2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems

Low Complexity Target Coverage Heuristics Using Mobile Cameras

Azin Neishaboori[†], Ahmed Saeed[†], Khaled A. Harras[*], Amr Mohamed[†]
[*]School of Computer Science, Carnegie Mellon University and [†]Department of CSE, Qatar University
{azin.neishaboori, cse.saeed}@gmail.com, kharras@cs.cmu.edu, amrm@ieee.org

***Abstract*—Wireless sensor and actuator networks have been extensively deployed for enhancing industrial control processes and supply-chains, and many forms of surveillance and environmental monitoring. The availability of low-cost mobile robots equipped with a variety of sensors in addition to communication and computational capabilities makes them particularly promising in target coverage tasks for ad hoc surveillance, where quick, low-cost or non-lasting visual sensing solutions are required, e.g. in border protection and disaster recovery. In this paper, we consider the problem of low complexity placement and orientation of mobile cameras to cover arbitrary targets. We tackle this problem by clustering proximal targets, while calculating/estimating the camera location/direction for each cluster separately through our *cover-set coverage* method. Our proposed solutions provide extremely computationally efficient heuristics with only a small increase in number of cameras used, and a small decrease in number of covered targets.**

I. Introduction

Recent advancements in manufacturing low-cost wireless battery operated cameras have made them increasingly more feasible and affordable in a variety of applications such as smart surveillance, environment monitoring, traffic management, and health care [1]. Mobile cameras can be additionally used in Visual Sensor Networks (VSNs) for ad-hoc surveillance where a set of wireless cameras are to survey areas with little to no available infrastructure, or when rapid and non-lasting deployment is necessary [2]. Recently, Micro Air Vehicles (MAVs) (a.k.a. microdrones or small/micro UAVs), typically equipped with cameras, were proposed to be used as mobile cameras [3]. One of the main advantages of using MAVs is their maneuverability and small size enabling them to be placed in locations that achieve optimal sensing coverage in both indoor [4] and outdoor scenarios [3].

Smart surveillance with the assistance of mobile cameras requires tackling multiple challenges including *targets* or *area* coverage, tracking, activity detection and others. In this paper, we address 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 in both isotropic [5] and anisotropic sensors [6]. Therefore, it has been simplified in many forms in the field of robotics and sensor networks [7]. Various studies have addressed area coverage [6], [8], [9] and target coverage [2], [5], [6], often making simplifications include fixing camera locations, discretizing space and/or camera pan. Approximation algorithms for the case of anisotropic sensors have also been proposed [5], [6]. Despite these efforts, finding a near-optimal computationally efficient algorithm in arbitrarily large areas and/or for an arbitrary number of targets has remained a challenge.

Motivated by the need for computationally efficient algorithms for autonomous control of the mobile visual sensors, we propose efficient near-optimal algorithms for finding the minimum number of cameras to cover a high ratio of a set of targets. First, we develop a basic method, called cover-set coverage to find the location/direction of a single camera for a group of targets. This method is based on finding candidate points for each possible camera direction and spanning the direction space via discretizing camera pans. We then propose two algorithms which divide targets into multiple clusters and use the cover-set coverage method to find the camera location/direction for each cluster: (1) Smart Start K-Camera Clustering (SSKCAM): starting from a given set of clusters identified by an off-the-shelf clustering algorithm, we iteratively adjust clusters based on the coverage status of their comprising targets, and recalculate the camera location/direction for each individual cluster until convergence is achieved; (2) Fuzzy Coverage (FC) algorithm: we cluster targets allowing overlapping (fuzzy) clusters and then find the camera location/direction for each cluster. We evaluate our proposed algorithms via a simulation study in MATLAB and observe that our algorithms offer lower computational complexity compared to earlier work. This gap increases as the coverage range or number of targets increase reaching more than $50\times$ faster performance using similar number of cameras to cover at least a pre-determined fraction of targets.

