# Occlusion-Aware Multi-Robot 3D Tracking

Karol Hausman[1] Gregory Kahn[2] Sachin Patil[2] Jörg Müller[1]
Ken Goldberg[2,3] Pieter Abbeel[2] Gaurav S. Sukhatme[1]

***Abstract*— We introduce an optimization-based control approach that enables a team of robots to cooperatively track a target using onboard sensing. In this setting, the robots are required to estimate their own positions as well as concurrently track the target. Our probabilistic method generates controls that minimize the expected uncertainty of the target. Additionally, our method efficiently reasons about occlusions between robots and takes them into account for the control generation. We evaluate our approach in a number of experiments in which we simulate a team of quadrotor robots flying in three-dimensional space to track a moving target on the ground. We compare our method to other state-of-the-art approaches represented by the random sampling technique, lattice planning method, and our previous method. Our experimental results indicate that our method achieves up to 8 times smaller maximum tracking error and up to 2 times smaller average tracking error than the next best approach in the presented scenarios.**

## I. Introduction

Tracking a moving target has many potential applications in various fields. For example, consider a search and rescue scenario where autonomous ground robots are deployed to assist disaster victims but they are not able to localize themselves in an unknown environment. In this case, one could use flying robots that due to their higher altitude can take advantage of, e.g., GPS signals on the one hand and can help to localize the ground robots by observing them on the other hand. Although it comes at the cost of higher complexity in motion planning, using multiple cooperatively controlled robots in this setting provides advantages, such as increased coverage and robustness to failure.

Consider a scenario depicted in Fig. 1 where a team of aerial robots equipped with onboard cameras is tasked with tracking and estimating the position of a mobile robot with respect to a global frame of reference. Some of the quadrotors may be in the field of view of the external global sensor while others are not. The ideal cooperative control algorithm for this team would take into account all visibility constraints and uncertainties in order to establish a configuration of quadrotors that propagates position information from the global sensors through the quadrotors to the target.

We introduce a centralized planning approach that generates controls for multiple robots and minimizes the uncertainty of the tracked target. At any particular instance an individual quadrotor may not be able to see the target due to occlusions or the sensor's limited field-of-view. To formulate a cost function that considers these discontinuities in the sensor domain, we extend the approach from [26] to multiple robots and penalize the trace of the target's covariance. Using this cost function, we employ an optimization framework to find locally optimal controls. We execute these controls in the Model Predictive Control (MPC) fashion and estimate the state of the quadrotor team and target using an Extended Kalman Filter (EKF) [30].

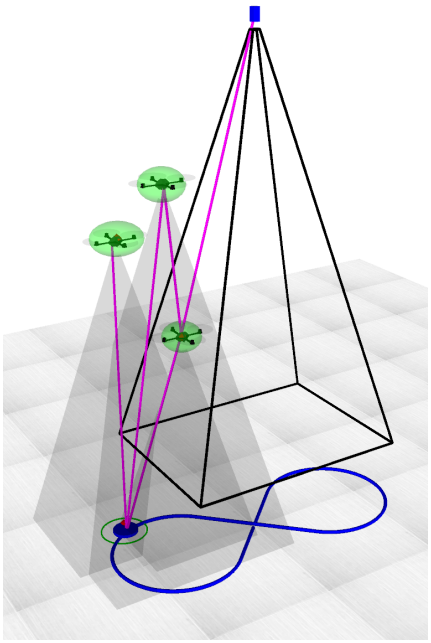

Fig. 1. Three quadrotors cooperatively tracking a target following a figure-eight trajectory (blue line). The green ellipsoids show the 3-dimensional covariances of quadrotors' positions. The green ellipse represents the 2-dimensional covariance of the position of the target. Magenta lines depict the measurements between quadrotors and the global camera (blue cube at the top). How should the quadrotors move in order to minimize the uncertainty of the target?

The novel contributions of this paper are as follows: a) we take into account sensing discontinuities caused, for example, by occlusions in different multi-robot configurations in a manner that is amenable to continuous optimization, and b) we generate 3D positional control inputs for all the quadrotors. We evaluated our approach in a number of simulations and compared it to our previous method in which we introduced cooperative multi-robot control with switching of sensing topologies [12, 13].

The video of the simulation experiments that present the results of our method is available at:

[1]Karol Hausman, Jörg Müller and Gaurav S. Sukhatme are with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA. hausman@usc.edu

[2]Gregory Kahn, Sachin Patil, Ken Goldberg, and Pieter Abbeel are with the Department of Electrical Engineering and Computer Sciences, University of California Berkeley, Berkeley, CA 94720, USA.

http://tinyurl.com/iros16tracking.

## II. Related Work

The task of cooperative target tracking has been addressed in various ways. Many researchers considered centralized [6, 8], decentralized [1, 23, 25], and distributed [17, 18, 31] approaches to control multiple aerial or ground robots. Nevertheless, all of the above approaches do not take into account the position uncertainty of the robots that are deployed to perform the tracking task. The position of the robots is assumed to either be known or independently obtained with high accuracy. In this work, we consider both the position uncertainty of the tracked target as well as the uncertainty in the robots' poses.

The problem of target localization is very similar to the task of target tracking. There have been many authors that worked on target localization [9, 10, 11, 14, 29] in a multi-robot scenario with onboard sensing. It is worth noting that these approaches are implemented in a distributed fashion which makes them well-suited for multi-robot scenarios with limited communication. One of the simplifications introduced in these approaches, however, is to limit the robots to planar movements and disable the possibility that the robots can be perceived by each other. In our work, we relax these assumptions and show how to cope with occlusions between different robots. Furthermore, the above-mentioned approaches use the mutual information measures as an uncertainty measure. In contrast, we use the trace of the covariance. We chose this utility measure because it copes better with degenerate covariances (and equally well with circular covariances) than the mutual information approach, as shown by Beinhofer et al. [4] and our previous work [13]. For more details please see [13].

