

Motion Planning for Surveillance of Risk-Sensitive Areas by a Team of Unmanned Aerial Vehicles

Alex Wallar, Erion Plaku, Donald A. Sofge

Abstract—This paper proposes a motion-planning approach for persistent surveillance of risk-sensitive areas by a team of unmanned aerial vehicles (UAVs). The planner, termed PARCOV (Planner for Autonomous Risk-sensitive Coverage), seeks to (i) maximize the area covered by sensors mounted on each UAV; (ii) ensure that no region in the area remains too long without being surveyed; (iii) maintain high sensor data quality, and (iv) reduce detection risk. To achieve the stated objectives, PARCOV combines into a cost function the detection risk with an uncertainty measure designed to keep track of the regions that have been surveyed and the times they were last surveyed. PARCOV then reduces the uncertainty and detection risk by moving each quadcopter toward a low-cost region in its vicinity. By reducing the uncertainty, PARCOV is able to increase the area coverage and provide persistent surveillance. Moreover, a nonlinear optimization formulation is used to determine the optimal altitude for flying each quadcopter in order to maximize the sensor data quality while minimizing risk.

The efficiency and scalability of PARCOV is demonstrated in simulation using different risk models and an increasing number of UAVs to conduct risk-sensitive surveillance. Evidence of successful physical deployment is provided by experiments with AscTec Pelican quadcopters.

Note to Practitioners: This paper was motivated by the viability of UAVs to enhance automation in environmental monitoring, search-and-rescue missions, package delivery, and many other applications. As UAVs become an economically-feasible option for deployment, it becomes important to enhance their autonomy so as to increase productivity. In this paper, we develop an approach that uses simple interactions among UAVs to promote maximizing the area coverage while maintaining high sensor data quality and reducing the detection risk. The approach provides scalability, making it easy for UAVs to leave and join the mission as needed. Experimental results in simulation and with real quadcopters provide promising results. In future research, we would like to test and enhance the approach so that it can be used in various applications extending beyond laboratory testings.

I. INTRODUCTION

UAVs are becoming central in environmental monitoring, search-and-rescue missions, package delivery, target tracking, and many other applications. In addition, with UAVs such as ARDrone and AscTec Pelican quadcopters becoming more commercially available, they are an economically-feasible option for deployment in autonomous aerial missions.

A. Wallar is with the School of Computer Science, University of St Andrews, Fife KY16 9AJ, Scotland, UK. E. Plaku is with the Dept. of Electrical Engineering and Computer Science, Catholic University of America, Washington DC 20064 USA. D. Sofge is with the Naval Research Laboratory, Washington, DC 20375 USA.

Toward increasing the autonomy of UAVs, this paper proposes a motion-planning approach, termed PARCOV (Planner for Autonomous Risk-sensitive Coverage), for persistent coverage of risk-sensitive areas using multiple cooperative quadcopters. Persistent coverage is important in many UAV applications in order to ensure that no region goes too long without being surveyed. Moreover, in order to utilize the information gathered during surveillance, it becomes important for the team of quadcopters to maintain high sensor data quality. Accounting for risk is also crucial in many aerial missions. In surveillance, for example, risk could represent the likelihood of the quadcopters being detected by a possibly hostile agent. As another example, when monitoring a brushfire, risk could represent the likelihood of the quadcopters being damaged. PARCOV is general and can accommodate various risk metrics that decrease in value as the altitude increases.

Achieving persistent coverage while minimizing risk and maintaining high sensor data quality presents significant challenges. In particular, scalability is desired to ensure that the approach can be applied to an increasing number of quadcopters. As risk and sensor data quality often increase when flying closer to the ground, it becomes challenging to find an optimal altitude. Related work has focused on various aspects of planning for UAVs [1], [2]. Ergezer and Leblebicioğlu [3] describe a path-planning algorithm that seeks to maximize the information collected from desired regions while avoiding forbidden regions. Evolutionary algorithms and particle-swarm optimizations have also been proposed for UAV path planning [4]–[9]. A closed-loop path optimization approach is developed by Kuhlman et al. [10] to obtain persistent area coverage by a single UAV while maximizing the information gained from information-rich areas. Communication-range constraints are taken into account in [11] so that UAVs are able to maintain communication while visiting desired regions and avoiding forbidden regions. Cheng, Keller, and Kumar [12] plan time-optimal trajectories for a single UAV providing sensor coverage of urban structures. Huynh, Enright, and Frazzoli [13] propose control policies for the persistent-patrol problem in order to reduce the expected time it takes to detect an incident. Cooperative control techniques are developed in [14] to reduce the likelihood of UAVs being detected by radars. Dynamic task distribution is proposed in [15]–[18] in order to enhance cooperative searching. Different techniques based on sequential decision processes, computer vision, and dynamic programming have been proposed from target tracking by a single or multiple UAVs [19]–[21]. Sydney, Paley, and Sofge [22] develop a physicomimetic approach which tracks targets by following the gradient of a surface defined based on mutual

information between the UAVs and the targets. Apker et al. [23] propose a bio-inspired approach which models the search space as pastures and the UAVs as grazing animals that seek to consume the available information.

PARCov offers several contributions. In particular, it utilizes simple interactions among UAVs to promote an emergent behavior that maximizes coverage while reducing risk and maintaining high sensor data quality. Fig. 1 provides some illustrations of how the quadcopters cover the area. Scalability is achieved by separating motion planning in xy from determining the optimal altitude for each quadcopter. PARCov maintains an uncertainty measure designed to keep track of the regions that have been surveyed and the times they were last surveyed. In this way, the uncertainty measure of a region increases the longer the region remains out of the sensor footprint of the quadcopters. Moreover, PARCov combines the uncertainty measure with the risk into a cost function. In order to reduce the risk, the cost function increases as risk increases. To promote coverage of regions that have not been sensed in a long time, the cost function decreases as uncertainty increases. As a result, PARCov seeks to move each quadcopter toward a low-cost region in its vicinity. Such motions have the desired effect of reducing both the risk and the uncertainty which promotes persistent coverage of risk-sensitive areas. Moreover, PARCov uses a nonlinear optimization formulation to determine the optimal altitude of each quadcopter in order to maximize the sensor data quality while minimizing risk. Experimental validation is provided both in simulation and with real AscTec Pelican quadcopters.

A preliminary version of PARCov appeared as a symposium proceeding [24]. This paper offers several improvements over the preliminary version such as the introduction of the uncertainty measure and its combination with the risk into the cost function to effectively guide the quadcopters toward low-cost regions. Moreover, this paper provides extended experimental evaluation using different risk models and an increasing number of UAVs. While the preliminary version was limited to simulation, this paper incorporates PID controllers with PARCov and is applied to real AscTec Pelican quadcopters.

