III. Proposed Solutions

Before we delve into our proposed solutions, we first formally define our problem and introduce a definition which will be used in the rest of this section.

Problem Statement: *Given a set of co-planar targets in a two-dimensional plane and using homogeneous horizontal cameras with a given maximum AOV and maximum coverage range $R_{\max}$, find the minimum number of cameras, their location and orientation such that the fraction of targets visible by at least one camera is equal to or greater than a pre-determined value.*

Definition: *A group of targets, $T_i, i \in \{1..N\}$ form a cover-set with respect to a camera with given $R_{\max}$ and AOV if it is feasible to cover all of them by one camera.*

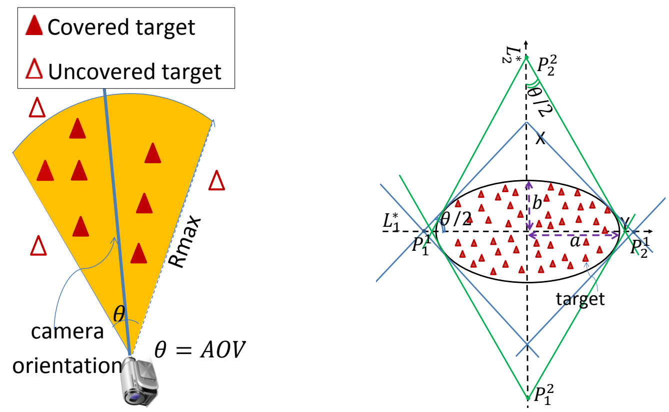
In the subsections that follow, we first describe the basis of our solution. We then utilize this basis to find the location/direction of one camera for a *cover-set*. We then propose two methods, *K-Camera clustering* and *cluster-first algorithm*, to divide arbitrary set of targets to a "small" number of cover-sets.

A. Coverage for a cover-set

The distribution of an arbitrary co-planar set of targets falls between the following two extreme and degenerate cases: (1) all targets reside on one line, and (2) targets are evenly distributed in a circle. Let us consider a cover-set with a constellation that is somewhere between the two described above. We call the smallest (in the sense of area) ellipse, with parameters a and b, containing all the targets within the given cover-set as ξ_{opt}, as shown in Figure 1b.

For non-degenerate cases, given a camera with sectoral coverage and maximum AOV of θ, depending on the relationship between a, b, AOV and $R_{\max}$, the following two camera configurations are proposed 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} (along L_1^* and L_2^*), such that the upper and lower boundaries are tangent to it. The



(a) coverage model (b) coverage for a cover-set

Fig. 1. Camera coverage model and placement/direction for a cover-set

possible camera locations are shown as P_1^1 and P_2^1 on the major axis and as P_1^2 and P_2^2 on the minor axis. If the resulting location of the sector vertex is within zero and $R_{\max}$ to all targets, this sector can cover them all. Using geometry one can prove that the maximum distance between the camera and the farthest possible target, $d_{\max}^1$ and $d_{\max}^2$ for the first and second configuration are as follows: $d_{\max}^1 = a + \sqrt{a^2 + \frac{b^2}{\tan^2(\theta/2)}}$ and $d_{\max}^2 = b + \sqrt{b^2 + \frac{a^2}{\tan^2(\theta/2)}}$. If both $d_{\max}^1 \leq R_{\max}$ and $d_{\max}^2 \leq R_{\max}$, either method could be used. Otherwise, method i with $d_{\max}^i \leq R_{\max}$ will be selected. 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 orientations are equally suitable.