Cooperative localization (CL) is the use of multiple robots for accurate state estimation [24, 33]. These works mainly address state estimation and localization, while we focus on tracking and control generation.

Another tracking approach was proposed by Ahmad and Lima [2] where the authors weight the observations of individual robots based on their localization uncertainty. Zhou and Roumeliotis [32] proposed a similar approach but their focus is on the non-convex optimization of the cost function and evaluation of different sensor models. In both of these approaches, the authors decouple the target tracking from the robots' localization, which does not take into account the high correlation of the target's and the robots' position estimates.

Lima et al. [21] propose a distributed formation control approach for target tracking. Although the target is tracked using a modified particle filter, the non-linear model-predictive controller models the uncertainty of the target using a covariance-based method that does not account for occlusions. Our method uses a similar non-linear model-predictive controller formulation that does account for occlusions.

Another approach by Ahmad et al. [3] addresses the problem of cooperative target tracking and robot localization using least square minimization. However, their graph optimization approach does not provide the uncertainty of the associated estimates, thus the optimization is unable to minimize the uncertainty.

There have been several optimization-based approaches [3, 15, 16] that tackle the problem of cooperative multi-robot localization using only onboard sensors. These approaches, however, provide the maximum-likelihood state estimates which does not directly minimize the uncertainty of the target pose.

In our previous work [12] we introduced the concept of level-based sensing topologies, which simplifies the reasoning about occlusions and reduces the considered control space of each robot to a two-dimensional plane. In this paper, we tackle the problem of generating locally optimal controls for each quadrotor in the full 3-dimensional space and we explicitly reason about occlusions in the trajectory optimization algorithm. In addition, we do not, as previously, apply the same controls over the look-ahead horizon in the optimization but consider arbitrary trajectories with different controls for different time steps. Moreover, by removing the level-based sensing topologies in this work, we enable more realistic simulations as the quadrotors are not assumed to instantly jump from one topology level to another.

## III. Problem Definition

We propose a centralized approach where we jointly estimate the positions of the quadrotors and the target using an EKF. First, we describe our state estimation technique to then focus on the control generation and reasoning about occlusions.

### A. Assumptions

In the presented approach, we assume certain properties of the system that enable us to simplify the computation. First of all, we apply an EKF, hence the Markov assumption [30] that the measurements are conditionally independent given the joint state. We assume that there exists one global fixed sensor with limited range, therefore necessitating the use of a quadrotor team for tracking the target. We fix the orientation of the moving quadrotors so that one can only control their 3D position. For the moving target, a standard uncontrolled motion model is applied. We assume the target tracking never experiences complete failure, otherwise the scenario would evolve into a simultaneous exploration and tracking problem. For the part of the system responsible for the control generation, we follow Platt et al.[28] and assume that the maximum likelihood observation is obtained at each time step. This eliminates the stochasticity of the cost functions.

### B. State Estimation with EKF

*1) System Parametrization:* The state at time $t$ consists of $n$ individual quadrotor poses $\mathbf{x}_t^{(i)}, i \in [1, n]$ and the target pose $\mathbf{x}_t^{(targ)}$. Let $\mathbf{u}_t^{(i)}$ be the control input applied to the $i$th robot at time $t$. The joint state and control input are defined

as:

$$\mathbf{x}_t = [\mathbf{x}_t^{(1)}, \ldots, \mathbf{x}_t^{(n)}, \mathbf{x}_t^{(targ)}] \quad (1)$$

$$\mathbf{u}_t = [\mathbf{u}_t^{(1)}, \ldots, \mathbf{u}_t^{(n)}] \quad (2)$$

Let $\Sigma_t$ be the uncertainty covariance of the joint state.

The dynamics and measurement models for the joint state are given by the stochastic, differentiable functions $\mathbf{f}$ and $\mathbf{h}$:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{q}_t), \quad \mathbf{q}_t \sim N(\mathbf{0}, Q_t) \quad (3)$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{r}_t), \quad \mathbf{r}_t \sim N(\mathbf{0}, R_t) \quad (4)$$

where $\mathbf{q}_t$ is the dynamics noise, $\mathbf{r}_t$ is the measurement noise, and they are assumed to be zero-mean Gaussian distributed with state-dependent covariances $Q_t$ and $R_t$, respectively.

We consider two types of sensors and corresponding measurements: absolute (global) measurements, e.g., GPS, and relative measurements between two quadrotors or a quadrotor and the target, e.g., distance or relative pose measurements. The stochastic measurement function of the absolute sensors is given by:

$$\mathbf{z}_t^{(i)} = \mathbf{h}^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{r}_t^{(i)})$$

while the relative sensor model is:

$$\mathbf{z}_t^{(i,j)} = \mathbf{h}^{(i,j)}(\mathbf{x}_t^{(i)}, \mathbf{x}_t^{(j)}, \mathbf{r}_t^{(i,j)})$$

All measurement functions can be naturally extended for the joint state [22].

*2) Uncertainty Model:* Given the current belief $(\mathbf{x}_t, \Sigma_t)$, control input $\mathbf{u}_t$ and measurement $\mathbf{z}_{t+1}$, the belief dynamics is described by the EKF equations.

In order to model the discontinuity in the sensing domain, which can be caused either by a limited field of view, sensor failure or occlusions, we follow the method from [26] and introduce a binary vector $\delta_t \in \mathbb{R}^{\dim[\mathbf{z}]}$. The $k$th entry in the vector $\delta_t$ takes the value 1 if the $k$th dimension of $\mathbf{z}_t$ is available and a value of 0 if no measurement is obtained. We detail the method for computing $\delta_t$ in Sec. III-D.