The rest of this paper is organized as follows. In Section II, we state our assumptions and pose our problem. In Section III, we present the details of our proposed algorithms. We evaluate these algorithms in Section IV and conclude the paper in Section V.

II. Assumptions and Problem Statement

A. Assumptions

Targets: We assume that targets reside on a 2D plane, as typically done in target coverage problems [2]. We also assume that targets are represented as points and that their location are known [2]. This information may be obtained by a higher tier low-granularity camera, used only for detection and localization, and conveyed to lower tier cameras [10], or via using RFIDs [11].

Cameras and Camera Coverage: We assume having horizontal cameras with coverage areas shaped as circular sectors, highlighted in orange in Figure 1. The maximum depth of view of a camera is the radius of this sector, $R_{\max}$. The Angle of View (AOV) of the camera is the angular width of this sector, and is approximately inversely proportional to the Lens's focal length. For target T_j to be covered by camera C_i, (1) T_j should be within $R_{\max}$ distance of the camera, and (2) the angle between C_i's direction and T_j should be within the AOV of the camera. We impose hard constraints on coverage of a target: either completely covered or not covered at all. Occlusions are not considered herein as it is typical in many target coverage studies. We assume mobile cameras capable of moving to a certain position with a certain orientation when commanded to. A minimum acceptable threshold is posed

978-1-4799-6036-1/14 $31.00 © 2014 IEEE
DOI 10.1109/MASS.2014.70

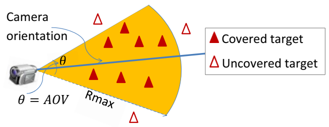


Fig. 1. Coverage area of one camera

on the fraction of covered targets, referred to by Coverage Termination Criterion (CTC).

Camera Configuration System and Medium: A centralized computational entity calculates and informs the location/orientation of all mobile cameras. This may be a separate entity or one of the mobile cameras. A wireless communication channel exists between the mobile cameras and the central computational entity. To avoid the necessity of multihop communication, coverage areas are assumed to be smaller than the wireless communication range.

B. Problem Statement

Target Coverage Problem: *Given a set of 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 position and orientation such that all targets are visible by at least one camera.*

The above problem is proved NP-Complete [5]. We simplify it by changing the objective to covering at least a predetermined percentage of targets with the minimum number of cameras whose locations and directions are to be found. Note that this percentage may be made arbitrarily small.

III. Methodology and Proposed Solutions

In this section we first describe our proposed method, *Cover-Set Coverage*, to find the location/direction for a single camera that is to cover one *cover-set* in III-A. We then propose two algorithms, *Smart Start K-Camera Clustering* and *Fuzzy Coverage Algorithm*, to divide a group of targets with arbitrary locations to a small number of cover-sets.

A. Cover-Set Coverage Method

Definition 1: **A cover-set**: A group of targets, $T_i, i \in \{1..N\}$ that can be covered by **one** camera under certain camera specifications, i.e. given $R_{\max}$ and AOV.

Cover-Set Problem: Given a set of targets $S = \{T_1..T_K\}$, determine whether this set is a cover-set, and if so, find the location and direction of the single camera, with $AOV = \theta$ and maximum coverage range of $R_{\max}$ that covers all targets.

To solve this problem we first form $H(S)$, convex hull of S. $H(S)$ is a polygon in 2D, see Figure 2. To cover all targets in set S using one camera with $\theta < 2\pi$, the camera has to reside outside $H(S)$. This can be easily seen by noting that the sum of angles from a point inside a convex polygon to all its vertices is 2π. Therefore, unless $\theta = 2\pi$, it is not possible to cover all targets in S with a point inside the convex polygon.