II. PROBLEM FORMULATION

The problem considered in this paper is to enable a team of quadcopters to persistently cover a given area \mathcal{A} while reducing risk and maintaining high sensor data quality. The models used in this paper for sensor coverage, sensor data quality, and risk are described below.

a) Area coverage and persistency: Let Quadcopters = $\{q_1, \dots, q_n\}$ denote the set of quadcopters. Each quadcopter q_i has a sensor mounted at a fixed angle ϕ . The area on the xy plane sensed by q_i at time t is denoted by $\text{SensedArea}_{q_i}(t)$. For the spotlight sensors used in the experiments, $\text{SensedArea}_{q_i}(t)$ corresponds to an ellipse defined by the position and orientation of q_i , the mounting angle ϕ , and the conic aperture α . Fig. 1 provides an illustration. The overall area sensed by the team is then defined as

$$\text{SensedArea}(t) = \bigcup_{q_i \in \text{Quadcopters}} \text{SensedArea}_{q_i}(t).$$

As the number of quadcopters might not be sufficiently large to cover the entire area at a time t , i.e., $\text{SensedArea}(t) \ll \text{area}(\mathcal{A})$, PARCov seeks to ensure that no region of \mathcal{A} remains unsensed for a long period of time. In this way, rather than remain still, the quadcopters will fly from one region to the next in order to persistently cover \mathcal{A} .

b) Sensor data quality: PARCov also seeks to maintain high sensor data quality, which is needed in many surveillance and target-tracking applications. For modeling, it is assumed that there is an optimal altitude, denoted by μ_{sq} , which yields the highest sensor data quality. The optimal altitude, which is a user-defined argument, depends on the particular task and can vary from situation to situation. As an example, the optimal altitude when tracking a person can be smaller than when tracking a tank since the tank is larger and faster. Furthermore, it is assumed that there is an exponential decrease in the sensor data quality as the deviation from μ_{sq} increases. More specifically, the sensor data quality is modeled as a distribution with mean μ_{sq} and standard deviation σ_{sq} , i.e.,

$$SQ(z) = \exp\left(-\frac{z}{\cos \phi} - \mu_{sq}\right)^2 / (2\sigma_{sq}^2),$$

where z is the altitude of the quadcopter and ϕ is the angle at which the sensor is mounted.

c) Risk model: Another objective of PARCov is to minimize the risk associated with the task assigned to the team of the quadcopters. To provide generality, the risk at the ground level is defined first via a function $R_0 : \mathbb{R}^2 \rightarrow [0, 1]$. As the altitude increases, it is assumed that the risk decreases exponentially. In this way, the risk function $R : \mathbb{R}^3 \rightarrow [0, 1]$, where $R(x, y, z)$ represents the risk at the location $(x, y, z) \in \mathbb{R}^3$, is defined as

$$R(x, y, z) = R_0(x, y) \cdot \exp\left(-\frac{z^2}{K \cdot R_0(x, y)^2}\right),$$

where K is a scaling constant. As a result, quadcopters need to fly at higher altitudes over areas with high ground-level risk in order to reduce the likelihood of being detected.

Note that PARCov can work with any risk model. The ground-risk models used in the experiments are generated by using the diamond-square algorithm [25], which constructs random heatmaps that have values from zero to one. Random heatmaps have also been used in [26] to represent random risk for path planning problems. Figs. 1 and 3 provide several illustrations of different heatmaps generated by the diamond-square algorithm. More details about the risk models used in the experiments are provided in Section IV.

d) Problem statement: The problem considered in this paper can now be stated as follows: Given an area \mathcal{A} to be surveyed, risk and sensor data quality models, and an initial placement of the quadcopters, move the quadcopters so that they persistently cover \mathcal{A} while reducing the risk and maintaining high sensor data quality.

III. METHOD

Planning in PARCov occurs in two stages. During the first stage, PARCov plans motions of the quadcopters in xy to promote persistent area coverage. During the second stage,

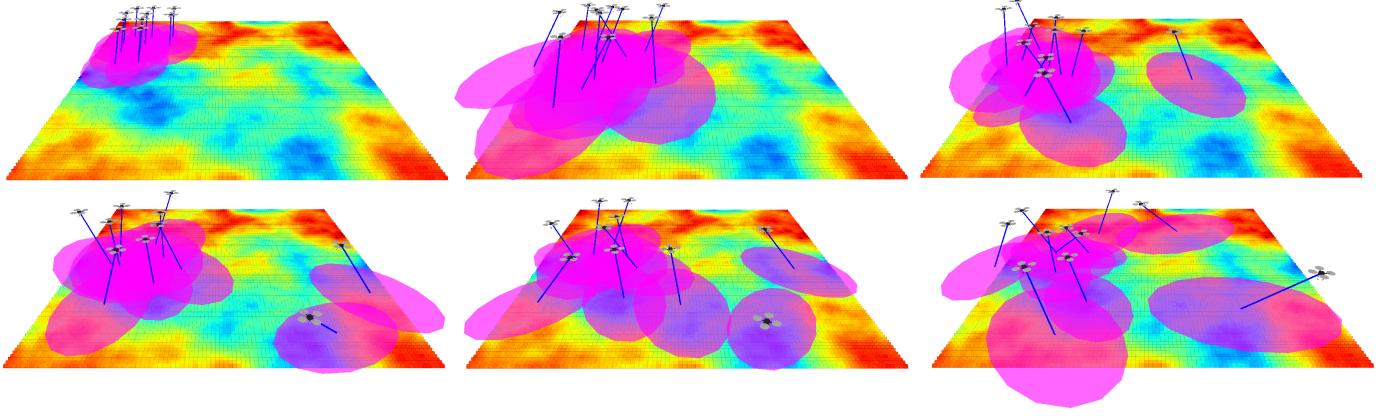


Fig. 1. Snapshots of PARCOV at different iterations showing how the quadcopters cover the designated area. The risk model is shown as a heatmap with red indicating high risk and blue indicating low risk. Figures may be best viewed in color and on screen.

PARCov adjusts the altitude of the quadcopters to minimize the risk while maximizing sensor data quality. Pseudocode is given in Alg. 1. Descriptions of the main steps follow.

Algorithm 1 Pseudocode for PARCov

input:

- area to be surveyed \mathcal{A}
- Quadcopters $= \{q_1, \dots, q_n\}$
- sensor model where $\text{SensedArea}(q_i) \subseteq \mathcal{A}$
- sensor data quality model $SQ : \mathbb{R} \rightarrow [0, 1]$
- ground-level risk $R_0 : \mathbb{R}^2 \rightarrow [0, 1]$
- risk model $R : \mathbb{R}^3 \rightarrow [0, 1]$
- COMBINE : $[0, 1] \times \mathbb{R}^+ \times \mathbb{N} \rightarrow \mathbb{R}^+$ to combine ground-level risk R_0 with uncertainty measure U at time $t \in \mathbb{N}$

output:

- new positions and orientations of each quadcopter

```

1:  $U_{\text{grid}} \leftarrow \text{INITUNCERTAINTYGRID}(); t \leftarrow 1$ 
2: while FINISHED() = false do
3:   for  $q \in \text{Quadcopters}$  do
4:      $(v', \beta') \leftarrow \text{DIRANDANGLE}_{(R_0, U_{\text{grid}}, \text{COMBINE})}(q, t)$ 
5:      $\begin{bmatrix} x' \\ y' \end{bmatrix} \leftarrow \begin{bmatrix} q.x \\ q.y \end{bmatrix} + \text{step} \cdot \frac{v'}{\|v'\|}$ 
6:      $z' \leftarrow \text{DETERMINEALTITUDE}_{(R, SQ)}(x', y')$ 
7:      $\text{SETTARGETPOSANDORIENTATION}(q, x', y', z', \beta')$ 
8:      $\text{CONTROLLER}(q, x', y', z', \beta')$ 
9:      $\text{UPDATEUNCERTAINTYGRID}(U_{\text{grid}}, q, t)$ 
10:     $t \leftarrow t + 1$ 

