Multi-Target Visual Tracking With Aerial Robots

Pratap Tokekar, Volkan Isler and Antonio Franchi

***Abstract*— We study the problem of tracking mobile targets using a team of aerial robots. Each robot carries a camera to detect targets moving on the ground. The overall goal is to plan for the trajectories of the robots in order to track the most number of targets, and accurately estimate the target locations using the images. The two objectives can conflict since a robot may fly to a higher altitude and potentially cover a larger number of targets at the expense of accuracy.**

We start by showing that $k \geq 3$ robots may not be able to track all n targets while maintaining a constant factor approximation of the optimal quality of tracking at all times. Next, we study the problem of choosing robot trajectories to maximize either the number of targets tracked or the quality of tracking. We formulate this problem as the weighted version of a combinatorial optimization problem known as the Maximum Group Coverage (MGC) problem. A greedy algorithm yields a $1/2$ approximation for the weighted MGC problem. Finally, we evaluate the algorithm and the sensing model through simulations and preliminary experiments.

I. Introduction

We study the problem of tracking multiple moving targets using aerial robots. We consider the scenario where cameras that face downwards are mounted on the robots to track targets moving on the ground plane. A robot can potentially track more targets by flying to a higher altitude, thus increasing its camera footprint. However, this may reduce the quality of the view due to the increased distance between the cameras and the targets. There is a trade-off between the number of targets tracked and the corresponding quality of tracking. We investigate this trade-off and present an approximation algorithm for multi-target tracking.

We start by showing that it may not always be possible to track all targets while always maintaining the optimal quality of tracking (or any factor of the optimal quality), even if the targets' motion is fully known. Hence, we focus on the following two variants: maximize the number of targets tracked subject to a desired tracking quality per target, and maximize the sum of quality of tracking for all targets. The two problems can be formulated as the unweighted and weighted versions of the Maximum Group Coverage Problem (MGC). A simple greedy approach provides a 1/2 approximation to unweighted MGC [1]. We show that the approximation guarantee also holds for the weighted case which allows a practical solution to the trajectory planning problem with provable performance guarantees. We evaluate the algorithm in simulations and preliminary experiments with an indoor platform using four aerial robots.

The rest of the paper is organized as follows. We begin with the related work in Section II. The problem setup and a discussion of the sensing quality are presented in Section III. The infeasibility of tracking all targets with a constant factor of the optimal quality is proven in Section IV. The tracking algorithm is presented in Section V, and evaluated through simulations and preliminary experiments in Sections VI and VII respectively. Section VIII concludes the paper.

P. Tokekar and V. Isler are with the Department of Computer Science & Engineering, University of Minnesota, U.S.A. {tokekar,isler} at cs.umn.edu

A. Franchi is with the Centre National de la Recherche Scientifique (CNRS), Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), France. antonio.franchi at laas.fr

This material is based upon work supported by the National Science Foundation under Grant Nos. 1317788 and 1111638.

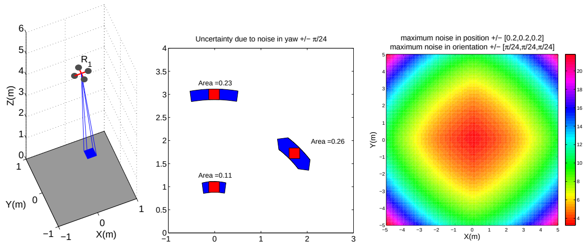
II. Related Work

Target tracking is an important problem for robotics, and has been widely studied under different settings. Spletzer and Taylor [2] considered the problem of tracking multiple mobile targets with multiple robots. They presented a general solution based on particle filtering in order to choose robot locations for the next time step that maximizes the quality of tracking. Frew [3] studied the problem of designing a robot trajectory, and not just the next robot location, in order to maximize the quality of tracking a single moving target. LaValle et al. [4] studied the problem of maintaining the visibility of a single target from a robot for the maximum time. Gans et al. [5] presented a controller that can keep up to three targets in one robot's field-of-view.

When the motion of the targets is fully known, the tracking problem can be formulated as a kinetic facility location problem. The goal of the stationary version is to place k facilities (robots) given the location of n sites (targets), so as to minimize the maximum distance between a facility and a site. For the kinetic version, Bespamyatnikh et al. [6] and Durocher [7] presented approximation algorithms to control respectively one and two mobile facilities, when the trajectories for the sites are given. Recently, de Berg et al. [8] presented improved approximation algorithms with two mobile facilities when only an upper bound on the velocities of the sites is available. However, the general problem of kinetic facility location with k facilities is open.

