

Occlusion-aware Multi-UAV Surveillance of Multiple Urban Areas

Michal Jakob, Eduard Semsch, Dušan Pavlíček and Michal Pěchouček

Agent Technology Center, Dept. of Cybernetics, FEE, Czech Technical University
Technická 2, 16627 Prague 6, Czech Republic
{jakob, semsch, pavlicek, pechoucek}@agents.felk.cvut.cz

ABSTRACT

We present an agent-based coordination and planning method for aerial surveillance of multiple urban areas using a group of fixed-wing unmanned aerial vehicles (UAVs). The method differs from the existing work by explicit consideration of sensor occlusions that can occur due to high buildings and other obstacles in the target area. The solution employs a decomposition of the problem in two subproblems: the problem of single-area surveillance and the problem of allocating UAVs to multiple areas. Three occlusion-aware methods for single-area surveillance are presented and compared. An algorithm for UAV allocation is presented and its optimality proved. The performance of all algorithms is evaluated empirically on a realistic simulation of aerial surveillance, built using the AGENTFLY framework, and is compared to theoretical estimates.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—*Coherence and coordination, multiagent systems*

General Terms

Algorithms, Performance

Keywords

autonomous aircrafts, UAV-based surveillance, UAV control, resource allocation, simulation, sensor occlusions

1. INTRODUCTION

There has been a growing interest in using unmanned aerial vehicles (UAVs) for information collection tasks, initially in military and later also in civil domains [2]. With the increasing numbers of UAVs, there is a growing need to enable the UAVs to perform their information collection missions autonomously without the need for direct human control, which is costly. Intelligent multi-agent techniques have been employed to address this problem [9][1].

Area surveillance is one of the most common information collection tasks, typically defined as a problem maintaining an up-to-date picture of the situation in a given area. Sometimes the task is termed *persistent* surveillance to highlight the fact that the area is to be monitored for a prolonged period of time.

In this paper, we address a particularly challenging variant of the problem – controlling a team of autonomous UAVs performing persistent surveillance of *geometrically complex*

environments such as those present in dense urban areas. In such environments, the field of view of UAV’s on-board sensors can get occluded in the presence of tall buildings and/or narrow streets (see Figure 1). This can result in areas left uncovered, which might be exploited by an adversary. The problem is likely to get more acute in the future as small, low-flying UAVs are going to be deployed.

In this paper, we address the problem by providing a multi-agent coordination mechanism that realistically models and explicitly eliminates the effect of occlusions. Although the problem of occlusions has been studied in other contexts [6, 12], *occlusion-aware* surveillance has not yet been considered in the field of autonomous UAV control. In addition to providing a solution to the problem of occlusions, we also show how multiple disjoint areas can be surveilled, which is novel too.

The paper proceeds as follows. In Section 2, we formally define the problem of UAV-based surveillance of occlusion-affected environments. In Section 3, we introduce our two-stage method; its two constituent components are described in two subsequent sections – single area surveillance in Section 4 and multiple-area surveillance in Section 5. Section 6 provides evaluation results, Section 7 reviews related work and, finally, Section 8 concludes.

2. MULTI-UAV INFORMATION AGE MINIMIZATION PROBLEM

We formally define multi-UAV area surveillance as a constraint optimization problem of finding a set of trajectories for a group of UAVs which, when followed, minimize the average age of information collected about a set of points of interest located within one or multiple disjoint areas.

2.1 Problem Domain

The environment is a set $\mathcal{A} = \{A_1, A_2, \dots, A_k\}$ of k rectangular areas A_i , each area A_i defined as a quadruple $A_i = \langle x_0, y_0, w, h \rangle$, where x_0, y_0 are the coordinates of its bottom left corner and w, h are its width and height. In each of the areas, a set of buildings may be present represented as quadrilateral prisms with their bases on the ground plane (defined by equation $z = 0$.) Furthermore, for each area A_i there is a finite set of *points of interest* $P_i = \{p_1, p_2, \dots, p_m\}$ located inside the respective area¹; these points should be seen as often as possible.

We control a fleet of fixed-wing UAVs $U = \{u_1, u_2, \dots, u_n\}$. The UAVs are modeled as point masses moving with a con-

¹Typically, the points of interest will be uniformly distributed within the target area

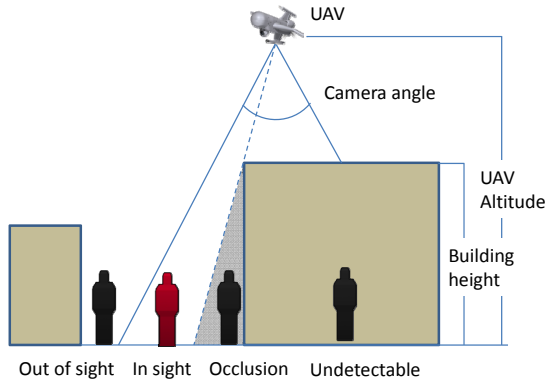


Figure 1: Occlusions in urban environment.

stant speed v and capable of turning with a minimum turning radius R (such a model is generally referred to as *Dubins vehicle* [7]). Each UAV carries a sensor of a conical field of view pointing down to the ground², with apex at the position of the UAV, and the field of view angle φ . The sensor is capable of observing ground points which are inside the sensor’s field of view and are *not occluded* by a building. The situation is illustrated in Figure 1. We define the function $\tau(p, t)$ as the last moment in time prior to time t when a point of interest p was seen by a UAV. If the point has not yet been seen, we set the value to 0.