```

A. Planning motions in 2D

In order to promote area coverage and persistency while reducing the risk, PARCov relies on a cost function $\text{COST} : \mathbb{R}^2 \times \mathbb{N} \rightarrow \mathbb{R}^+$ to determine the xy motions of each quadcopter, where $\text{COST}(x, y, t)$ denotes the cost associated with the point (x, y) at time t (expressed as an iteration count). Note that the cost function changes dynamically based on the movements of the quadcopters. In other words, at time t , the current cost values are known but not the cost values at later times. The cost function is constructed by combining the ground-level

risk $R_0(x, y)$ with an uncertainty measure $U : \mathbb{R}^2 \times \mathbb{N} \rightarrow \mathbb{R}^+$, where $U(x, y, t)$ indicates the time that has elapsed since (x, y) was last sensed by a quadcopter. In this way, the uncertainty associated with (x, y) increases as (x, y) goes longer and longer without being sensed. This paper allows for any combination of R_0 and U . Generally,

$$\text{COST}(x, y, t) \stackrel{\text{def}}{=} \text{COMBINE}(R_0(x, y), U(x, y, t), t)$$

where $\text{COMBINE} : [0, 1] \times \mathbb{R}^+ \times \mathbb{N} \rightarrow \mathbb{R}^+$ is a user-defined function. For the experiments in this paper, the cost function is defined as a linear combination of the ground-level risk R_0 and the uncertainty measure U , i.e.,

$$\text{COST}(x, y, t) = \alpha R_0(x, y) + t - U(x, y, t),$$

where α is a scaling constant (set to 100 in the experiments). In this way, in order to reduce the detection risk, the cost function increases as the risk increases. In order to promote coverage of areas that have not been sensed in a long time, the cost function decreases as uncertainty increases.

As an implementation note, the uncertainty measure is maintained by first imposing a grid U_{grid} over the xy bounding box of the area \mathcal{A} being surveyed. The uncertainty grid is used to keep track of the regions in \mathcal{A} that have been surveyed and the times that they were last surveyed. More specifically, $U_{\text{grid}}(c)$ returns the time, as an iteration count, when the grid cell c was last sensed by a quadcopter. The uncertainty grid is initialized by setting each $U_{\text{grid}}(c)$ to zero (Alg. 1:1). Whenever a grid cell c is sensed by some quadcopter, i.e., $c \subseteq \text{SensedArea}_{q_i}(t)$, the current iteration count t is stored in c (Alg. 1:9). The uncertainty measure $U(x, y, t)$, which keeps track of the time that has elapsed since (x, y) was last sensed, is then computed as $t - U_{\text{grid}}(c)$ where t is the current iteration count and c is the grid cell that contains (x, y) . Note that c can be computed in constant time from (x, y) .

Illustrations of the uncertainty measure and the cost function at various time instances are provided in Fig. 2. As the algorithm progresses the uncertainty of the area changes and so does the associated cost. These measures are vital in order to effectively plan the motions of the quadcopters.

In order to reduce the uncertainty and detection risk, PARCOV seeks to move each quadcopter along the direction

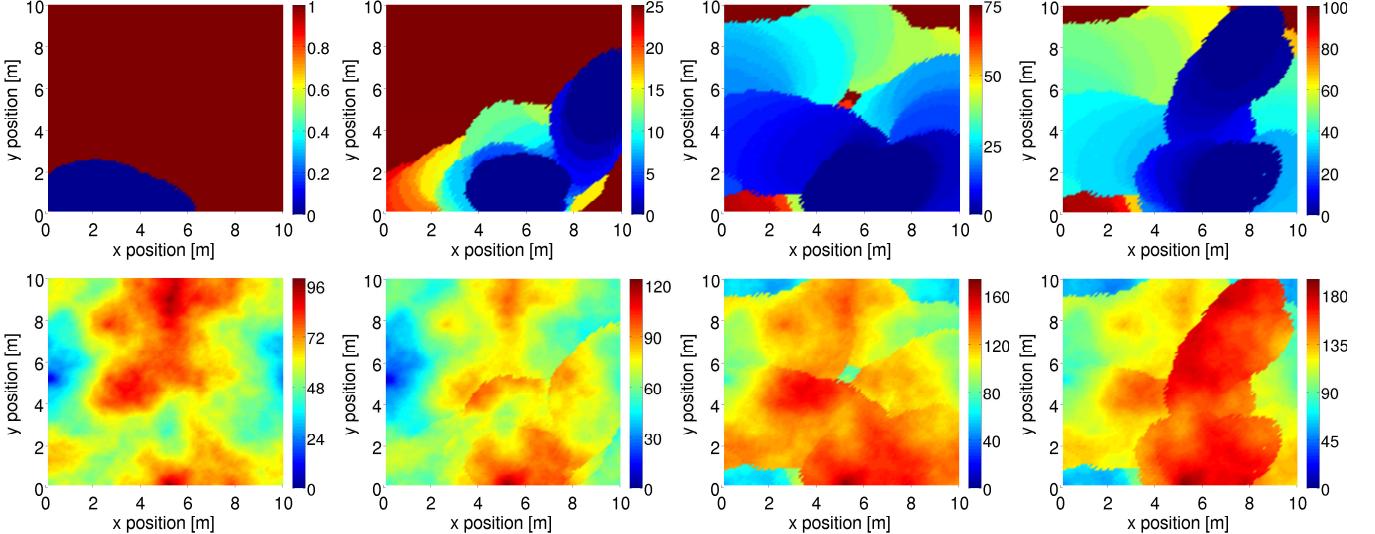


Fig. 2. Instances of the uncertainty measure U (top row) and cost function (bottom row) after 1, 25, 75, 100 iterations of PARCOV. For the uncertainty measure, the color in the spectrum represents the time that has elapsed, as an iteration count, since the point (x, y) was last visited. As blue represents recently visited areas, the uncertainty increases from blue to red, as indicated in the adjacent color spectrum scale. Recall that the cost function (bottom row) is computed by combining the uncertainty measure with the ground-level risk (which is shown in Fig. 1). Figures may be best viewed in color and on screen.

that would minimize the cost function. Pseudocode is given in Alg. 2. PARCov takes advantage of locality by considering moving the quadcopter q in the vicinity of the area currently sensed by q . More specifically, PARCov first generates a set of candidate orientations $\{\beta_1, \dots, \beta_\ell\}$ at random, i.e., $\beta_i = \text{RAND}(0, 2\pi)$ (Alg. 2:2). Let $\text{SensedArea}(q, \beta)$ denote the area that would be sensed by q when keeping the same position but setting the orientation to β . As an illustration, the sensed area for a spotlight sensor model would be an ellipse defined with respect to the parameter $\omega \in [0, 2\pi]$ as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} z \cdot \tan(\phi - \alpha) + A_M \cdot (1 + \cos \omega) \\ A_m \cdot \sin \omega \end{bmatrix},$$

where (x, y, z) is the position of the quadcopter, ϕ is the angle at which the sensor is mounted, α is the conic aperture, $A_M = z \cdot \tan(\phi + \alpha) - z \cdot \tan \phi$ and $A_m = \frac{z \cdot \tan \alpha}{\cos \phi}$ are the major and minor axis, respectively.

To take advantage of locality, PARCov considers $\xi(q, \beta)$ which represents $\text{SensedArea}(q, \beta)$ enlarged by some $\epsilon > 0$ (Alg. 2:4). To determine where to move q so that it can lower the cost, PARCov generates several segments along the perimeter of $\xi(q, \beta)$. More specifically, let p_i denote the point on the perimeter of $\xi(q, \beta)$ corresponding to the angle parameter $i \cdot \zeta$, where ζ defines the sampling value (set to 10° in the experiments). The i -th segment is then obtained by connecting p_i to p_{i+1} . A cost is computed for each segment s , denoted by $\text{SegmentCost}(s)$, as the average of the costs of a number of equally-spaced points along s . PARCov will then move the quadcopter q toward the segment with the minimum cost, i.e.,

$$s = \arg \min_{s' \in \text{AllSegments}} \text{SegmentCost}(s')$$

where $\text{AllSegments} = \bigcup_{\beta \in \text{orientations}} \text{segments}(\xi(q, \beta))$.