In the extreme case where no prior information of the targets is available, the multi-robot tracking problem can be formulated as a coverage problem [9]. Schwager et al. [10] presented strategies to control the position and orientation of overhead cameras mounted on aerial robots in order to achieve equal visual coverage of the ground plane.

Unlike previous works, we study the trade-off between quality of tracking, and the number of targets tracked. We present an algorithm that chooses *trajectories* for each robot, instead of choosing just the next best location. This algorithm



(a) (b) (c)

Fig. 1. (a) Backprojection from a pixel yields a pyramid. (b) Uncertainty in target's estimate due to uncertain yaw angle of the robot. (c) Map showing the area of projection for the true target at $[x, y, 0]$ (best viewed in color). The camera pose is estimated to have position $[0, 0, 5]$ m and roll, pitch and yaw angles as 0 radians. Maximum image noise is ± 5 pixels.

can be applied to the following two versions of the problem: tracking maximum number of targets, and maximizing the quality of tracking. We begin by formulating the problem and describing the sensing model.

III. Preliminaries and Problem Formulation

Let k denote the number of robots, and n denote the total number of targets in the environment. The position of any robot or target is specified by their 3D coordinates x, y, z. The position of the i^{th} robot at time τ is denoted by $r_i(\tau)$. Let $z_{\min}$ be the minimum flying altitude. All robots have a camera that faces downwards. Let ϕ represent the field-of-view angle for the cameras. We assume that the robots can communicate amongst each other at all times.

Let $t_i(\tau)$ denote the position of the i^{th} target. $t_i(\tau)$ is given by the position of a reference point that the robots can use to uniquely identify any target. For example, the reference point can be the centroid of a colored patch or a unique feature point on the object. All targets always move on the ground plane, i.e., $z = 0$ for all t_i.

The reference point of any target t_i in the field-of-view of a robot projects to some pixel in the image. A pixel can be backprojected to a ray in the world frame. In general, with no other information, it is not possible to solve for the target's location along this ray with a single camera measurement. However, since we assume that all targets move on the ground plane, we can solve for the coordinates of t_i.

Ideally, we can exactly estimate t_i given an image measurement, the camera pose, and the projection matrix. In practice, however, the following factors lead to an uncertain estimate of t_i:
(1) The backprojection of camera pixels, which have quantized, integer coordinates, is not longer single ray but a pyramid (Figure 1(a)).
(2) Pixel measurements may be corrupted by noise. If the maximum noise is bounded by Δp pixels, we backproject the set of pixels $\pm\Delta p$ around the measured pixel. The true target location is contained within the larger backprojection.
(3) The pose of the camera (or the robot) may not be accurately known. Typically, using exteroceptive sensors such as GPS and compass, we can bound the maximum uncertainty in estimating the robot pose. When the robot pose is known up to a bounded uncertain set, we can compute the backprojection for each pose within the set (Figure 1(b)).

In general, the quality of tracking under the three sources of errors, is a function of the relative distance and angle between the robot and the target, as seen in Figure 1(c). For a given true location of the target and an estimate of the robot pose, Figure 1(c) plots the maximum area of backprojection over all possible noisy measurements of the target, and all possible true robot poses.

While tracking, robots only have an estimate of the true target position. The uncertain estimate can be represented as a set of possible target locations on the ground plane. Given a motion model, the robots can propagate the set to obtain predicted target position, e.g., using particle filtering [11]. The maximum area of backprojection can be computed for each predicted target position as shown in Figure 1(c).

The quality of tracking for a given target and robot pair can be defined as some measure of the areas of backprojection found for a predicted target position. Let $q_i(r_j, \tau)$ denote the measure for target t_i and robot r_j at time τ. The quality of tracking t_i at τ, is given by the best quality of tracking amongst all robots tracking t_i, i.e., $q_i(\tau) = \max_j q_i(r_j, \tau)$. Finally, the total quality of tracking at τ is given by the sum of quality over all targets $Q(\tau) = \sum_{\forall i} q_i(\tau)$ over all targets. Alternatively, we may also consider the bottleneck quality over all targets $Q(\tau) = \min_i q_i(\tau)$.