B. K-camera clustering

This method starts with an initial set of clusters formed based on target proximity, and moves targets to adjacent clusters until convergence is reached at which point *ideally* all clusters form cover-sets. At every stage, the cluster's camera location/direction is found using the method described in subsection III-A. The algorithm's pseudo-code is depicted in Algorithm 1. In line 3, $C_{\max}$ is the maximum number of

Algorithm 1 Algorithm k-camera clustering

```
1: k → k_0
2: Divide given area to k arbitrary clusters.
3: while coverage fraction ¡ CTC and k < C_max do
4:     k → k + 1, m → 0
5:     while Convergence not met and m < M do
6:         for all clusters do
7:             Find location and pose of camera for cluster i using method in
               III-A
8:             for all uncovered targets in cluster i do
9:                 find the "best" adjacent cluster
10:            end for
11:        end for
12:        m → m + 1
13:    end while
14: end while
```

clusters allowed. In line 5, the convergence criterion is based on the number of transitions of targets from one cluster to the other, while an upper bound of M specifies the maximum number of iterations allowed. In line 9, the decision on which cluster to move an uncovered target to can be made as follows. First we identify the adjacent clusters whose cameras can cover the target at their current position and coordination. If more than one choice is available, we select the adjacent cluster which covers the target with the highest angular confidence. If no qualifying cluster exists, the target remains in its current cluster.

This algorithm's complexity can be shown to be $O((K - k_0)NM)$, where N, k_0 and K are the number of targets and the initial and final number of cameras (obtained from the algorithm above) respectively.

C. *Cluster-First Algorithm*

An alternative to the K-Camera clustering method is to run an (off-the-shelf) unsupervised clustering algorithm and divide the targets into subsets. Once a set of clusters are obtained, we handle them as if they form a cover-set and use the method described in subsection III-A to find the location/direction of the camera for each cluster. If polar coordinates are used instead of Cartesian coordinates, scaling is required to balance angle and magnitude values. The computational complexity of this method also depends on the complexity of the clustering algorithm. Again, the number of clusters (hence cameras) are increased till the minimum required fraction of targets are covered. If k-means clustering is used for example, which is linear in both dimensions and number of targets [20], cluster-first method would also be an $O(N)$ method.

IV. PERFORMACE EVALUATION

We use MATLAB simulations to compare our proposed methods' performance against those of previously proposed in the literature [11]. We generate location of targets randomly using a *uniform distribution* over the given area. We generate 10 random scenarios in this manner and average the performance metrics obtained in each scenario.

We compare our work against two heuristic algorithms, amongst several proposed in [11] and modify them to cover specific targets instead of a whole area: greedy search, which is the closest to optimal in coverage, but is most computationally demanding method, and dual-sampling, which is their most computationally efficient method proposed. To the best of our knowledge there are no other heuristic algorithms which use similar assumptions and would allow us to compare our methods against them.

The greedy algorithm in [11] is based on the idea of placing sensors one at a time. The selection of position and orientation for each additional sensor is decided based on the rank of all possible location-orientation combinations. The rank of each position/orientation combination is determined by how many remaining targets it can cover. In the dual-sampling method, one target is randomly selected at a time. The area from which the camera location is chosen is limited to the $R_{\max}$ vicinity of this targets. Then from all possible camera location-orientation combinations, the one with the highest rank that can cover the selected target is chosen. In this section, k-camera clustering, cluster-first method, greedy and dual-sampling methods are referred to by KCam, CF, Greedy and 2Smp respectively.

Parameters: The parameters are summarized in Table I. We also set the maximum number of cameras allowed to half the number of targets, however since this limit is never reached it is not included in the table. The parameter $c_{\theta\to r}$ is the coefficient used to balance between the values of angular and Euclidean distance from origin (polar coordinates) for each target. We assign it 50 empirically as it showed reasonable but non-optimal results to reflect more realistic settings where fine-tuning is not performed per scenario. For most scenarios discussed in this section, KCam converges in less than 20 iterations.

Parameter	Range	Nominal value
Dim	$50m \times 50m$	$50m \times 50m$
AOV	$\{45^\circ - 180^\circ\}$	90°
Target count	$10 - 100$	50
Rmax	$20m - 50m$	$30m$
CTC	$0.8 - 0.9$	0.9
$c_{\theta\to r}$	-	50

TABLE I. SIMULATION PARAMETERS

Metrics: We consider the following metrics: (1) fraction of covered targets, (2) execution time, and (3) number of Cameras-To-number of Targets (CTR) ratio.