The EKF update equations are as follows:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{0}) + K_t(\mathbf{z}_t - \mathbf{h}(\mathbf{x}_t)) \quad (5a)$$

$$\Sigma_{t+1} = (I - K_t H_t)\Sigma_{t+1} \quad (5b)$$

$$K_t = \Sigma_{t+1} H_t^\intercal \Delta_t [\Delta_t H_t \Sigma_{t+1} H_t^\intercal \Delta_t + W_t R_t W_t^\intercal]^{-1} \Delta_t \quad (5c)$$

$$\Sigma_{t+1} = A_t \Sigma_t A_t^\intercal + V_t Q_t V_t^\intercal \quad (5d)$$

$$A_t = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{0}), \quad V_t = \frac{\partial \mathbf{f}}{\partial \mathbf{q}}(\mathbf{x}_t, \mathbf{0}) \quad (5e)$$

$$H_t = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x}_{t+1}, \mathbf{0}), \quad W_t = \frac{\partial \mathbf{h}}{\partial \mathbf{r}}(\mathbf{x}_{t+1}, \mathbf{0}) \,, \quad (5f)$$

where $\Delta_t = \mathrm{diag}[\delta_t]$ and $\mathbf{z}_t$ is a measurement obtained at time step $t$.

It is worth noting that the Kalman gain update in Eq. 5c includes the binary matrix $\Delta_t$ to account for discontinuities in the sensor domains. Due to the Markov assumption, the individual measurements can be separately fused into the belief using the EKF update equations.

*3) Dynamics Model:* The dynamics of an individual quadrotor is given by: $\mathbf{f}^{(i)}(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{q}^{(i)}) = \mathbf{x}^{(i)} + \mathbf{u}^{(i)}\Delta t + \mathbf{q}^{(i)}$ where $\mathbf{x}^{(i)}, \mathbf{u}^{(i)} \in \mathbb{R}^3$ are the 3D position and velocity and $\Delta t$ is the length of a time step. In this paper, we use a constant velocity motion model for the target, however our approach generalizes to any uncontrolled motion model.

*4) Observation Model:* The cameras are assumed to be at the center of each quadrotor facing down. The absolute sensor provides the 3D position of the observed quadrotor/target as a measurement:

$$\mathbf{h}^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{r}_t^{(i)}) = \mathbf{x}_t^{(i)} + \mathbf{r}_t^{(i)}$$

The relative sensor model provides the position of the (observed) $j$th quadrotor/target relative to the (observing) $i$th quadrotor:

$$\mathbf{h}^{(i,j)}(\mathbf{x}_t^{(i)}, \mathbf{x}_t^{(j)}, \mathbf{r}_t^{(i,j)}) = (\mathbf{x}_t^{(j)} - \mathbf{x}_t^{(i)}) + \mathbf{r}_t^{(i,j)} \,.$$

### C. Control Generation using Optimization

At each time step $t$ we seek a set of control inputs $\mathbf{u}_{t:T=t+h}$ that for a time horizon $h$ minimizes the uncertainty of the target while penalizing collisions and the control effort. In order to minimize the uncertainty of the target, we, similarly to [13], penalize the trace of the target covariance. In consequence, it is possible for the quadrotor covariance to get large if the quadrotor is not near the other quadrotors or the global sensor. However, it is possible to bring the quadrotor back near the target if it is feasible within the time horizon of the controller. The other components included in the cost function are: the distance of every quadrotor pair to avoid collisions and the cost of the control effort. The final cost function is composed as follows:

$$c_t(\mathbf{x}_t, \Sigma_t, \mathbf{u}_t) = \alpha \,\mathrm{tr}(\Sigma_t^{(targ)}) + \beta c_{collision}(\mathbf{x}_t) + \gamma ||\mathbf{u}_t||_2^2$$

$$c_T(\mathbf{x}_T, \Sigma_T) = \alpha \,\mathrm{tr}(\Sigma_T^{(targ)}) + \beta c_{collision}(\mathbf{x}_T)$$

$$c_{collision}(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \max(d_{max} - ||\mathbf{x}^{(j)} - \mathbf{x}^{(i)}||_2^2, 0)$$

where $\alpha, \beta$, and $\gamma$ are user-defined scalar weighting parameters and $d_{max}$ is the maximum distance for which the collision cost takes effect.

The final objective function is:

$$\min_{\mathbf{x}_{t:T}, \mathbf{u}_{t:T}} E[c_T(\mathbf{x}_T, \Sigma_T) + \sum_{t}^{T-1} c_t(\mathbf{x}_t, \Sigma_t, \mathbf{u}_t)] \quad (6)$$

$$\text{s.t. } \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{0})$$

$$\mathbf{x}_t \in \mathscr{X}_{feasible}, \ \mathbf{u}_t \in \mathscr{U}_{feasible}$$

$$\mathbf{x}_0 = \mathbf{x}_{init}, \ \Sigma_0 = \Sigma_{init}$$

For trajectory optimization, one can solve a nonlinear optimization problem to find a locally optimal set of controls [27]. In this work, we used sequential quadratic programming (SQP) to locally optimize the non-convex, constrained optimization problem. The innermost QP solver was generated by a numerical optimization code framework called FORCES [7]. FORCES generates code for solving

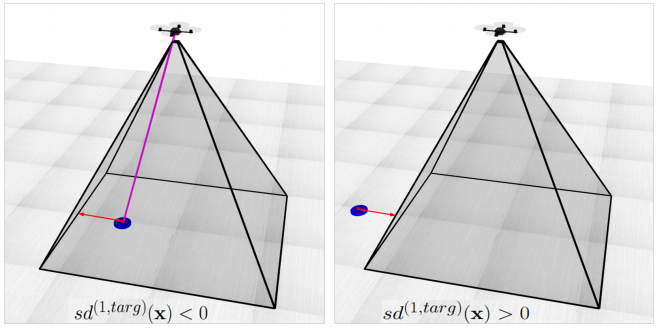


Fig. 2. Signed distance if the quadrotor/target is inside the view frustum (left) and outside the view frustum (right).

QPs that is specialized for convex multistage problems such as trajectory optimization. The running time of each optimization is $O(\dim[\mathbf{x}]^3 T)$ [27].

Given the output controls $\mathbf{u}_{t:T\ \ 1}$ computed using trajectory optimization, we follow the model predictive control paradigm [5] by executing a subset of the optimized controls at each time step and then replanning. We use the previously computed controls as an initialization to warm start the optimization.