IV. Infeasibility of Tracking All Targets

In this section, we show the infeasibility of tracking all targets while maintaining any constant factor approximation of the optimal quality of tracking. We prove this by constructing an instance where the two goals, track all targets and maximize quality of tracking, conflict each other. We create a simple instance on a line where the quality of tracking is inversely proportional to the distance between the robot and the target: $q_i(r_j, \tau) = 1/d(t_i(\tau), r_j(\tau))$ if t_i is in the field-of-view of r_j, and $q_i(r_j, \tau) = 0$ otherwise. The overall quality of tracking will be given by the bottleneck quality $Q(\tau) = \min_i q_i(\tau)$.

We use the instantaneous optimal quality of tracking, $Q^*(\tau)$, as the baseline for comparison. $Q^*(\tau)$ is the quality of tracking at τ, if one were to optimally *place* all the cameras at any location for any τ, regardless of their locations before τ. The placement of k cameras achieving $Q^*(\tau)$ may be significantly different from the placement achieving $Q^*(\tau - \epsilon)$. There may or may not exist k continuous robot trajectories achieving $Q^*(\tau)$. Nevertheless, $Q^*(\tau)$ is an upper bound on the quality of tracking. This raises the question of whether we can at least maintain a constant-factor approximation of $Q^*(\tau)$ while tracking all targets. The theorem given next shows this is not possible, even when the motion of the targets is fully known.

Theorem 1 *Let $Q^*(\tau)$ be the instantaneous optimal quality of tracking at time τ. Let the maximum speed of all targets be v. For any $0 < \alpha \leq 1$ and $\beta > 0$, no algorithm can track all $n > k$ targets with at least $\alpha Q^*(\tau)$ quality for all τ with $k \geq 3$ robots having a maximum speed of βv.*

Proof: Consider Figure 2. We have $k = 3$ robots and $n = 4$ targets on a line. The distance between t_3 and t_4 is 0 at time 0. Targets t_1, t_2 and t_3 remain stationary at all times, and t_4 moves with $v = 1$ to the right on the line. $z_{\min} = 1$ and $\phi = \pi/4$ denote the minimum flying altitude and field-of-view angles (Section III).

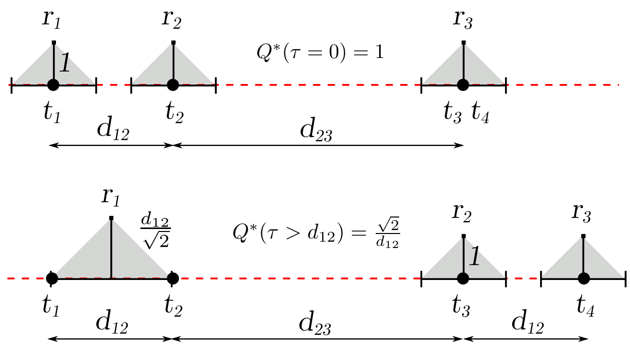


Fig. 2. At $\tau = 0$, t_3 and t_4 are covered by the same robot to achieve $Q^*(0)$, where as for $\tau > d_{12}$, t_3 and t_4 are covered by separate robots.

If we have 4 targets and 3 robots, then there must exist a robot covering at least two targets at any given time. At $\tau = 0$, we can verify that the optimal algorithm uses separate robots to cover t_1 and t_2, and one robot to cover t_3 and t_4 (Figure 2). That is, $Q^*(0) = 1$. Similarly, for any time $\tau > d_{12}$, optimal uses separate robots to cover t_3 and t_4, and same the robot to cover t_1 and t_2 making $Q^*(\tau) = \frac{\sqrt{2}}{d_{12}}$.

Thus, in any optimal algorithm, of the two robots covering t_1 and t_2, one will switch to cover either t_3 or t_4, after $\tau = d_{12}$. *An approximation algorithm, on the other hand, does not necessarily have to make the same switch.* Nevertheless, by setting d_{12} appropriately, we will show that any approximation algorithm will be required to make the same switch at some time. By making d_{23} sufficiently large, we will show that such a switch is infeasible with bounded velocity robots. The rest of the proof shows the existence of appropriate d_{12} and d_{23} values. This construction is similar to the one used by Durocher [7] to prove the inapproximability of the kinetic k–center problem. For the case of aerial robots, however we show how to additionally take into account non-zero $z_{\min}$ and ϕ values.