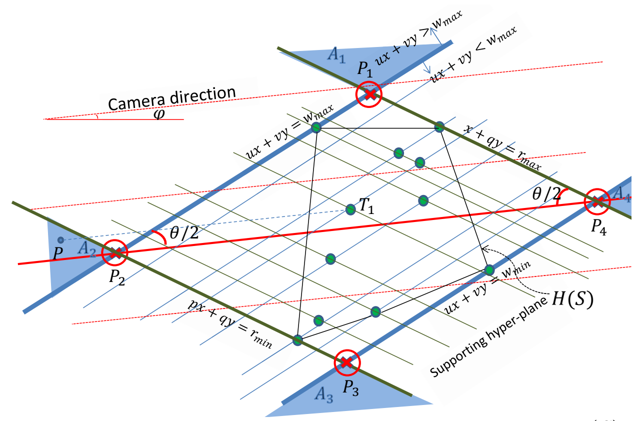


Fig. 2. Cover-Set Coverage Method: Targets are the green dots. $H(S)$ is the polygon with black boundaries. For a given camera direction ϕ (the red line), maximum of 4 points are selected as camera location candidates. Two lines which make $\theta/2$ (green lines) and $-\theta/2$ (blue lines) with the camera direction are passed through each target. These "outer-most" lines (bolded) intersect at four points P_1 to P_4. Each point is associated with an area filled in blue, e.g. point P_1 with area A_1

1) Base Case: Continuous Camera Direction Angle Space: To find regions that meet the AOV requirement for all targets at a **specific** given camera direction, the following steps will be followed, also illustrated in Figure 2.

(i) Set camera direction (denoted by the red lines in the figure) at angle ϕ relative to the x-axis. (ii) Find u and v values such that $-\frac{u}{v} = \tan(|\phi - \theta/2|)$ (many pairs of such values may be found and any of them would work for our purpose). (iii) For each target point, pass a line with the same slope of $-\frac{u}{v}$. For each target $T_i = (x_i, y_i)$, such a line will follow equation $ux_i + vy_i = w_i$. (iv) For all targets $T_i \in S$, find $w^*_{\max} = \max w_i, \forall i$. (v) Since $H(S)$ is a convex polygon, the line corresponding to $c^*_{\max}$, i.e. $ux + vy = w^*_{\max}$ forms a supporting hyper-plane to it. (vi) Similarly, find the supporting hyperplane to this polygon from underneath by finding $w^*_{\min} = \min\{w_i, \forall i\}$. The corresponding line $ux + vy = w^*_{\min}$ also forms a supporting hyper-plane to this polygon. Note that the above set of steps make an $O(N)$ operation.

For the camera to be able to cover all targets, its location must be in the region either on or above the max hyperplane, $ux + vy > w^*_{\max}$, or in the region on or below the min hyperplane, $ux + vy < w^*_{\min}$. Otherwise, one of targets will be viewed by an angle larger than $\theta/2$ relative to the selected camera direction. Using the exact procedure above, find lines $px + qy = r^*_{\min}$ and $px + qy = r^*_{\max}$ where $-\frac{p}{q} = \tan(\phi + \theta/2)$. Again, the solution has to be on or above the max hyperplane, $px + qy > r^*_{\max}$ or on or below the min hyperplane $px + qy < r^*_{\min}$.

Next, to find the camera location candidates, the four intersection points will be found from the above procedure:

(i) P_1 is the intersection of $ux + vy = w^*_{\max}$ and $px + qy = r^*_{\max}$, (ii) P_2 is the intersection of lines $ux + vy = w^*_{\max}$ and $ux + vy = w^*_{\min}$, (iii) P_3 is the intersection of $ux + vy = w^*_{\min}$ and $px + qy = r^*_{\min}$, (iv) P_4 is the intersection of lines $px + qy = r^*_{\max}$ and $px + qy = r^*_{\min}$.