### D. Reasoning about Occlusions

The absolute/relative position of a quadrotor/target may not be observable due to occlusions from other quadrotors and the limited field-of-view of the sensors. As previously mentioned, we model this discontinuity with the binary variable $\delta$ (according to [26]). In order to make the objective function differentiable, we approximate $\delta$ with a sigmoid function. It is worth noting that this $\delta$ is only required in the optimization step, the state estimation EKF step remains as defined previously.

Let $\mathtt{sd}^{(i,j)}(\mathbf{x})$ be the signed distance of $\mathbf{x}^{(j)}$ to the field-of-view of the $j$th quadrotor (see Fig. 2). The signed distance is negative if the $j$th quadrotor is visible and positive otherwise. We introduce the parameter $\alpha$ which determines the slope of the sigmoidal approximation. The sigmoidal approximation of the measurement availability $\delta$ is given by:

$$\delta^{(i,j)} = \frac{1}{1 + \exp[-\alpha \cdot \mathtt{sd}^{(i,j)}(\mathbf{x})]} \tag{7}$$

For more details on the sigmoidal approximation of the availability of the measurement we refer the reader to [26].

To calculate the signed distance of $\mathbf{x}^{(j)}$ to the field-of-view of $\mathbf{x}^{(i)}$, we first represent the field-of-view as a truncated view frustum with a minimum and maximum distance given by the sensor model as depicted in Fig. 3. If $\mathbf{x}^{(j)}$ is outside of the view frustum, $\mathtt{sd}^{(i,j)}(\mathbf{x})$ is the distance of $\mathbf{x}^{(j)}$ to the view frustum as shown in the right part of Fig. 2. If $\mathbf{x}^{(j)}$ is inside the view frustum and there are no occlusions, the signed distance is computed as shown in the left part of Fig. 2. In the presence of occlusions, we first determine the shadows of all the occlusions in the plane of $\mathbf{x}^{(j)}$. In the next step, we use an open source 2D polygon clipping library - GPC[1] to generate the 2D polygon field-of-view, and then calculate the signed distance. Fig. 3 shows an example of a signed distance function in the presence of an occlusion.

[1] http://www.cs.man.ac.uk/~toby/gpc/

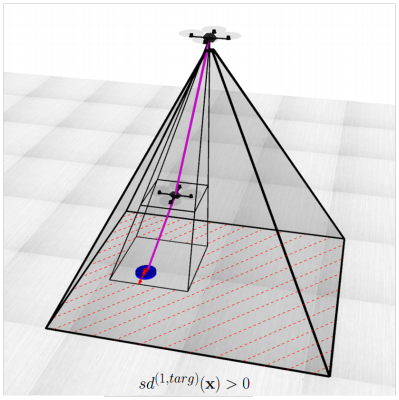


Fig. 3. Signed distance function in the presence of occlusions. First, we determine the shadows of all the occlusions such that the resulting field-of-view (the shaded area) is calculated. The signed distance is computed as the distance to the field-of-view.

## IV. Evaluation and Discussion

We evaluated our approach in a number of simulation experiments. The simulation environment consists of a fixed global down-looking camera attached 4 meters above the origin of the coordinate system, a ground robot target and a varying number of quadrotors equipped with down-looking cameras. These scenarios are similar to the real robot experiments presented in [13]. Each quadrotor is controlled through the velocity commands $\mathbf{u}^{(i)} = [v_x, v_y, v_z]$. For each quadrotor, $\mathscr{X}_{feasible}$ is unrestricted while $\mathscr{U}_{feasible}$ is define as $-\mathbf{1} \leq \mathbf{u}^{(i)} \leq \mathbf{1}$. The cost function parameters are: $\alpha = 10^6$, $\beta = 10^2$, and $\gamma = 1$. The state of the target consists of its position and velocity $\mathbf{x}^{(targ)} = [x, y, v_x, v_y]$ and the target moves on the XY-plane. The length of the simulation time step is equal to $0.1s$. All camera sensors in this setup have the same properties as described in Sec. III-B.4, and can detect objects in a 3-meter high truncated pyramid. Consequently, the global camera is not able to see the target directly. The camera measurement standard deviation is set to $0.02m$, based on the average standard deviation of a commodity depth sensor [19]. In addition, the measurement covariance scales quarticly with the distance to the measured quadrotor (it is equal to $0.02$ only at the zero distance). In order to make simulations even more realistic, we also introduce motion noise with standard deviation equal to $0.1m/s$. Each experiment uses a different random seed. We evaluated our approach for 3 different target trajectories, including a figure eight trajectory (see Fig. 1), a spiral trajectory of a approximate size of 1.6mx1.6m, and a random walk trajectory with a step size of 0.03m. The figure eight and spiral trajectories simulate an adversarial target that is attempting to evade being tracked, while the random walk trajectory simulates a non-adversarial (but still non-cooperative) target.

An example of a system setup with the field of view of each camera is depicted in Fig. 1, where the target trajectory is a figure eight. The target trajectory is not known to the quadrotors. Our system aims to estimate the position of

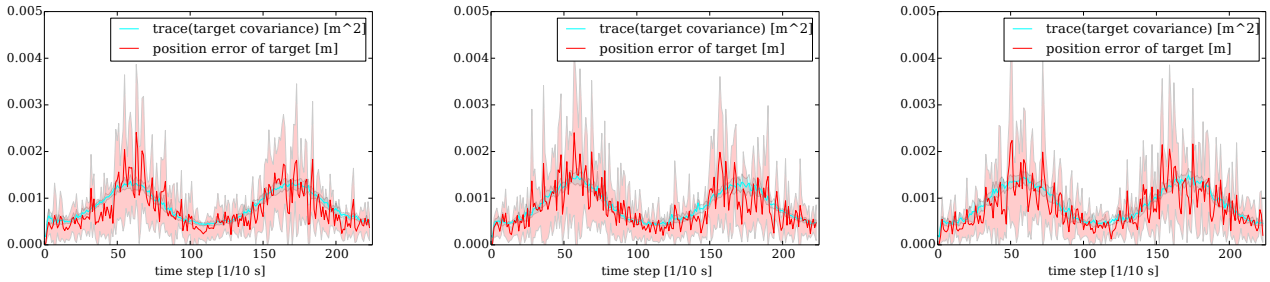


Fig. 4. Comparison of different time horizons in the optimization step. The time horizons are 2, 5, and 10 time steps, respectively. Evaluation measures are shown with 95% confidence intervals over 10 runs with 3 quadrotors. Results are presented only for the figure eight trajectory for brevity, however, we obtained similar results for other trajectories, i.e. spiral and random walk. There is no significant difference in the tracking error for these time horizons.