Let ALG be any algorithm that maintains a quality $Q(\tau) \geq \alpha Q^*(\tau)$. If we set $d_{12} > \frac{\sqrt{2}}{\alpha}$, then ALG cannot use the same robot to cover t_1 and t_2 at time $\tau = 0$. Else, $Q(0) < \alpha = \alpha Q^*(0)$ which violates the approximation guarantee. Hence, ALG uses separate robots to cover t_1 and t_2 at time 0.

Similarly, we can show that for any time $\tau > \frac{d_{12}}{\alpha}$, ALG must use separate robots to cover t_3 and t_4. Else $Q(\tau) < \frac{\sqrt{2}}{\tau} < \alpha Q^*(\tau)$ violating the approximation guarantee.

One of the two separate robots, say r, covering t_1 and t_2 initially, must cover either t_3 and t_4 at time $\tau > \frac{d_{12}}{\alpha}$. In time τ, r must travel at least $d_{23} - \frac{1}{\alpha} - \frac{d_{12}}{\sqrt{2}\alpha}$ distance. Here, $\frac{1}{\alpha}$ and $\frac{d_{12}}{\sqrt{2}\alpha}$ come from the condition that $Q(0) \geq \alpha$ and $Q(\tau) \geq \alpha \frac{\sqrt{2}}{d_{12}}$.

Consider a time $\tau = \frac{2d_{12}}{\alpha}$. At this time, r covers a maximum distance of $\beta\tau = \beta\frac{2d_{12}}{\alpha}$. Set $d_{23} > \beta\frac{2d_{12}}{\alpha} + \frac{1}{\alpha} + \frac{d_{12}}{\sqrt{2}\alpha}$. r cannot simultaneously cover at least one of t_1 or t_2 at time 0, and at least one of t_3 or t_4 at time τ, which is a contradiction. Hence, ALG cannot maintain an α approximation of Q^* for all times.

■

The instance created in the proof above uses minimum flying altitude $z_{\min} = 1$ and camera field-of-view angle $\phi = \pi/4$. We can create corresponding instances for any other values of these parameters. In light of Theorem 1, we drop the requirement that all targets must always be tracked. Instead we focus on the case when the robots are allowed to track a fraction of all targets.

V. 1/2 Approximation Algorithm

In this section, we present the main algorithm to maximize the number of targets tracked, or maximize the quality of tracking. We divide the time into rounds of fixed duration. We consider the scenario where using measurements from previous rounds, the robots are able to predict the motion of the targets for the current round. For each robot, we create a set of m candidate trajectories that can be followed for the current round. For example, these trajectories can be generated using existing grid-based or sampling-based methods [12]. Our goal is to choose a trajectory for each of the robots for the current round.

Figure 3 shows a simple instance with two robots, and three candidate trajectories each robot can follow. The camera footprint along two such trajectories as well as the set of targets covered by these trajectories are shown. Note that the trajectories need neither be restricted to any discretized grid, nor have uniform length or uniform speed.

Let $R_j(x)$ denote the set of targets predicted to be covered by x^{th} trajectory followed by j^{th} robot. We create a set system $(X, \mathcal{R})$ where X is the set of all targets and $\mathcal{R}$ is a collection of all $R_j(x)$ sets. We group sets in $\mathcal{R}$ into k collections, one per robot. Each group contains m sets each. That is,

$$\mathcal{R} = \{ \underbrace{R_1(1), \ldots, R_1(m)}_{\text{candidate trajectories for } r_1}, \ldots, \underbrace{R_k(1), \ldots, R_k(m)}_{\text{candidate trajectories for } r_k} \} \quad (1)$$

A valid assignment of trajectories can be represented by a map, $\sigma : [1, \ldots, k] \to [1, \ldots, m]$, indicating trajectory $\sigma(j)$ (i.e., the set $R_j(\sigma(j))$) is chosen for the j^{th} robot. We can remove a target from the set $R_j(x)$ if it does not satisfy a given minimum quality of tracking requirement.