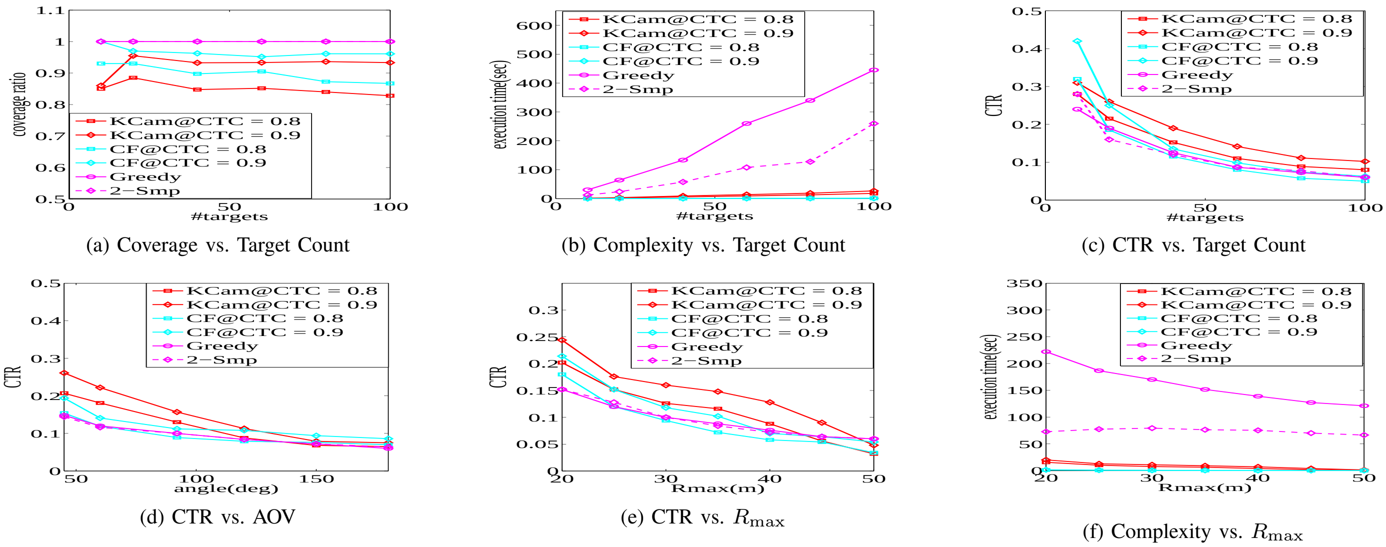
-Results

1. **The effect of target density:** The performance of KCam, CF, Greedy and 2Smp are compared in Figure 2. Figure 2a shows that the Greedy and 2Smp method achieve perfect coverage, while KCam and CF both result in lower coverage. Figure 2b shows that both KCam and CF enjoy a much lower computational complexity than Greedy and 2Smp. Specifically, the complexity of the KCam and CF algorithm is more than 10 times and 100 times less than that of 2Smp respectively. Such advantage in complexity is a result of replacing solving linear programming problems for exact solution, with clustering targets that have a higher probability to be covered by one camera, for approximate solutions. The CTR for all 4 methods is depicted in Figure 2c. This value is higher for KCam and CF, although this difference decreases as the number of targets increases.

2. **The effect of AOV and $R_{\max}$:** Some of the results for varying AOV and $R_{\max}$ are shown in Figures 2d, 2f and 2e. The overall performance of all 4 algorithms naturally shows improvement with wider AOV and longer $R_{\max}$. Most notably, as either the AOV or $R_{\max}$ increased, the CTR values for KCam and CF decreased and got closer a to those of Greedy and 2Smp and even less (better) for high $R_{\max}$ values. This is due to the additional slackness in the limitations forced by AOV or $R_{\max}$ which allows more targets in a cluster be covered by a single camera. In Figure 2f and 2e, the performance comparison between several $R_{\max}$ values has been done using two different values of CTC for KCam and CF: 0.8 and 0.9. We have omitted the figures depicting coverage for both cases of varying AOV and $R_{\max}$ due to lack of space. The trend for both resembles those in 2a. Also, the execution time graphs when increasing AOV and $R_{\max}$ were similar and hence only the latter is shown.