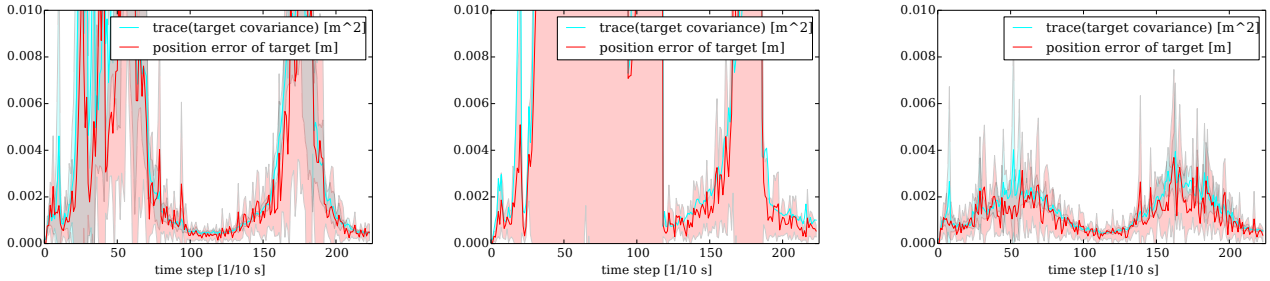


Fig. 5. Comparison between different sampling approaches and the optimization approach presented in this paper. From left to right: random sampling, lattice sampling and optimization. Evaluation measures are shown with 95% confidence intervals over 10 runs with 3 quadrotors. Results are presented only for the figure eight trajectory for brevity, however, we obtained similar results for other trajectories, i.e. spiral and random walk. The quantitative results from this experiment for all the trajectories are summarized in Table I. The optimization approach yields better results than the other presented methods.

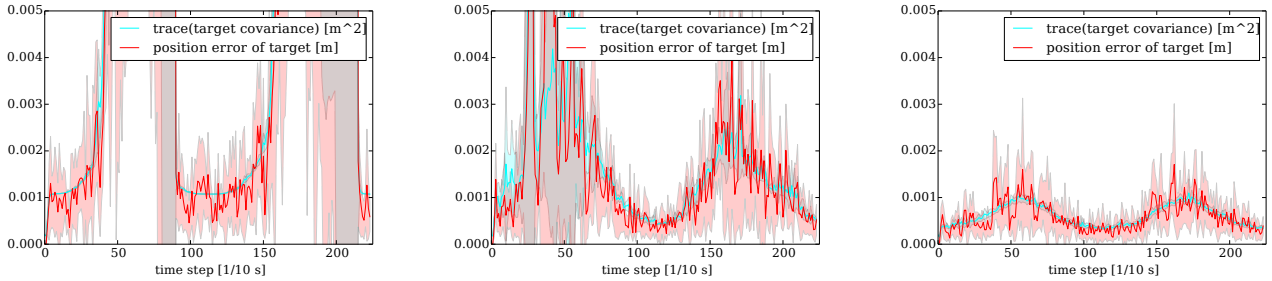


Fig. 6. Comparison of our optimization-based approach applied to different number of quadrotors used for the tracking task. From left to right the number of quadrotors are: 1, 3 and 5. Evaluation measures are shown with 95% confidence intervals over 10 runs. Results are presented only for the figure eight trajectory for brevity, however, we obtained similar results for other trajectories, i.e. spiral and random walk. The experiments confirms the intuition that deploying more quadrotors improves the tracking performance up to a point of diminishing returns.

the target as accurately as possible by actively controlling the quadrotors. Some of the simulation experiments are presented in the video attached to this submission[2].

### A. Experiments

We present different sets of experiments illustrating various evaluation criteria of our system.

*1) Time Horizon Experiment:* We evaluate the influence of different time horizons on the tracking accuracy. Fig. 4 shows the statistics for three different time horizons for the figure eight trajectory. We obtained similar results for the spiral and random trajectories. Based on these results, the time horizon has no significant effect on the performance in this setting, most likely because we are using MPC in which we plan over the given time horizon, execute the first control, then repeat. In addition, the optimization applies the constant velocity prediction of the target, which is constantly violated by the fact that the target is moving on a curved trajectory. We therefore set the time horizon to be 2 time steps for the remaining experiments.

*2) Sampling vs. Optimization Experiment:* We compare our optimization-based control generation with random sampling [6] and lattice planners [20]. Random sampling methods randomly sample controls to generate trajectories while lattice planners draw samples from a predefined manifold, which in our case was a sphere with radius equal to the maximum allowed control effort. For both methods, the cost of each resulting trajectory is evaluated and the controls corresponding to the minimum cost trajectory are executed. Fig. 5 shows the statistical results of the comparison between