A. Maximizing Number of Targets

First consider the case of maximizing the number of targets tracked by k robots. This problem is a generalization of the maximum coverage problem [13] stated as: *choose k subsets to maximize the cardinality of the union of all subsets*. In our case, we cannot arbitrarily pick k subsets since they must belong to distinct groups (i.e., the same robot cannot be assigned to two trajectories).

The maximum coverage problem, under group constraints, can be stated as: *choose k subsets of $\mathcal{R}$ given by a map,*

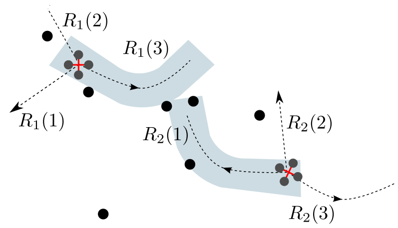


Fig. 3. At the start of each round, we have a set of m candidate trajectories per robot. The trajectories may be non-uniform and of varying speeds. Using the predicted motion of the targets, we can determine which targets will be covered for a given trajectory and the corresponding quality of tracking.

$\sigma : [1, \ldots, k] \to [1, \ldots, m]$ *such that the union of all subsets is maximized*. The constraint that the same robot cannot be assigned to two trajectories is enforced by requiring the output be a map σ. This problem is known as the Maximum Group Coverage (MGC) problem. Chekuri and Kumar [1] proved that the greedy algorithm yields a $1/2$ approximation for MGC. Their algorithm can directly be applied to track half the number of targets as an optimal algorithm. Our contribution is to extend the analysis to the weighted case.

B. Maximizing Quality of Tracking

For the case of maximizing the overall quality of tracking, we formulate a weighted version of MGC. Let $q_i(R_j(x))$ be the quality of tracking target t_i with robot r_j following the x^{th} trajectory. $q_i(R_j(x))$ can represent the expected quality of tracking as described in Section III. The weight of any set $R_j(x) \in \mathcal{R}$ is given by the sum of qualities of all targets tracked by $R_j(x)$. The objective is to maximize the sum of quality of tracking for all targets[1].

The greedy algorithm for the unweighted MGC can be modified for the weighted setting (Algorithm 1). In each iteration, we choose a set $R_j(x)$ greedily that maximizes the total weight. We add $R_j(x)$ to the solution, and discard all other sets belonging to the same group, i.e., all other candidate trajectories for the same robot r_j. This proceeds until we have chosen a trajectory for all robots.

Algorithm 1: Greedy Weighted MGC Algorithm

```
C ← ∅, I ← ∅
for p = 1 to k do
    Find R_i(x) such that Q(R_i(x) ∪ C) is greatest, and
    i ∉ I
    σ(i) ← x
    C ← C ∪ R_i(x)
    I ← I ∪ {i}
end
Return σ
```

Theorem 2 *Algorithm 1 gives a $(1/2 - \epsilon)$ approximation for the weighted MGC problem for any $\epsilon > 0$ in polynomial time.*

The analysis by Chekuri and Kumar [1] for the unweighted case can be modified for this weighted case. We present our full proof in the accompanying technical report [14], for completeness.

We now evaluate the greedy algorithm through simulations and preliminary experiments.

VI. SIMULATIONS

In this section, we describe our implementation of the algorithm, and evaluate its performance through simulations. We carried out the simulations using the SwarmSimX simulation environment [15]. SwarmSimX is a real-time multi-robot simulator designed for modeling rigid-body dynamics in 3D environments. Models of the MikroKopter Quadrotor[2] were used to simulate the motion of the robots.

For simulating the targets, we generated random trajectories as follows. Each target randomly chooses a speed and direction and moves along this direction for a random interval of time, drawn from a normal distribution. This class of trajectories is motivated by wildlife monitoring applications, where foraging animals have been found to follow such mobility models [16]. The mean and standard deviation of the normal distribution were set to 10 s and 1 s, respectively in the simulations.

The target trajectories were restricted to 20×20 m square on the ground plane. The initial locations of all targets were chosen uniformly at random near the robot locations. A moving average filter of window length 5 running at 10 Hz was used to estimate the position and velocity of the observed targets for the next planning round. A measurement for a target was obtained only if it was contained within the field-of-view of some robot.