3. **The effect of minimum coverage criterion:** As shown in Figure 2e, the value of CTR is less for CTC of 0.8 than that of 0.9. However, this is gained at the cost of lower coverage, lower-bounded by selected CTC (the graphs omitted due to lack of space). In other words, by allowing the coverage algorithm to ignore the "outliers", we can use fewer cameras.

KCam versus CF: CF is based on clustering targets based on their location, and then finding the camera location for each cluster. KCam's clusters are initially assigned arbitrarily and are later modified dynamically based on targets coverage,



(a) Coverage vs. Target Count
(b) Complexity vs. Target Count
(c) CTR vs. Target Count
(d) CTR vs. AOV
(e) CTR vs. $R_{\max}$
(f) Complexity vs. $R_{\max}$

Fig. 2. Metrics vs. target density, Rmax and AOV

and are hence more prone to get trapped in poor quality equilibriums. Therefore, CF performs better than KCam when there are more restrictions on coverage condition, i.e. small value of AOV and/or $R_{\max}$. As these restrictions are relaxed, this performance gap diminishes.

V. Conclusion and Future Work

In this paper, we have taken the first steps towards the development of an automated, distributed and fast CPS surveillance system. We studied the problem of positioning and coordination of mobile cameras to cover a given group of targets. We have proposed two heuristic computationally efficient and centralized methods: k-camera clustering and cluster-first method. We have used simulation to evaluate our methods and found that they have much less computational complexity, but require higher number of cameras and provide lower coverage than the computationally expensive but near-optimal methods. However this gap decreased significantly as the number of targets, cameras' AOV or cameras' coverage range increased.

Our next steps are, to develop distributed versions of the proposed algorithms, and address the communication and path-planning issues that will arise in such settings. We also plan to test them in a testbed using multiple camera-equipped MAVs.

References

[1] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner, "Networked uavs as aerial sensor network for disaster management applications," *Elektrotechnik und Informationstechnik*, vol. 127, no. 3, pp. 56–63, Mar 2010.

[2] 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." ACM SenSys, Nov 2007, pp. 117–129.

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

[4] I. Akyildiz, T. Melodia, and K. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, p. 921960, Dec 2007.

[5] H. Alemdar, Y. Durmus, and C. Ersoy, "Wireless healthcare monitoring with RFID-enhanced video sensor networks," *Distributed Sensor Networks*, vol. 10, no. 1, pp. 290–300.

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

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

[8] J. Urrutia, "Art gallery and illumination problems," *Handbook of Computational Geometry*, vol. 1, no. 1, pp. 973–1027, 2000.

[9] C. S. S. Eidenbenz and P.Widmayer, "Inapproximability results for guarding polygons and terrains," *Algorithmica*, vol. 31, no. 1, pp. 181–192, Nov 2001.

[10] O. Cheong, A. Efrat, and S. Har-Peled, "Finding a guard that sees most and a shop that sells most," *Discrete Comput. Geom.*, vol. 37, no. 4, pp. 545–563, June 2007.

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

[12] J. Ai and A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *Combin. Optim.*, vol. 11, no. 1, pp. 21–41, June 2006.

[13] V. P. Munishwar and N. B. Abu-Ghazaleh, "Coverage algorithms for visual sensor networks," *ACM Trans on Sensor Networks*, vol. 9, no. 4, pp. 1–34, July 2013.

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

[15] C. Soto, B. Song, and R. Chowdhury, "Distributed multi-target tracking in a self-configuring camera network." IEEE CVPR, Nov 2009, p. 14861493.

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

[17] 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.

[18] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, "Senseye: a multi-tier camera sensor network," no. 1. ACM Multimedia, November 2005, pp. 71–77.

[19] X. Yu, "Hybrid radio frequency and video framework for identinity-aware augmented perception in disaster management and assistive technologies," Ph.D. dissertation, UMass Amherst, MA, 2013.

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