The new orientation of q is set to $\beta \in \{\beta_1, \dots, \beta_\ell\}$ from which $\xi(q, \beta)$ that contains the segment s with minimum cost

Algorithm 2 DIRANDANGLE($R_0, U_{\text{grid}}, \text{COMBINE}$) (q, t)

```

define COST( $x, y, t$ ) = COMBINE( $R_0(x, y), U(x, y, t), t$ )
1: minAvgCost  $\leftarrow \infty$ ;  $v' \leftarrow (0, 0)$ ;  $\beta' \leftarrow 0$ 
2: orientations  $\leftarrow \text{GETRANDOMSAMPLES}(0, 2\pi)$ 
3: for  $\beta \in \text{orientations}$  do
4:    $\xi \leftarrow \text{ENLARGE}(\text{SensedArea}(q.x, q.y, q.z, \beta), \epsilon)$ 
5:   segments  $\leftarrow \text{GETSEGMENTS}(\xi)$ 
6:   for  $s \in \text{segments}$  do
7:     points  $\leftarrow \text{GETPOINTS}(s)$ 
8:     avgCost  $\leftarrow \frac{1}{|\text{points}|} \sum_{(x,y) \in \text{points}} \text{COST}(x, y, t)$ 
9:     if minAvgCost  $<$  avgCost then
10:      minAvgCost  $\leftarrow$  avgCost
11:       $\beta' \leftarrow \beta$ 
12:      sumWeight  $\leftarrow \sum_{(x,y) \in \text{points}} 1/\text{COST}(x, y, t)$ 
13:       $v' \leftarrow \begin{bmatrix} -q.x \\ -q.y \end{bmatrix} + \sum_{(x,y) \in \text{points}} \frac{1/\text{COST}(x, y, t)}{\text{sumWeight}} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$ 
14: return  $(v', \beta')$ 

```

was derived (Alg. 2:11). The new direction is set by taking a weighted average of the points along s (Alg. 2:12–13), i.e.,

$$\begin{bmatrix} -q.x \\ -q.y \end{bmatrix} + \sum_{(x,y) \in \text{points}(s)} \frac{1/\text{COST}(x, y, t)}{w} \cdot \begin{bmatrix} x \\ y \end{bmatrix},$$

where $w = \sum_{(x',y') \in \text{points}(s)} 1/\text{COST}(x', y', t)$. In this way, points in s associated with lower costs exert a higher influence when determining the new direction. Finally, the xy position is set by taking a small step along the new direction (Alg. 1:5).

Such planning has desirable emergent properties for the team. By sharing the same cost function quadcopters will act cooperatively to cover the unvisited areas without having to explicitly coordinate with one another. As each quadcopter moves towards a low-cost area in its vicinity, it becomes less likely for the quadcopters to cluster together. In fact, suppose

that several quadcopters are moving towards the same segment s . As soon as s is sensed by a quadcopter, the uncertainty associated with s becomes zero so its cost increases. As a result, the other quadcopters will move away from s toward other segments that have lower costs.

Moreover, quadcopters have the flexibility to leave and join the team at any time. Since PARCov does not explicitly divide the area among the quadcopters participating in the task, if a quadcopter leaves the area, it would simply no longer update the uncertainty measure. The rest of the team would have no knowledge that it left so they would continue to update the uncertainty measure and move toward areas with low cost. Similarly, a new quadcopter can join the team at any time by accessing the cost function and updating the uncertainty measure. The flexibility of leaving and joining the team as needed is particularly important for missions that combine persistent coverage and target tracking. When the team covering the area detects moving targets, a number of quadcopters from the team can be deployed to track them while the rest continue to survey the area.

B. Determining the altitude

After computing the xy position, the new altitude at which a quadcopter should fly is determined by optimizing an objective function that maximizes the sensor data quality and minimizes the risk, i.e.,

$$J(x, y, z) = SQ(z) - R(x, y, z).$$

In this way, the optimal altitude corresponds to the z value that maximizes J for a given x, y , i.e.,

$$\text{DETERMINEALTITUDE}(x, y) = \arg \max_{z \in [z_{\min}, z_{\max}]} J(x, y, z).$$

Nonlinear optimization solvers, such as SciPy, can then be used to numerically compute the altitude that maximizes J .

C. Controller and updates to the uncertainty grid

A PID controller is employed to steer each quadcopter toward the new position and orientation (Alg. 1:7–8). The PID controller used in simulation is the same as the one we use to steer the real AscTec Pelican quadcopters toward desired targets. After each motion step, the uncertainty grid U_{grid} is updated accordingly to account for the newly sensed area (Alg. 1:7). Since the quadcopters share U_{grid} , the team will dynamically react to new information. In particular, when a quadcopter decreases its altitude there will be more uncovered space around it so other quadcopters will move in to cover these areas. These dynamic adjustments, as shown next, make it possible to efficiently cover the area while maintaining high sensor data quality and reducing the risk.

IV. EXPERIMENTS AND RESULTS

The performance of PARCov was tested in simulation using an increasing number of quadcopters and various risk maps. Experiments were also conducted with real AscTec Pelicans at the Laboratory for Autonomous Systems Research at the Naval Research Laboratory.

A. Scenes

Experiments were conducted with three different scene sizes: small ($10m \times 10m$), medium ($15m \times 15m$), and large ($20m \times 20m$). In each case, the quadcopters were required to fly between $z_{\min} = 0.6m$ and $z_{\max} = 4m$. For each scene size, three different heatmaps, representing different ground-level risks, were generated by using the diamond-square algorithm [25]. Thus, a scene is defined by its size and the corresponding ground-risk heatmap. For clarity, a scene is referred to as X scene with risk model i where $X \in \{\text{small, medium, large}\}$ and $i \in \{1, 2, 3\}$. Fig. 3 shows all the nine scenes used in the experiments.

B. Performance criteria

The performance of PARCov is measured according to several criteria: current coverage (C_1), cumulative coverage (C_2), 90%-persistent coverage (C_3), sensor quality (C_4), risk (C_5), and average wait time (C_6), as described below.

Current coverage (C_1) measures the mean percentage of the area covered at each iteration of PARCov, i.e.,

$$C_1(t) = 100 \frac{1}{t} \sum_{i=1}^t \frac{\text{SensedArea}(i)}{\text{area}(\mathcal{A})},$$

where t denotes the current iteration count and, as defined in Section II, $\text{SensedArea}(i) = \text{SensedArea}_{q_1}(i) \cup \dots \cup \text{SensedArea}_{q_n}(i)$ denotes the area sensed by the quadcopters at the i -th iteration. In the experiments, after updating the uncertainty grid U_{grid} (Alg. 1:9) for all the quadcopters, $\text{SensedArea}(i)$ is computed by adding the areas of the grid cells in U_{grid} that are sensed during the i -th iteration. Current coverage (C_1) provides a measure of how well the quadcopters spread out to cover the area at any given instance.

Cumulative coverage (C_2) measures the percentage of the area covered over all the iterations of PARCov, i.e.,

$$C_2(t) = 100 \frac{1}{\text{area}(\mathcal{A})} \bigcup_{i=1}^t \text{SensedArea}(i).$$

It is expected, as the iteration count t increases, PARCov will cover the entire area. In the experiments, $C_2(t)$ is computed by adding up the areas of all the grid cells in U_{grid} that have an iteration count greater than zero.