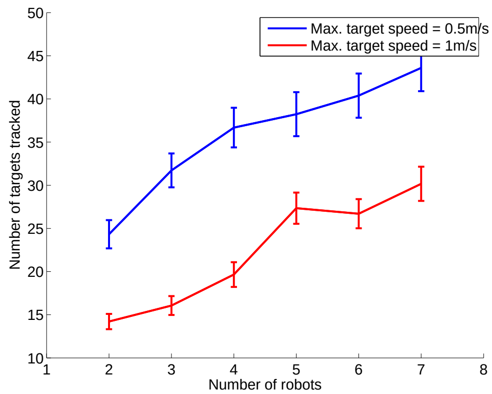
For each robot, we created the following set of candidate trajectories: (a) stay in place, and (b) radially symmetric along 8 horizontal directions with a speed of 0.5 m/s. Thus, each robot could choose from a set of 9 trajectories in a round. Each round was set to a duration of 2 s. A trial consisted of 50 rounds.

Figures 4(a) and 4(b) show the effect of the number of robots and the maximum speed of the targets. As expected, the number of tracks and quality of tracking increases as the number of robots increase. Increase in the maximum speeds of the targets has the effect of spreading them further apart, which further reduces the number of targets that can be tracked. For these trials, the height of the robots was fixed to 3.5 m (i.e., the size of the camera footprint was fixed). Figure 5 shows the total number of targets tracked in one representative trial as a function of the time. Once the robots have lost track of a particular target, they do no receive any position information about that target. Thus, they cannot predict the future locations for a lost target, unless it appears again in the field-of-view of some robot.

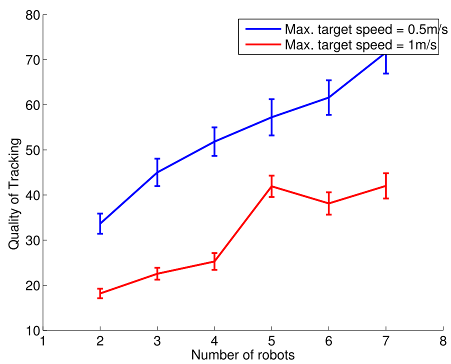
For the simulations, we did not incorporate the uncertainty due to sensing. In the next section, we validate the uncertainty model and present results from a preliminary experiment using 4 aerial robots.

[1]The bottleneck version of maximizing the minimum quality of tracking over all targets cannot be applied since not all targets are tracked.

[2]http://mikrokopter.de



(a)



(b)

Fig. 4. (a) Number of targets covered out of 50 targets in the environment. (b) The average quality of tracking. The weight $q_i(R_j(x)$ is computed as the inverse of the minimum distance between the target and the robot along $R_j(x)$.

VII. EXPERIMENTS

In order to validate our sensing model and the algorithm, we performed trials on an indoor setup (Figure 6). The setup consisted of four quadrotors controlled using the TeleKyb framework [17]. All robots communicated directly with a central computer via a wireless XBee link. Each robot was fitted with a downward facing camera. The cameras streamed the live images wirelessly directly to the central computer. An indoor motion capture system was used for position feedback, while orientation is stabilized onboard.

A. Validating the Sensing Model

We first conducted trials to validate the sensing model presented in Section III. A robot was programmed to fly along a given trajectory at heights of $1\,\text{m}$ and $1.5\,\text{m}$. The motion of the robot was smoothed, so as to ensure that the roll and pitch angles remained close to zero. Colored balls were placed on the ground (Figure 6). The pink and the yellow colored balls were fixed to motion capture markers to record their ground truth locations. All cameras were calibrated to obtain they camera parameters.

Figure 7 shows an image obtained using the on-board camera, along with the estimated and true locations of the balls. The backprojection area was computed considering ± 5

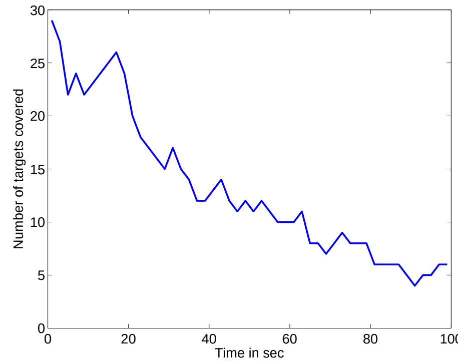


Fig. 5. The number of targets tracked in one trial. As the targets spread the total number of targets that can be tracked decreases. Once a target moves out of the field-of-view, the robots cannot predict their future locations.