[2] http://tinyurl.com/iros16tracking

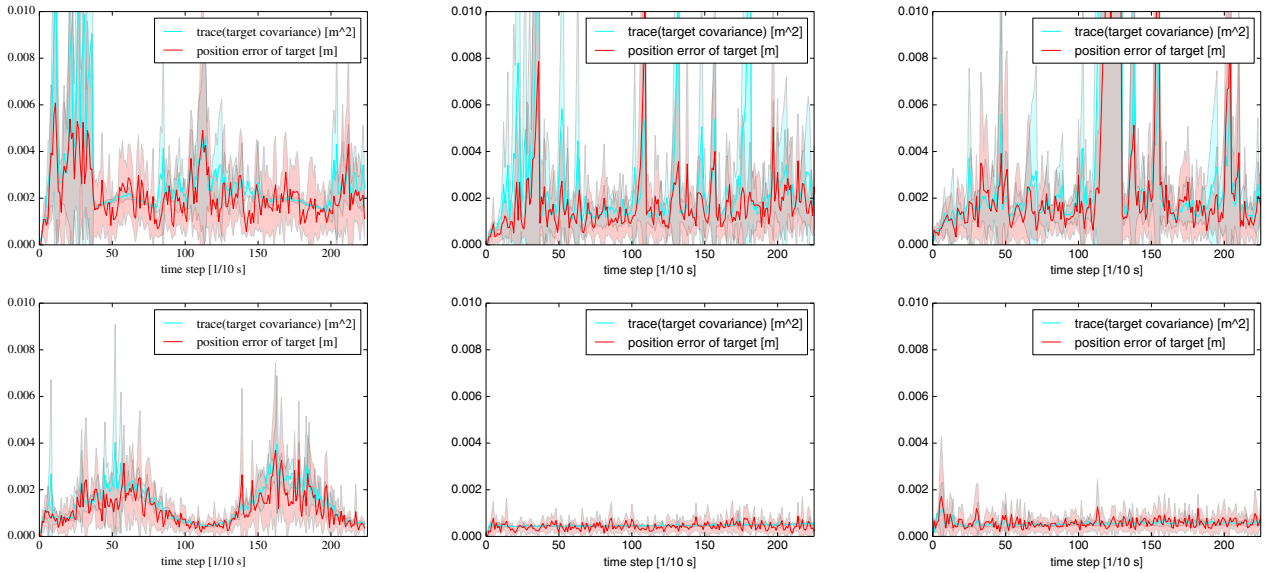


Fig. 7. Comparison between not taking occlusions into account in the optimization step (top) and accounting for occlusions using the signed distance function presented in this work (bottom). Results collected for 3 different trajectories: figure eight (left), spiral (middle) and random walk (right) Statistics and 95% confidence intervals over 10 runs with 3 quadrotors. Taking occlusions into account is beneficial especially at the spots right below the global camera where the quadrotors have to stay close to each other.

the sampling method, lattice sampling method and our approach for 3 quadrotors for the figure eight trajectory. In order to make a fair comparison for the sampling methods, we chose the number of samples such that the execution times per one optimization step of all the methods were similar and we averaged the results over 10 runs. One can notice a significantly larger tracking error and the trace of the target covariance in performance of both of the sampling methods compared to the trajectory optimization approach. It is worth noting that the lattice approach yields worse results than the random sampling approach. It is due to the fact that generating controls of the same magnitude throughout the trajectory causes the quadrotors to constantly overshoot the target.

*3) Number of Quadrotors Experiment:* We compared the tracking performance and the execution time for different numbers of quadrotors. The statistics presented in Fig. 6 demonstrates a visible improvement by using a larger number of quadrotors which confirms the intuition that larger teams facilitate more reliable and accurate target tracking. It is important to note that the quadrotors are able to leverage their large quantity despite the very narrow field of view of the cameras. This behavior is achieved because of the reasoning about occlusions which prevents the quadrotors to block each other's views. The average time for one optimization step for 3 quadrotors was equal to $0.156 \pm 0.002s$.

*4) Optimization with Occlusions Experiment:* To evaluate the importance of considering occlusions, we evaluated our method with and without reasoning about occlusions for a team of 3 quadrotors (Fig. 7) and measured the target covariance and target tracking error for all the trajectories. Of note is the much larger error and higher uncertainty in Fig. 7(top) than Fig. 7(bottom) for all the trajectories, indicating that reasoning about occlusions is particularly important for target tracking with multi-robot teams.

*5) Topology Switching Experiment:* We compare the method proposed in this paper to our previous target tracking method that introduced level-based sensing topologies and an efficient topology switching algorithm [12]. In this approach, the quadrotors are organized on different levels with an assumption that each level can only sense the adjacent level below it. At each time step the algorithm determines the planar controls for each of the quadrotors as well as determines whether to switch to one of the neighboring topologies by moving one of the quadrotors by one level up or down. This approach was introduced in order to avoid the reasoning about occlusions between quadrotors at different altitudes. However, a topology switch is assumed to be instantaneously completed, which is not realistic.

In order to make our approach and the level-based approach comparable for standard quadrotors, we introduce a delay of 3 time steps for the topology switch in order to realistically simulate a real quadrotor adjusting altitude. The performance of both algorithms for the figure eight trajectory is depicted in Fig. 8. The quadrotors perform better when explicitly reasoning about occlusions with our approach, especially when the target is far away from the global camera. This phenomena can be explained by Fig. 9. In the level-based approach, if the target is far away from the global camera the optimal topology is a chain topology. Our approach, however, is able to form a different sensing topology that is superior to the previous one due to its higher number of available measurements (see Fig. 9). This superior topology is not valid for the level-based approach as it violates the assumption of sensing only the adjacent level below the current level. This topology enables the algorithm to