The third criteria, referred to as 90%-persistent coverage (C_3), is designed to measure coverage persistency. In particular, it measures the average number of iterations to repeatedly reach 90% cumulative coverage. More specifically, we keep a counter to track the number of iterations to reach 90% cumulative coverage. After reaching it, the value of the counter is pushed onto a list and the cumulative coverage and the counter are set back to zero. At time t , the measure $C_3(t)$ is then computed as the average value in the list of counters. As an implementation note, the cumulative coverage needed for the purposes of computing $C_3(t)$ is obtained as

$$100 \frac{1}{\text{area}(\mathcal{A})} \bigcup_{i=1+t_{\text{last90}}}^t \text{SensedArea}(i),$$

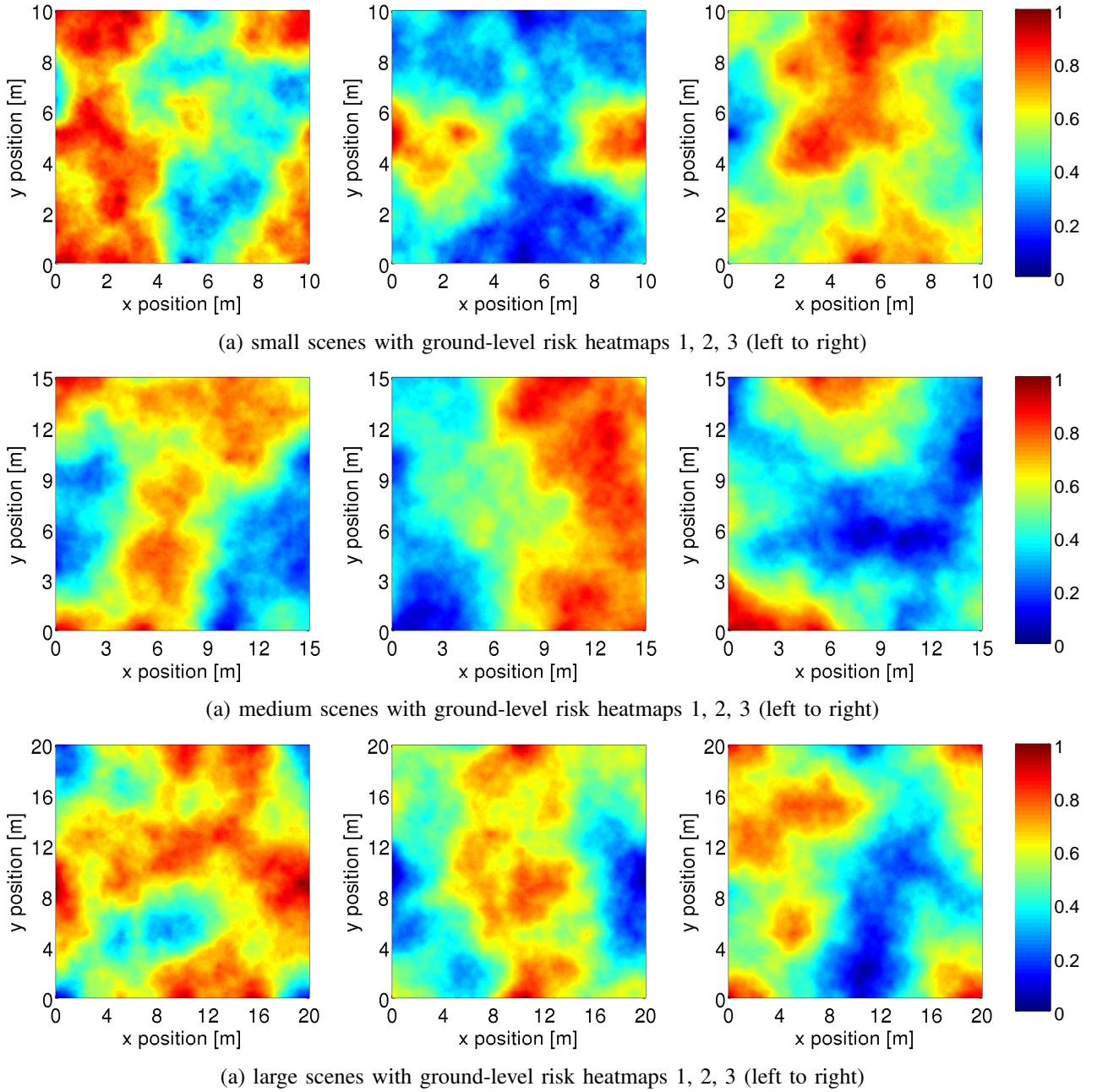


Fig. 3. The various ground-level risk heatmaps used in the experiments where the color indicates the risk value as shown in the color spectrum.

where $t_{\text{last}90}$ is the iteration count when 90% cumulative coverage was reached. This is computed by adding up the areas of all the grid cells in U_{grid} that have an iteration count greater than $t_{\text{last}90}$. In this way, the 90%-persistent coverage criterion measures how long it takes the quadcopters to cover 90% of the space starting from many different configurations.

Sensor quality (C_4) is measured by considering the sensor-quality model SQ (Section II) and the distances from the quadcopters to the grid cells that they cover. More specifically, consider a grid cell $c \in U_{\text{grid}}$. The sensor quality associated with c at iteration i is defined by taking the maximum sensor

quality among the quadcopters that sense c , i.e.,

$$SQ(c, i) = \max_{\substack{q \in \text{Quadcopters} \text{ and} \\ c \subseteq \text{SensedArea}_q(i)}} SQ(\text{dist}(c, q)),$$

where $\text{dist}(c, q)$ denotes the distance from the center of the cell c to the $(q.x, q.y, q.z)$ position of the quadcopter q . The sensor quality associated with U_{grid} at iteration count i is defined as the average of the sensor qualities associated with the grid cells sensed in the i -th iteration, i.e.,

$$SQ(U_{\text{grid}}, i) = \sum_{c \in \text{SensedCells}(i)} \frac{SQ(c, i)}{|\text{SensedCells}(i)|},$$

where $\text{SensedCells}(i) = \{c : c \in \text{cells}(U_{\text{grid}}) \text{ and } c \subseteq \text{SensedArea}(i)\}$. Then, the overall sensor-quality measure

(C_4) is defined as the average sensor-quality of U_{grid} over the iterations $1, \dots, t$ scaled to a percentage, i.e.,

$$C_4(t) = 100 \frac{1}{t} \sum_{i=1}^t SQ(U_{\text{grid}}, i).$$

The risk (C_5) measures the average detection risk over all the quadcopters and over the iterations $1, \dots, t$ scaled to a percentage, i.e.,

$$C_5(t) = 100 \frac{1}{t} \sum_{i=1}^t \sum_{q \in \text{Quadcopters}} \frac{R(q.x, q.y, q.z)}{|\text{Quadcopters}|}.$$

The average wait time (C_6) is used to show that no part of the area being surveyed remains unsensed for a long time. More specifically, at the end of each iteration i (Alg. 1:10), the average wait time for a grid cell is computed as

$$\text{WaitTime}(i) = \sum_{c \in \text{cells}(U_{\text{grid}})} \frac{i - U_{\text{grid}}(c)}{|\text{cells}(U_{\text{grid}})|}.$$

These wait times are stored in a list and $C_6(t)$ is computed by taking their average, i.e.,

$$C_6(t) = \frac{1}{t} \sum_{i=1}^t \text{WaitTime}(i).$$

C. Results

Before presenting quantitative results, we provide some qualitative illustrations to show PARCov in action. Fig. 1 shows how the quadcopters cover the designated area. Using the cost function obtained by combining the uncertainty measure with the risk, the quadcopters start moving toward areas associated with low cost.

Another illustration of PARCov in action is provided in Fig. 4, which shows trajectories taken by two quadcopters. Note how the quadcopters increase their altitude when surveying areas designated as high risk and reduce their altitude when going over low-risk areas. As discussed in Section III-B, PARCov uses a nonlinear optimization process to reduce detection risk while maintaining high sensor data quality.