Based on the value of θ (AOV), following cases will apply:

Case $\theta < \pi/2$ **:** There are two feasible regions for placing the camera such that the AOV requirements are met for all targets. These regions are convex cones centered at points P_2 and P_4 in the figure and are highlighted in blue. We call them A_2 and A_4. It can be shown that if neither of points P_2 and P_4 meet the coverage range requirements of all targets in S, then no other location in the two feasible regions A_2 and A_4 can. This is shown in the figure for area A_2 and target T_1.

Case $\theta \geq \pi/2$ **:** All four points P_1, P_2, P_3 and P_4 (and their associated areas A_1 to A_4) meet the AOV requirements. However, unlike in the previous case, only for points P_1 and P_3 we can ensure that if they do not meet the coverage range requirement, then no point in A_1 and A_3 can. To find the feasible solution regions in areas A_2 and A_4, circles centered at all targets $T_i \in S$, of radius $R_{\max}$, are to be overlapped. If there is an overlapping region, then all points in that region are solution points. This is an $O(N^3)$ operation.

Case $\theta = \pi/2$ **:** All four points P_1 to P_4 meet the AOV requirements, and are hence candidate points. Also, for all these points we can show that if their distance to a target does not meet the coverage range criterion, then no point in their associated region (A_1 to A_4) does.

2) Cover-Set Coverage By Discretizing Camera Direction Angle Space: The above method may lead to maximum of 4 points for each possible value of ϕ. To reduce our search space, we discretize ϕ by quantizing it to $2\pi/\Delta\phi$ values evenly distributed in the interval of $[0, 2\pi)$. For $AOV \leq \pi/2$, there will be 2 or 4 candidate points for $AOV < \pi/2$ and $AOV = \pi/2$ respectively. For $AOV > \pi/2$, finding candidate points requires overlapping areas A_2 and A_4 with circles of radius $R_{\max}$ formed around each targets (an $O(N^3)$ operation). To maintain complexity low, we instead use points P_2 and P_4 , see Figure 2. Therefore, the complexity of finding one camera location/direction for a set of targets S (which we hope form a cover-set) with size $||S||$ is $O(||S||2\pi/\Delta\phi)$.

B. Smart-Start K-camera Clustering

This algorithm is akin to k-means clustering. However, unlike k-means clustering where the criterion for clustering data points together is only the distance between them (in the selected features), here the goal is for targets that can be covered by one camera (meeting both range and AOV requirements) to be clustered together. In Smart-Start KCamera clustering (SSKCAM), we initially cluster targets using k-means clustering. Afterwards, targets are re-assigned to adjacent clusters which can cover them, and find new location/direction for camera of each cluster using the method in III-A, until no further move is possible. SSKCAM is an improved version of our previously proposed algorithm KCAM in [12]. We refer the readers to that paper for the pseudo-code of KCAM which may be modified to reflect SSKCAM.

The computational complexity of SSKCAM depends on that of the clustering algorithm, and the cover-set coverage method used to find camera location/direction for each cluster. For k-means clustering, with binary updating in k, the complexity for clustering is $O(log(K)NM)$ where N is the number of targets, K is the final number of cameras, and M is the number of iterations to reach an equilibrium for each given k. Therefore, the overall complexity is the sum of the complexity of the two subtasks, i.e. $O(log(K)NM) + O(\frac{2\pi}{\Delta}\phi \times N)$.

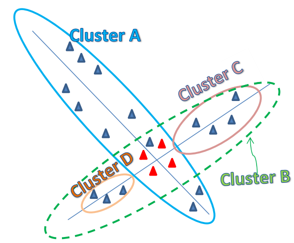


Fig. 3. Overlapping clusters allow less clusters

C. Fuzzy Clustering

SSKCAM allows a target to belong to only one cluster. This is a short-coming in the example shown in Figure 3 and explained below. In this figure, with overlapping clusters, the red targets would be clustered with both cluster A and B, resulting in a total of two clusters. With no overlap however, at least 3 clusters are required to cover all targets, e.g. A, C and D. We therefore consider using **Fuzzy or overlapping classification** where a target can belong to multiple clusters each with a different probability. For each identified cluster, we then apply the cover-set coverage method described in III-A to find its camera location and orientation.