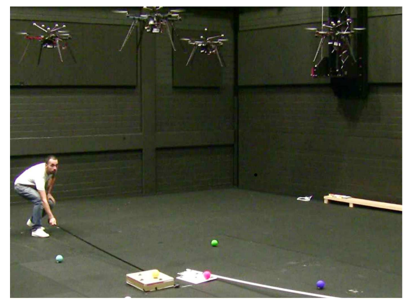
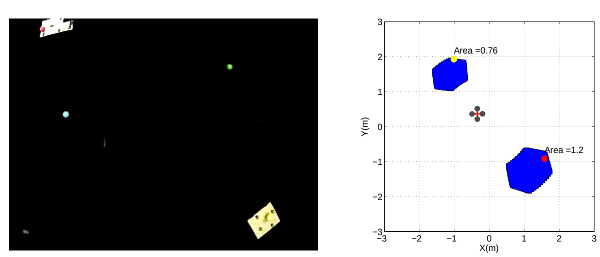


Fig. 6. Experimental setup. Each robot is fitted with a downward facing wireless camera. All robots directly communicate with a central computer.

maximum measurement error in pixels, $\pm 5\,\text{cm}$ maximum error in robot position, $\pm\pi/18$ radians maximum error in the yaw angle, and $\pm\pi/48$ radians maximum error in the roll and pitch angles. The average area of backprojection (for 50 images which contained either the yellow or pink balls) was $0.46\,\text{m}^2$. The average error between the centroid of the projected area and the true location was $0.28\,\text{m}$, with a standard deviation of $0.3\,\text{m}$.

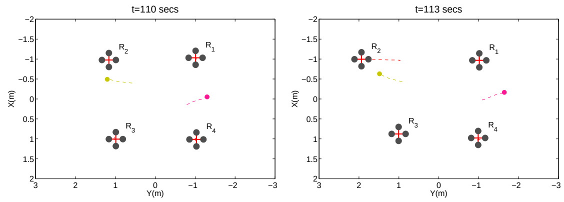
B. Tracking Experiment

We implemented the greedy algorithm on the four robots. The controller on-board the robot was set to operate the robots smoothly in near-hovering mode at an average speed $0.5\,\text{m/s}$. Each round lasted for 3 seconds. The pink and yellow balls were moved manually (Figure 6). For this trial,

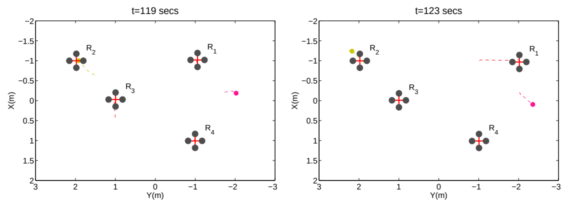


(a) (b)

Fig. 7. Validating the sensing model. (a) On-board camera image. (b) The true target location (colored circles) in the global frame, and the estimated locations using the method described in Section III.



(a) At $t = 110\,\mathrm{s}$, R_3 chose the trajectory moving to the left to keep tracking the yellow target.



(b) At $t = 119\,\mathrm{s}$, R_1 chose the trajectory moving to the right to keep tracking the pink target.

Fig. 8. Start (left figures) and end (right figures) of two rounds. Dashed trail shows the locations of the robots and targets in the preceding 5 secs.

the locations of the targets were obtained from the motion capture system. The robots used a moving average filter to predict the locations of the targets, based on previous measurements. A radius of $\sqrt{2}\,\mathrm{m}$ was found empirically to correspond to the camera footprint when the robots operated at a height of $2.5\,\mathrm{m}$. The robots had one of the four grid neighbors in the $z = 2.5\,\mathrm{m}$ plane as candidate trajectories.

Figure 8 shows the locations of the robots and the targets before and after two key rounds: at times $110\,\mathrm{s}$ and $119\,\mathrm{s}$. The two rounds show events when the robots predicted that the target would move out of the coverage area in the next round. Hence, as an outcome of the greedy algorithm, the robots chose corresponding trajectories in order to continue to track the targets.

The sensing validation and tracking trials presented here demonstrate a proof-of-concept implementation of the components of our system. Our ongoing efforts are directed towards performing large scale experiments with this system.