| | level-based 2D opt. | lattice | sampling | 3D opt. no occ. | ours (3D opt.) |
|---|---|---|---|---|---|
| avg. max tracking error (8) [m] | $0.016 \pm 0.024$ | $0.41 \pm 1.00$ | $0.027 \pm 0.028$ | $0.0061 \pm 0.0072$ | $\mathbf{0.0037 \pm 0.0050}$ |
| avg. max tracking error rel. (8) | 26x | 1x | 15x | 67x | **111x** |
| avg. tracking error (8) [m] | $0.0030 \pm 0.0036$ | $0.037 \pm 0.073$ | $0.0035 \pm 0.0044$ | $0.0021 \pm 0.0010$ | $\mathbf{0.0012 \pm 0.00071}$ |
| avg. tracking error rel. (8) | 12x | 1x | 11x | 18x | **31x** |
| avg. max tracking error (spiral) [m] | $0.017 \pm 0.018$ | $0.070 \pm 0.14$ | $0.028 \pm 0.061$ | $0.011 \pm 0.027$ | $\mathbf{0.0015 \pm 0.00088}$ |
| avg. max tracking error rel. (spiral) | 4x | 1x | 2.5x | 6x | **47x** |
| avg. tracking error (spiral) [m] | $0.0010 \pm 0.0027$ | $0.0055 \pm 0.014$ | $0.0020 \pm 0.0043$ | $0.0018 \pm 0.0013$ | $\mathbf{0.00052 \pm 0.00023}$ |
| avg. tracking error rel. (spiral) | 5.5x | 1x | 3x | 3x | **11x** |
| avg. max tracking error (rw) [m] | $0.0065 \pm 0.013$ | $0.0071 \pm 0.018$ | $0.15 \pm 0.44$ | $0.018 \pm 0.033$ | $\mathbf{0.0042 \pm 0.011}$ |
| avg. max tracking error rel. (rw) | 23x | 21x | 1x | 8x | **36x** |
| avg. tracking error (rw) [m] | $0.00091 \pm 0.0011$ | $0.0012 \pm 0.00090$ | $0.0030 \pm 0.014$ | $0.0021 \pm 0.0024$ | $\mathbf{0.00063 \pm 0.00034}$ |
| avg. tracking error rel. (rw) | 3x | 2.5x | 1x | 1.5x | **5x** |

TABLE I. Quantitative results for the experiments with 3 quadrotors for 3 different trajectories: 8 - figure eight, spiral and rw - random walk. Max target tracking error was averaged over 10 runs, target tracking error was averaged over all time steps and 10 runs. Our approach yields significantly smaller error than the other baseline methods for all the tested trajectories.

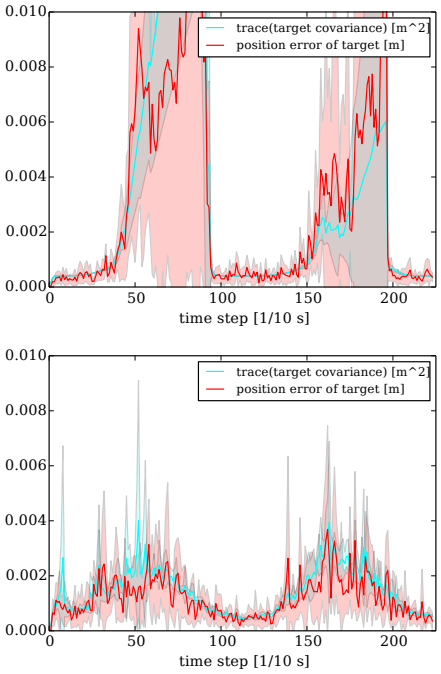


Fig. 8. Target tracking results for our previous level-based approach [12] (top) and the hereby presented method without explicitly reasoning about sensing topologies (bottom). Statistics and 95% confidence intervals over 10 runs with 3 quadrotors. Results are presented only for the figure eight trajectory for brevity, however, we obtained similar results for other trajectories, i.e. spiral and random walk. The quantitative results from this experiment for all the trajectories are summarized in Table I. The approach presented in this paper is more beneficial for the reasons explained in Fig. 9.

localize the quadrotor on the left side due to the measurement of the other quadrotor and have two localized quadrotors observe the target. It is worth noting that our novel approach can create a greater variety of sensing topologies compared to our previous approach without explicitly reasoning about sensing topologies.

*6) Average Error Comparisons:* Table I summarizes the statistics of the average tracking error and maximum average tracking error for different approaches and trajectories tested in our experiments. It is worth noting that our approach achieved up to 8 times smaller average maximum tracking error and up to 2 times smaller average tracking error

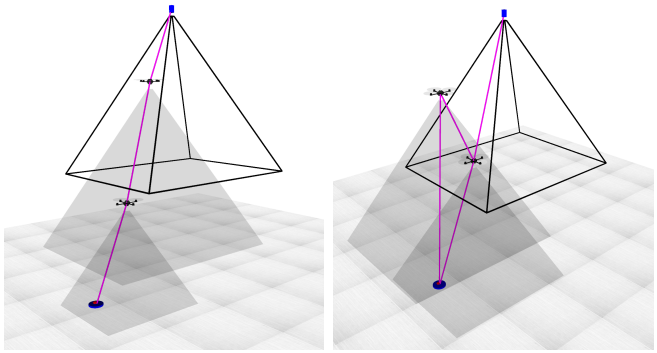

Fig. 9. An example of an advantage of our approach over the level-based approach. Left: most beneficial sensing topology for the level-based approach. The quadrotors form a chain topology as each level of quadrotors is allowed to sense only one level below it. Right: A superior sensing topology achieved by our novel method. The lower quadrotor is seen by the global sensor and the other quadrotor. The upper quadrotor can localize itself based on the observation of the lower quadrotor. Both quadrotors can observe the tracking target which provides more information than the level-based approach. There are no constraints for the topology levels.

compared to the the next best method. It is also apparent that our method yields the largest improvement for the spiral trajectory. The reason for that is that in the initial phases of this trajectory the target stays underneath the camera so the quadrotors have to stay close together to be in the global camera view frustum. Hence, it is likely that the quadrotors may occlude each other's views. Since our method explicitly reasons about these occlusions, it is able to outperform other methods.

## V. Conclusions and Future Work

We presented an optimization-based probabilistic multi-robot target tracking approach that efficiently reasons about occlusions. We evaluated our approach in a number of simulation experiments. We have compared our method to other baseline approaches such as random sampling, lattice sampling, and our previous work on sensing topologies [12]. Our experimental results indicated that our method achieves up to 8 times smaller average maximum tracking error and up to 2 times smaller average tracking error than the next best approach in the presented scenarios.