The computational complexity of this algorithm depends on that of the fuzzy clustering algorithm used, and that of cover-set coverage method applied for finding camera location/direction for each cluster. Using fuzzy k-means clustering, the overall complexity will remain linear in both dimensions and number of targets [13].

IV. Evaluation

A. Simulation Set-Up

MATLAB simulation was used to compare the performance of the proposed algorithms against those of others previously proposed in the literature. The location of targets were randomly generated using a *uniform distribution* over a given square-shaped area. All results are averages of ten randomly generated scenarios.

Two near-optimal heuristic algorithms amongst those proposed in [9] were selected to compare against ours, and were modified to cover specific targets instead of a whole area: (i) greedy search, is the closest to optimal in coverage, but is the most computationally demanding algorithm, and (ii) dual-sampling, which is the most computationally efficient algorithm proposed therein. To the best of our knowledge, no other heuristic algorithms could be used which made similar assumptions and would allow us to compare our algorithms against. Note that it is difficult to make a meaningful comparison between our algorithms and those which make additional assumptions on location of cameras.

In greedy search, sensors are placed one at a time. The position and orientation of each additional sensor is decided considering the rank of all possible location-orientation pairs. Each position/orientation pair is ranked by how many remaining targets it can cover. Therefore, this algorithm requires $O(D^2 \frac{2\pi}{\Delta\phi} N^2 \log N)$ computations and $O(D^2 \frac{2\pi}{\Delta\phi})$ stored elements, where D is the dimension of the area, N is the number of targets, and $\Delta\phi$ is the pan step. In dual-sampling (D-Smp), one target is randomly selected each time. The location of the camera is chosen from a limited area in the $R_{\max}$

Parameter	Range	Nominal value
Dim	$50m \times 50m$	$50m \times 50m$
AOV	$\in \{45^\circ - 150^\circ\}$	90°
Target count	$20 - 200$	50
$R_{\max}$	$5m - 30m$	$15m$
$\Delta\phi$	-	$\pi/6$
CTC	-	0.9

TABLE I. SIMULATION PARAMETERS: RANGE AND NOMINAL VALUES

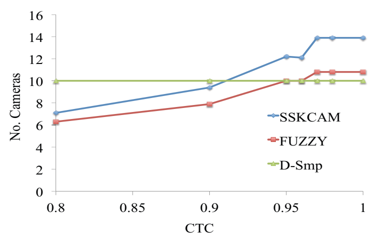


Fig. 4. The impact of CTC on required number of cameras

vicinity of this target. Then the location-orientation pair with the highest rank that can cover the selected target is selected. Therefore, it requires $O(N^2 \log N R_{\max}^2 \times \frac{2\pi}{\Delta\phi})$ computations and $O(R_{\max}^2 \frac{2\pi}{\Delta\phi})$ stored elements.

Parameters: The values of the parameter are summarized in Table I. We set the maximum allowed number of cameras to $2/3$ of targets. Coverage Termination Criterion (CTC) indicates the minimum acceptable covered target fraction.

Metrics: (1) number of cameras, (2) execution time, and (3) fraction of uncovered targets.

B. Performance Results

The effect of minimum acceptable coverage criterion: It is possible to set CTC arbitrarily high or even set it to one. However, this may add to the number of cameras required for the coverage task. Depending on the specific application and its allocated budget, we may be willing to compromise coverage to a small extent to save in the number of required cameras. In Figure 4, we depict the impact of CTC choice on number of cameras. We display the number of cameras obtained by dual-sampling (with 100% coverage) as a base for comparison. Note that by allowing 10% uncovered targets, we can save about 40% in the number of cameras. For this reason, for the rest of our simulations, we set the nominal value of CTC at 0.9 (i.e. 90% coverage).