VIII. Conclusion

In this paper, we studied a visual tracking problem in which a team of robots equipped with cameras are charged with tracking the locations of targets moving on the ground. We discussed the sources of uncertainty that affect the quality of estimating the locations of ground targets using overhead images. We showed the infeasibility of tracking all targets while maintaining the optimal quality of tracking, or any factor of the optimal quality, at all times. We then formulated the target tracking problem where the goal is to assign trajectories for each robot in order to maximize the quality of tracking. When we are given a set of candidate robot trajectories, we showed how the problem can be posed as a combinatorial optimization problem. A simple and easy-to-implement greedy algorithm applied to this problem yields a 1/2 approximation. Finally, we presented results from simulations and preliminary experiments validating the sensing model and demonstrating the feasibility of implementing the algorithm. Future work includes investigating the problem under inter-robot communication constraints, and conducting larger scale experimental validation.

IX. Acknowledgements

We thank the Max Planck Institute for Biological Cybernetics in Germany, for the experimental setup, and Simon Hartmann and Massimo Basile for their help during the experiments.

References

[1] C. Chekuri and A. Kumar, “Maximum coverage problem with group budget constraints and applications,” in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2004, pp. 72–83.

[2] J. R. Spletzer and C. J. Taylor, “Dynamic sensor planning and control for optimally tracking targets,” *The International Journal of Robotics Research*, vol. 22, no. 1, pp. 7–20, 2003.

[3] E. W. Frew, “Observer trajectory generation for target-motion estimation using monocular vision,” Ph.D. dissertation, Stanford University, 2003.

[4] S. M. LaValle, H. H. González-Banos, C. Becker, and J.-C. Latombe, “Motion strategies for maintaining visibility of a moving target,” in *IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 1997, pp. 731–736.

[5] N. R. Gans, G. Hu, K. Nagarajan, and W. E. Dixon, “Keeping multiple moving targets in the field of view of a mobile camera,” *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 822–828, 2011.

[6] S. Bespamyatnikh, B. Bhattacharya, D. Kirkpatrick, and M. Segal, “Mobile facility location,” in *Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile computing and Communications*. ACM, 2000, pp. 46–53.

[7] S. Durocher, “Geometric facility location under continuous motion,” Ph.D. dissertation, The University of British Columbia, 2006.

[8] M. de Berg, M. Roeloffzen, and B. Speckmann, “Kinetic 2-centers in the black-box model,” in *Proceedings of the 29th Annual on Symposium on Computational Geometry*. ACM, 2013, pp. 145–154.

[9] S. Martinez, J. Cortes, and F. Bullo, “Motion coordination with distributed information,” *IEEE Control Systems*, vol. 27, no. 4, pp. 75–88, 2007.

[10] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, “Eyes in the sky: Decentralized control for the deployment of robotic camera networks,” *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.

[11] S. Thrun, W. Burgard, D. Fox, *et al.*, *Probabilistic robotics*. MIT press Cambridge, 2005, vol. 1.

[12] C. J. Green and A. Kelly, “Toward optimal sampling in the space of paths,” *Springer Tracts in Advanced Robotics*, vol. 66, pp. 281–292, 2010.

[13] D. S. Hochbaum and A. Pathria, “Analysis of the greedy approach in problems of maximum k-coverage,” *Naval Research Logistics*, vol. 45, no. 6, pp. 615–627, 1998.

[14] P. Tokekar, V. Isler, and A. Franchi, “Multi-target visual tracking with aerial robots,” Department of Computer Science & Engineering, University of Minnesota, Tech. Rep. 14-013, June 2014, http://www.cs.umn.edu/research/technical_reports/view/14-013.

[15] J. Lächele, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, “SwarmSimX: Real-time simulation environment for multi-robot systems,” in *3rd International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, Tsukuba, Japan, Nov. 2012.

[16] S. Benhamou, “How many animals really do the levy walk?” *Ecology*, vol. 88, no. 8, pp. 1962–1969, 2007.

[17] V. Grabe, M. Riedel, H. H. Bülthoff, P. Robuffo Giordano, and A. Franchi, “The TeleKyb framework for a modular and extendible ROS-based quadrotor control,” in *6th European Conference on Mobile Robots*, Barcelona, Spain, Sep. 2013, pp. 19–25.