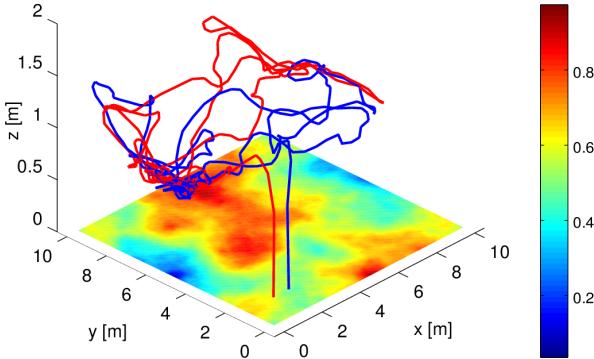


Fig. 4. Trajectories taken by two quadcopters shown in blue and red, respectively. The initial risk R_0 is shown as a heatmap.

Fig. 5 shows the runtime per iteration, where an iteration ends when the new position and orientation is determined for each quadcopter (Alg. 1:3–9). The results show that

the runtime per iteration increases linearly with the number of quadcopters. These results, as all the experiments, were obtained on an Intel Core i7 machine (CPU: 2.40GHz, RAM: 8GB) using Ubuntu 14.04. Code was written in Python 2.7. ROS rviz was used for visualization.

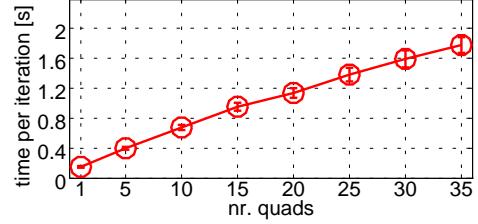


Fig. 5. Runtime per iteration. Bars indicate standard deviation. Results are shown for the large-size scene with risk model 3.

The rest of the section presents results on the performance of PARCov according to various criteria ($C_1 - C_6$). Results are also presented that show the performance as a function of number of iterations, number of quadcopters, scene size, and control noise. The section concludes with results from experiments with real AscTec Pelican quadcopters.

1) *Results on various performance criteria:* Fig. 6 shows the performance of PARCov in terms of the current coverage (C_1), cumulative coverage (C_2), sensor quality (C_4), and risk (C_5). The results indicate that PARCov effectively guides the quadcopters to quickly cover the designated area. In fact, the cumulative coverage (C_2) increases rapidly with the number of iterations. Even a single quadcopter is able to cover more than 90% of the space in a few hundred iterations. As the number of quadcopters increases, fewer and fewer iterations are needed to achieve high cumulative coverage. The current coverage (C_1) also increases with the number of quadcopters. The initial increase with the number of iterations results from the spreading of the quadcopters from the initial configuration which has them all start near each other in a corner of the scene. The figure also shows that PARCov enables the quadcopters to maintain high sensor data quality (C_4) while significantly reducing the risk (C_5).

Fig. 8 shows the performance of PARCov in terms of the 90%-persistent coverage criterion (C_3). The results show that the average number of iterations to reach 90% coverage decreases rapidly as more and more quadcopters work together to survey the area. To give some context for this graph, when considering the large scene ($20m \times 20m$) and 30 quadcopters, around 45 iterations are required to reach 90% coverage. Thus, when running the algorithm for 500 iterations, the quadcopters reach 90% coverage more than 10 times over this $400m^2$ area. This quick convergence combined with the data from Fig. 6 shows that PARCov is able to maintain persistent coverage of the area while maintaining low risk and high sensor quality.

Fig. 9 shows the average wait time (C_6) as a function of the number of quadcopters for different scene sizes. As the number of quadcopters increases, the average wait time decreases since the quadcopters spread through the space cooperatively and therefore cover the space more quickly. These results provide further evidence about the ability of PARCov to maintain persistent coverage.

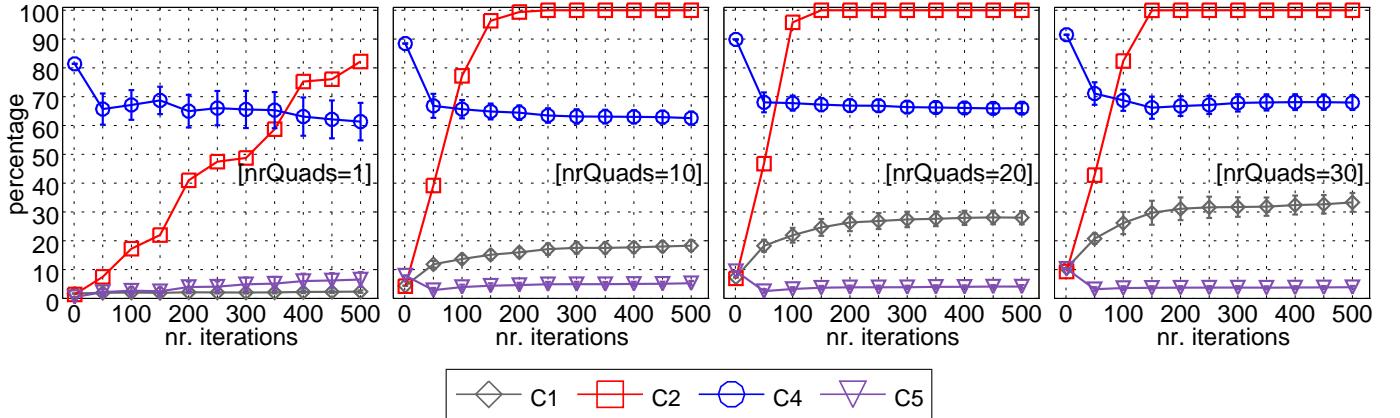


Fig. 6. Performance criteria: current coverage $C_1(t)$, cumulative coverage $C_2(t)$, sensor quality $C_4(t)$, and risk $C_5(t)$ with respect to the number of iterations t . Results are shown for the first large-size scene and different numbers of quadcopters.

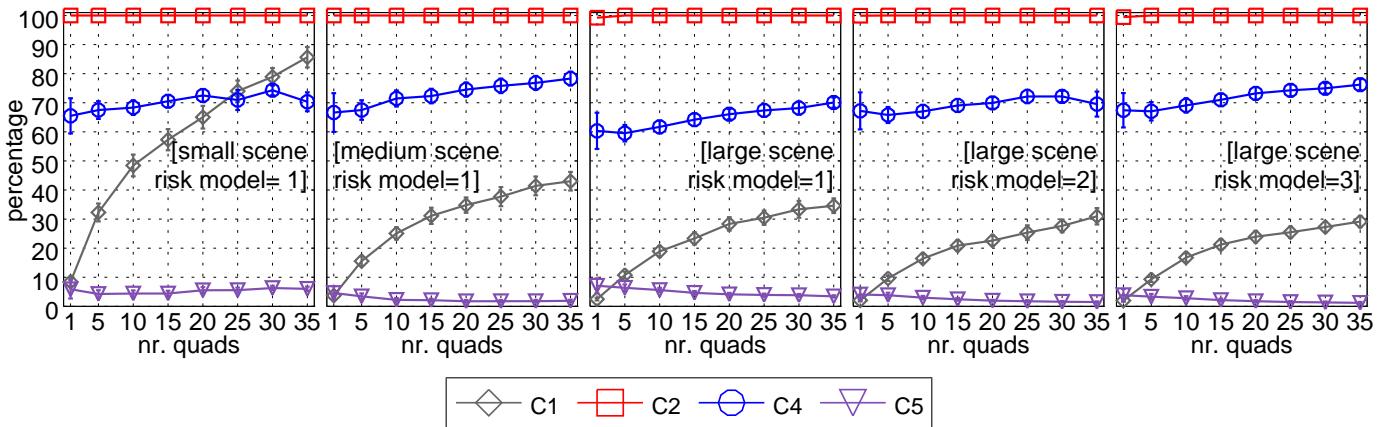


Fig. 7. Performance criteria: current coverage $C_1(500)$, cumulative coverage $C_2(500)$, sensor quality $C_4(500)$, and risk $C_5(500)$ when varying the scene size, the risk model, and the number of quadcopters.