The effect of target density: The performance of SSKCam, FC, Greedy and D-Smp are compared in Figures 5. Figure 5(a) shows that the Greedy and 2Smp algorithm achieve perfect coverage (in the tested scenarios), while SSKCam and FC leave a fraction of targets uncovered. This is because we allow our solutions to find imperfect coverage at the cost of lower computational complexity. SSKCAM shows a better coverage performance than FC. In Figure 5(b), we eliminated the results for greedy search because its values were so large that the scale of the figure would shadow the performance difference among other algorithms. As can be seen, FC has a very low execution time. Also SSKCAM does better than D-Smp, and exhibits better scalability as number of targets is increased. For FC, the advantage in computational complexity is a result of the one time operation of clustering (instead of iterating) and camera configuration computations in section III-A. However, this leaves more targets uncovered in comparison to SSKCAM. As can be seen in Figure 5(c), permitting compromise in fraction of covered targets also yields in camera numbers obtained to be very similar to those from greedy and D-Smp, for SSKCAM, and even lower (better) for FC. Note that the latter advantage for FC comes at the price of lower coverage fraction. Also note that for these figures and the ones that will follow for other parameters, the number of cameras and uncovered target fraction should be interpreted together. This is because of the CTC ratio fixed at 0.1, and that in SSKCAM, camera number incrementing is stopped once minimum coverage criterion is reached.

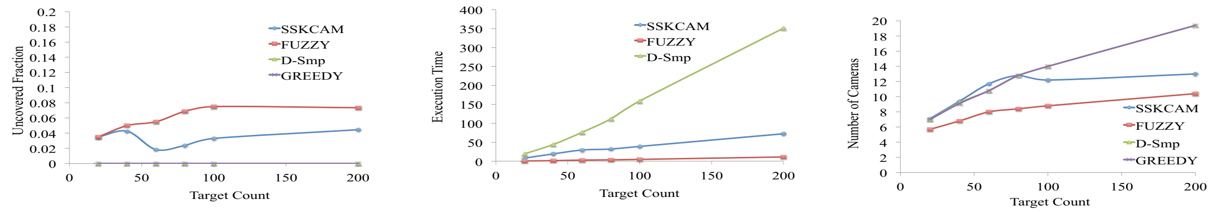
The effect of AOV: We compare the performance of the five algorithms for different values for AOV: 45° (AOV of an unzoomed webcam), 60°, 90°, 120 and 150 for fish-eyed lenses. The results are shown in Figures 6. It can be noticed from these figures that, as expected, the number of required cameras for all 5 algorithms improves (and almost converges) with wider AOV. The performance in terms of coverage and execution time across the different algorithms follows the same pattern as in Figure 5(c): perfect coverage for greedy and 2Smp, and lowest for FC. Execution time decreases as AOV increases for all algorithms; FC has a very low (good) complexity, will Greedy and 2Smp have the highest. Finally, the execution time of SSKCAM and FC has a bump at $AOV = \pi/2$ due to the existence of 4 candidates instead of 2 at every given camera direction ϕ (see section III-A).

Remark: Some range of values for $R_{\max}$ make the target coverage problem trivial. Consider the **covering-density** of K cameras with range $R_{\max}$ and $AOV = \theta$ defined in [6] as $\frac{K \times \pi R_{\max}^2 (\frac{\theta}{2\pi})}{D^2}$ where the nominator measures the area covered by all cameras and the denominator measures the total area surface. Since this calculation does not include area overlap, it is a conservative metric. Nonetheless when this value measures to one, we have covered all the given area, and therefore, the target coverage problem becomes trivial.

The effect of $R_{\max}$: We varied the value of $R_{\max}$ between 5m and 30m. Note that for one specific camera type this value reflects the image quality required by the application. The results are depicted in Figure 7. Again as expected, the number of required cameras decreased for all 5 algorithms as $R_{\max}$ increases. This quantity was also quite similar across all the algorithms. To compare the execution times of these algorithms, we again eliminated the results for greedy since they were very high and would skew the scaling. As can be seen in the figure, while FC and SSKCAM's execution time decreases as $R_{\max}$ increases, D-Smp exhibits an exponential increase in execution time.