In future work, we plan to extend our centralized planning

approach to fully decentralized, distributed planning. This is advantageous in multi-robot settings with limited communication. Another challenge lies in the scalability of our method with respect to the number of quadrotors. In this work we evaluate all the possible pairs of the quadrotors for potential measurements which results in the $n^2$ complexity. We plan to develop intelligent sampling strategies to only check these pairs of the quadrotors that are likely to observe each other in the time horizon. Finally, we would like to further demonstrate the applicability of our approach in a real robot scenario.

## REFERENCES


[1] E. Adamey and U. Ozguner. A decentralized approach for multi-UAV multitarget tracking and surveillance. In *SPIE Defense, Security, and Sensing*, pages 838915–838915, 2012.

[2] A. Ahmad and P. Lima. Multi-robot cooperative spherical-object tracking in 3D space based on particle filters. *Robotics and Autonomous Systems*, 61(10):1084–1093, 2013.

[3] A. Ahmad, G.D. Tipaldi, P. Lima, and W. Burgard. Cooperative robot localization and target tracking based on least squares minimization. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 5696–5701, 2013.

[4] M. Beinhofer, J. Müller, A. Krause, and W. Burgard. Robust landmark selection for mobile robot navigation. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2638–2643, 2013.

[5] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[6] B. Charrow, V. Kumar, and N. Michael. Approximate representations for multi-robot control policies that maximize mutual information. *Autonomous Robots*, 37(4):383–400, 2014.

[7] Alexander Domahidi. Forces: Fast optimization for real-time control on embedded systems. *Available at forces. ethz. ch*, 2012.

[8] J. Fink, A. Ribeiro, V. Kumar, and B.M. Sadler. Optimal robust multihop routing for wireless networks of mobile micro autonomous systems. In *Military Communications Conference (MILCOM)*, pages 1268–1273, 2010.

[9] B. Grocholsky. *Information-theoretic control of multiple sensor platforms*. PhD thesis, 2002.

[10] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte. Information-theoretic coordinated control of multiple sensor platforms. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, volume 1, pages 1521–1526, 2003.

[11] B. Grocholsky, S. Bayraktar, V. Kumar, C.J. Taylor, and G. Pappas. Synergies in feature localization by air-ground robot teams. In *Experimental Robotics IX*, pages 352–361. 2006.

[12] K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G.S. Sukhatme. Cooperative control for target tracking with onboard sensing. In *International Symposium on Experimental Robotics (ISER)*, 2014.

[13] K. Hausman, J Müller, A. Hariharan, N. Ayanian, and G. Sukhatme. Cooperative multi-robot control for target tracking with onboard sensing. *International Journal of Robotics Research*, 2015.

[14] G.M. Hoffmann and C.J. Tomlin. Mobile sensor network control using mutual information methods and particle filters. *IEEE Transactions on Automatic Control*, 55(1):32–47, 2010.

[15] A. Howard, M.J. Matarić, and G.S. Sukhatme. Localization for mobile robot teams using maximum likelihood estimation. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 1, pages 434–439, 2002.

[16] G. Huang, R. Truax, M. Kaess, and J.J. Leonard. Unscented iSAM: A consistent incremental solution to cooperative localization and target tracking. In *Proc. of the European Conf. on Mobile Robots (ECMR)*, pages 248–254, 2013.

[17] B. Jung and G.S. Sukhatme. Tracking targets using multiple robots: The effect of environment occlusion. *Autonomous Robots*, 13(3):191–205, 2002.

[18] B. Jung and G.S. Sukhatme. Cooperative multi-robot target tracking. In *Distributed Autonomous Robotic Systems 7*, pages 81–90. 2006.

[19] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 2012.

[20] M. Likhachev and D. Ferguson. Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles. *Int. Journal of Robotics Research*, 28(8):933–945, 2009.

[21] P. Lima, A. Ahmad, A. Dias, A. Conceição, A. Moreira, E. Silva, L. Almeida, L. Oliveira, and T. Nascimento. Formation control driven by cooperative object tracking. *Robotics and Autonomous Systems*, pages 68–79, 2014.

[22] A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using relative observations. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2797–2802, 2005.

[23] R. Mottaghi and R. Vaughan. An integrated particle filter and potential field method for cooperative robot target tracking. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1342–1347, 2006.

[24] Anastasios Mourikis, Stergios Roumeliotis, et al. Performance analysis of multirobot cooperative localization. *Robotics, IEEE Transactions on*, 22(4):666–681, 2006.

[25] L-L. Ong, B. Upcroft, T. Bailey, M. Ridley, S. Sukkarieh, and H. Durrant-Whyte. A decentralised particle filtering algorithm for multi-target tracking across multiple flight vehicles. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4539–4544, 2006.

[26] S. Patil, Y. Duan, J. Schulman, K. Goldberg, and P. Abbeel. Gaussian belief space planning with discontinuities in sensing domains. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6483–6490. IEEE, 2014.

[27] Sachin Patil, Gregory Kahn, Michael Laskey, John Schulman, Ken Goldberg, and Pieter Abbeel. Scaling up gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation. In *Intl. Workshop on the Algorithmic Foundations of Robotics*, 2014.

[28] R. Platt Jr, R. Tedrake, L. Kaelbling, and T. Lozano-Perez. Belief space planning assuming maximum likelihood observations. 2010.

[29] E. Stump, V. Kumar, B. Grocholsky, and P.M. Shiroma. Control for localization of targets using range-only sensors. *Int. Journal of Robotics Research*, 28(6):743–757, 2009.

[30] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[31] Z. Wang and D. Gu. Cooperative target tracking control of multiple robots. *IEEE Transactions on Industrial Electronics*, 59(8):3232–3240, 2012.

[32] K. Zhou and S. Roumeliotis. Multirobot active target tracking with combinations of relative observations. *IEEE Transactions on Robotics*, 27(4):678–695, 2011.

[33] Ke X Zhou, Stergios Roumeliotis, et al. A sparsity-aware qr decomposition algorithm for efficient cooperative localization. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 799–806. IEEE, 2012.