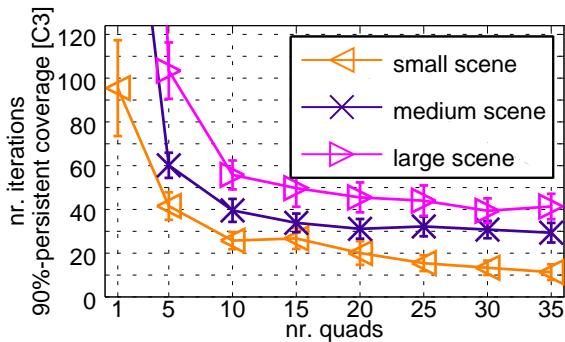


Fig. 8. 90% persistent coverage, i.e., $C_3(500)$, when varying the number of quadcopters and scene size. Bars indicate one standard deviation.

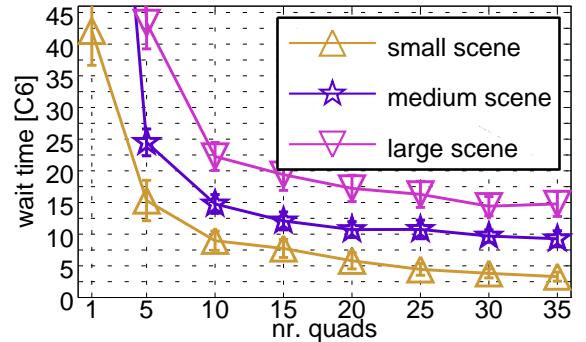


Fig. 9. Mean wait time, i.e., $C_6(500)$, when varying the number of quadcopters and scene size. Bars indicate one standard deviation.

2) *Performance when varying scene size and risk model:* Fig. 7 shows the performance of PARCov when varying the scene size, risk model, and the number of quadcopters. The algorithm works well for a variety of scenes and ground-risk models. As shown, PARCov effectively surveys the area while maintaining high sensor quality and minimizing risk.

3) *Performance when varying control noise:* In order to determine whether PARCov would be able to withstand an acceptable amount of error in the controller, we have conducted

experiments to test the performance by changing the amount of noise that is present in the simulated proportional-integral-derivative (PID) controller. We did this by adding Gaussian noise to the control output of the PID controller. The noise was quantified by altering the standard deviation of the Gaussian function. For the experiments, the mean for the Gaussian was set to zero and the standard deviation was set from 0 to 0.6 meters with a step of 0.2 meters. This range of values seemed to be an accurate representation of the control noise present on

TABLE I
PERFORMANCE CRITERIA WHEN VARYING THE CONTROL NOISE

[medium scene, risk model = 1] control noise std (m) \rightarrow	nr. quads = 10				nr. quads = 20				nr. quads = 30			
	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6
$C_1(500)$ current coverage	22.3	21.8	22.0	21.7	31.3	31.4	31.1	31.3	37.7	37.2	36.9	37.1
$C_2(500)$ cumulative coverage	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$C_3(500)$ 90%-persistent coverage	38.6	39.4	38.9	37.9	32.4	33.8	33.5	32.0	29.4	29.8	29.6	31.8
$C_4(500)$ sensor quality	72.6	73.7	73.3	73.8	76.5	76.3	76.7	76.4	78.9	79.3	79.8	79.5
$C_5(500)$ risk	2.1	2.0	2.1	1.8	1.5	1.7	1.6	1.6	1.7	1.6	1.6	1.7
$C_6(500)$ wait time	14.1	15.3	14.4	14.7	11.7	11.8	11.3	11.1	9.7	10.0	10.2	10.4

the AscTec Pelicans we used for the practical experimentation.

Table I provides a summary of the results. Since PARCov is purely reactive, we can see that within an acceptable amount of noise PARCov is still able to perform well according to all the metrics we have used.

D. Physical Experiments

We tested PARCov at the Laboratory for Autonomous Systems Research located at the Naval Research Laboratory in Washington DC. We used two AscTec Pelican quadcopters operated in a $5.2m \times 5.2m$ area surrounded by 10 Vicon motion capture cameras, as shown in Fig. 10. The quadcopters were equipped with passive motion capture markers which were seen by the Vicon cameras. The Vicon cameras have LEDs which were used to illuminate the passive markers. The 3D positions and the orientations of the quadcopters were sent to a laptop which uses the Robotic Operating System (ROS) and a swarm middleware called ZeroMQ-ROS to compute the control commands and to send them to the quadcopters. ROS is a open-source middleware combined with software libraries and tools to build robotic applications. ZeroMQ-ROS is a middleware for controlling a swarm using a ROS multi-master architecture. The control commands were sent over WiFi through a socket to each of the quadcopters.

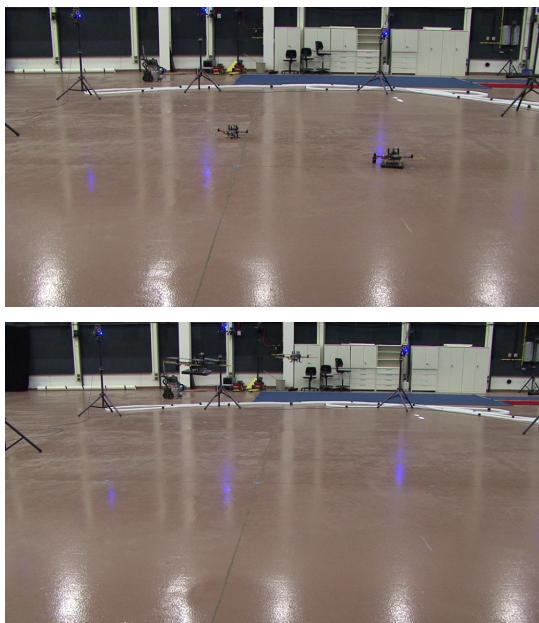


Fig. 10. Experimental setup with two AscTec Pelican quadcopters surrounded by 10 Vicon motion capture cameras

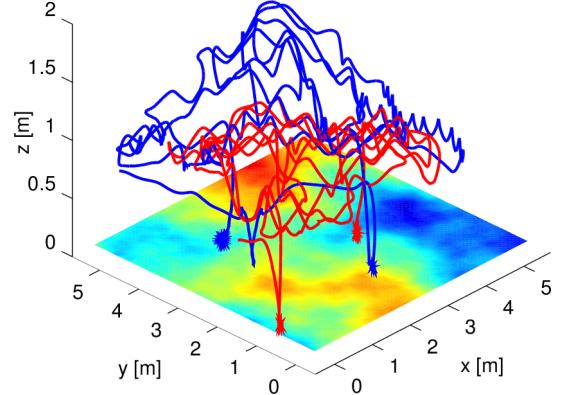


Fig. 11. Trajectories of the two AscTec Pelicans when surveying the area. The oscillations shown are caused by the control noise.