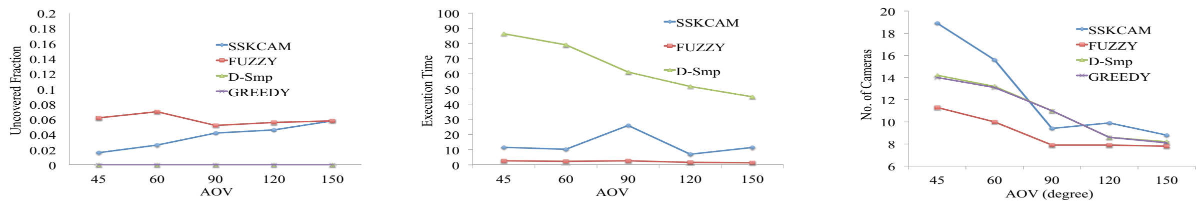
V. CONCLUSION AND FUTURE WORK

In this paper, we studied the problem of positioning and orienting mobile cameras to cover a given group of targets. We first develop cover-set coverage method, which finds the location/direction of a single camera such that the number of targets covered is near-maximal. We then proposed two heuristic computationally efficient and centralized algorithms: smart start k-camera clustering and fuzzy clustering algorithm, both of which divide targets to multiple clusters and apply cover-set coverage method on each cluster. We used simulation to evaluate our methods and found that they have less computational complexity, but provide lower coverage



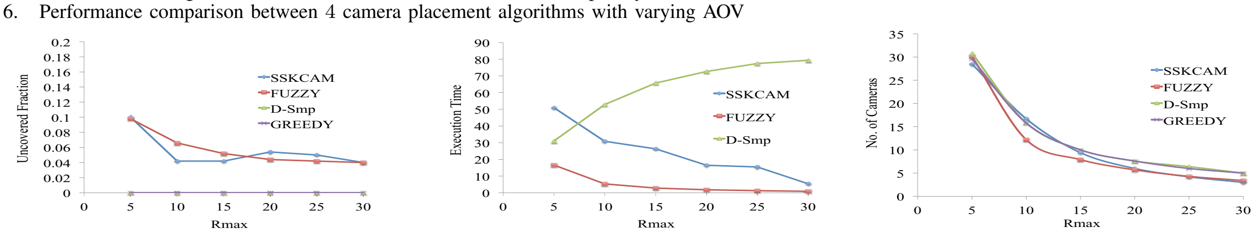
(a) Uncovered Target fraction vs. Target Count (b) Complexity vs. Target Count (c) Camera Count vs. Target Count

Fig. 5. Performance comparison between 4 camera placement/orientation algorithms vs target count



(a) Uncovered Target Fraction vs. AOV (b) Complexity vs. AOV (c) Camera Count vs. AOV

Fig. 6. Performance comparison between 4 camera placement algorithms with varying AOV



(a) Uncovered Target Fraction vs. $R_{\max}$ (b) Complexity vs. $R_{\max}$ (c) Camera Count vs. $R_{\max}$

Fig. 7. Performance comparison between 4 camera placement algorithms with varying $R_{\max}$

than the computationally expensive but near-optimal methods. Our next steps are, to develop distributed versions of the proposed algorithms, cover mobile targets, and address ad-hoc communication and path-planning while considering energy consumption issues that will arise in such settings.

VI. 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.

References

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

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[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] 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.

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

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

[11] 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.

[12] A. Neishaboori, A. Saeed, A. Mohamed, and K. Harras, "Target coverage heuristics using mobile cameras," *Intl workshop on Robotic Sensor Networks, CyPhy week,Berlin, April 2014*, 2014.

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