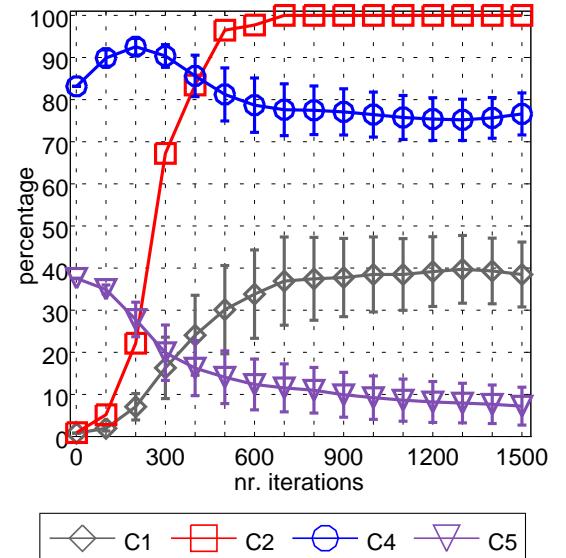


Fig. 12. Performance criteria from the experiments with the two AscTec Pelicans: current coverage $C_1(t)$, cumulative coverage $C_2(t)$, sensor quality $C_3(t)$, and risk $C_5(t)$ with respect to the number of iterations t .

Fig. 11 plots the trajectories followed by the AstTec Pelican quadcopters. Fig. 12 shows the performance metrics. The results indicate that PARCov enabled the AscTec Pelican quadcopters to effectively survey the entire area. Note that the quadcopters start at a high-risk area and remain there during takeoff, which takes some time. After that, PARCov guides the quadcopters toward areas the reduce the risk while maintaining high sensor data quality.

V. DISCUSSION

This paper developed PARCov, a motion-planning approach for persistent surveillance of risk-sensitive areas by a team of UAVs. PARCov relied on simple interactions among UAVs in order to promote an emergent behavior that maximized coverage while reducing the risk and maintaining high sensor data quality. Scalability was achieved by separating motion planning in xy from determining the optimal altitude for each quadcopter. Experiments in simulation and real AscTec Pelican quadcopters demonstrated the ability of the approach to provide persistent surveillance of risk-sensitive areas. In future work, we will enhance the approach to track moving targets. Another direction is to further improve the sensor quality by reducing the motion blur. We will also investigate extending the approach to teams conducting large-scale surveillance where power management becomes a critical issue in order to effectively survey areas of interest.

ACKNOWLEDGMENT

This work was performed at the Naval Research Laboratory and was funded by the US Department of Defense, Office of Naval Research under grant number N0001413WX21045, “Mobile Autonomous Teams for Navy Information Surveillance and Search (MANTISS).” The views, positions and conclusions expressed herein reflect only the authors opinions and expressly do not reflect those of the US Department of Defense, Office of Naval Research, or the Naval Research Laboratory. The work by E. Plaku is supported by NSF IIS-1449505. The authors thank Nitin Sydney for valuable insights during this work and Thomas Apker for helping with the physical experiments.

REFERENCES

- [1] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, and J. K. Hedrick, “An overview of emerging results in cooperative UAV control,” in *IEEE Conference on Decision and Control*, vol. 1, 2004, pp. 602–607.
- [2] C. Goerzen, Z. Kong, and B. Mettler, “A survey of motion planning algorithms from the perspective of autonomous uav guidance,” *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 65–100, 2010.
- [3] H. Ergezer and K. Leblebicioğlu, “3D path planning for multiple UAVs for maximum information collection,” *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 737–762, 2014.
- [4] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, “Evolutionary algorithm based offline/online path planner for uav navigation,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 33, no. 6, pp. 898–912, 2003.
- [5] Y. V. Pehlivanoglu, “A new vibrational genetic algorithm enhanced with a voronoi diagram for path planning of autonomous uav,” *Aerospace Science and Technology*, vol. 16, no. 1, pp. 47–55, 2012.
- [6] V. Roberge, M. Tarbouchi, and G. Labonté, “Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132–141, 2013.
- [7] M. Shammugavel, A. Tsourdos, B. White, and R. Źbikowski, “Cooperative path planning of multiple uavs using dubins paths with clothoid arcs,” *Control Engineering Practice*, vol. 18, no. 9, pp. 1084–1092, 2010.
- [8] Y. Fu, M. Ding, and C. Zhou, “Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for uav,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 42, no. 2, pp. 511–526, 2012.
- [9] P. Gaudiano, E. Bonabeau, and B. Shargel, “Evolving behaviors for a swarm of unmanned air vehicles,” in *IEEE Swarm Intelligence Symposium*, 2005, pp. 317–324.
- [10] M. J. Kuhlman, P. Svec, K. N. Kaipa, D. Sofge, and S. K. Gupta, “Physics-aware informative coverage planning for autonomous vehicles,” in *IEEE International Conference on Robotics and Automation*, 2014, in press.
- [11] R. W. Beard and T. W. McLain, “Multiple UAV cooperative search under collision avoidance and limited range communication constraints,” in *IEEE Conference on Decision and Control*, vol. 1, 2003, pp. 25–30.
- [12] P. Cheng, J. Keller, and V. Kumar, “Time-optimal UAV trajectory planning for 3d urban structure coverage,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2750–2757.
- [13] V. A. Huynh, J. J. Enright, and E. Frazzoli, “Persistent patrol with limited-range on-board sensors,” in *IEEE Conference on Decision and Control*, 2010, pp. 7661–7668.
- [14] P. R. Chandler, M. Pachter, and S. Rasmussen, “UAV cooperative control,” in *IEEE American Control Conference*, vol. 1, 2001, pp. 50–55.
- [15] I. Maza and A. Ollero, “Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms,” in *Distributed Autonomous Robotic Systems 6*. Springer, 2007, pp. 221–230.
- [16] Y. Jin, A. A. Minai, and M. M. Polycarpou, “Cooperative real-time search and task allocation in uav teams,” in *IEEE Conference on Decision and Control*, vol. 1, 2003, pp. 7–12.
- [17] T. Lemaire, R. Alami, and S. Lacroix, “A distributed tasks allocation scheme in multi-uav context,” in *IEEE International Conference on Robotics and Automation*, vol. 4, 2004, pp. 3622–3627.
- [18] D. Zhu, H. Huang, and S. X. Yang, “Dynamic task assignment and path planning of multi-aув system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace,” *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 504–514, 2013.
- [19] S. S. Baek, H. Kwon, J. A. Yoder, and D. Pack, “Optimal path planning of a target-following fixed-wing uav using sequential decision processes,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 2955–2962.
- [20] F. Lin, X. Dong, B. M. Chen, K.-Y. Lum, and T. H. Lee, “A robust real-time embedded vision system on an unmanned rotorcraft for ground target following,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1038–1049, 2012.
- [21] C. Teuliere, L. Eck, and E. Marchand, “Chasing a moving target from a flying uav,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4929–4934.
- [22] N. Sydney, D. A. Paley, and D. Sofge, “Physics-inspired robotic motion planning for cooperative bayesian target detection,” in *Robotics: Science and Systems*, 2014, workshop on Distributed Control and Estimation for Robotic Vehicle Networks.
- [23] T. Apker, S.-Y. Liu, D. Sofge, and J. K. Hendrick, “Application of grazing-inspired guidance laws to autonomous information gathering,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, in press.
- [24] A. Wallar, E. Plaku, and D. A. Sofge, “A planner for autonomous risk-sensitive coverage (parcov) by a team of unmanned aerial vehicles,” in *IEEE Symposium Series on Computational Intelligence*, 2014, in press.
- [25] A. Fournier, D. S. Fussell, and L. C. Carpenter, “Computer rendering of stochastic models,” *Commun. ACM*, vol. 25, no. 6, pp. 371–384, 1982.
- [26] L. Murphy and P. Newman, “Risky planning: Path planning over costmaps with a probabilistically bounded speed-accuracy tradeoff,” in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3727–3732.