2.2 Objective Function

For a time instance t and a point of interest p , we call the value $t - \tau(p, t)$ the *information age* of p at time t . The area surveillance problem is then to minimize the average information age of all points over a period of time, i.e., to minimize the expression

$$\frac{1}{t_1 - t_0} \frac{1}{|P|} \sum_{t=t_0}^{t_1} \sum_{i=1}^k \sum_{p \in P_i} (t - \tau(p, t)), \quad (1)$$

where discrete time model is assumed, and t_0 is the time at the beginning of the evaluation period (typically zero), t_1 at the end of the period, and $|P|$ is the number of points of interest. We term this objective function the *information age objective function* and the resulting optimization problem the *information age minimization problem*.

A solution of the problem is a set of flight trajectories for all the UAVs; the trajectories must respect the minimum turn radius of the UAVs.

3. TWO-STAGE MULTI-AREA SURVEILLANCE

The information age minimization problem is an instance of constraint optimization problems which are known to be generally intractable [16]. It is also known that the traveling repairman problem for Dubins vehicle, a special case of the single-area information age minimization, is NP-hard [13].

We therefore propose an approximate solution consisting of two stages:

1. Allocate UAVs to the areas. As a result, each UAV will have exactly one area assigned. Multiple UAVs can be assigned to one area.

²a tilting camera mount is assumed

2. For each area separately solve the single-area information age problem employing the allocated UAVs.

Note that we assume that the number of areas is not higher than the number of UAVs – an extension to the general case is possible but not presented in this paper. We describe the two stages in the following two sections, starting with the single-area surveillance algorithm.

4. SINGLE-AREA SURVEILLANCE

We now describe how a single rectangular area may be surveilled by one or more UAVs in a way that guarantees 100% coverage of all the points of interest in the area. We start with solving the problem for a single UAV; a straightforward extension to multiple UAVs is given in Section 4.5.

4.1 Single-UAV Surveillance

Assuming the structure of the surface to be a composition of quadrangular prisms, there always exists a finite set of points in the air, all lying at the same altitude, such that every point on the surface can be seen from at least one of the points in the set (for proof see e.g. [3]). We term any such set a *covering vantage point set*. We can then decompose the construction of an UAV’s flight trajectory into two steps:

1. Finding a covering vantage point set.
2. Finding the shortest trajectory travelling all the vantage points

Depending on the algorithm, the two steps can be performed in a serial order or combined into a single step.

The problem of finding a covering set of vantage points can be viewed as an instance of the *3D Art Gallery Problem*³, which has been shown to be NP-hard [8]. An approximation approach is thus frequently employed, consisting of discretizing the surveillance area and the area where the sensors can be placed, computing the visibility between the two sets, and finding a minimum set cover. While computing the minimum set cover is also a hard problem, efficient approximation algorithms exist.

We have developed and implemented three algorithms for single-area surveillance: (1) alternating algorithm, (2) spiral algorithm and (3) zig-zag algorithm. The first two algorithms separate the covering vantage point set generation from the construction of the flight trajectory; the zig-zag algorithm combines both steps together.

4.2 Alternating Algorithm

Alternating algorithm introduced in [13] is an approximation algorithm for solving the *Dubins vehicle travelling salesman problem* (DTSP) with known upper and lower bounds on solution quality. The algorithm works by employing an optimal solver for (standard) Euclidean TSP and then augmenting the solution by calculating suitable heading vectors for each waypoint.

The application to the occlusion-aware surveillance is preceded by generating a covering set of vantage points through

³The art gallery problem amounts to finding the minimal number of sensors and their positions in a polygonal area with or without polygonal holes such that any point inside the polygonal area can be seen by at least one sensor. In the basic formulation, the sensors are assumed to be omnidirectional.

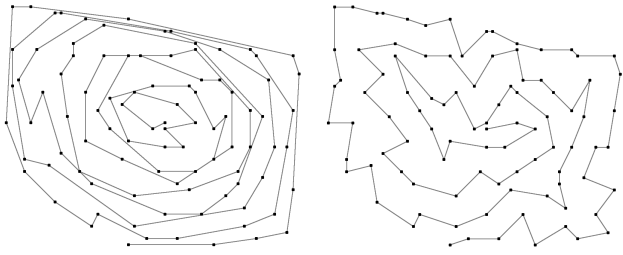


Figure 2: Spiral trajectory consisting of non-relaxed convex hulls (left) vs. relaxed hulls (right); the latter is 40% shorter.

which the UAV is supposed to pass. The set is then given to the alternating algorithm which produces the order in which the points should be visited. The resulting sequence determines the trajectory of the surveilling UAV.

In our implementation, we have used the freely available Euclidian TSP solver LINKERN⁴ and fed the resulting way-point sequence into AGENTFLY path planner [15] to find the shortest path respecting UAV constraints.

4.3 Spiral Algorithm

We have developed the spiral algorithm as a lightweight alternative to the alternating algorithm. Similarly to the alternating algorithm, the application of the spiral algorithm also requires that a covering set of vantage points is first generated. Given a set of vantage points, the algorithm arranges the points in such an order that they can be traversed using a spiral-like path respecting the UAV’s turn radius constraints.

The algorithm is iterative. Given a set of points, the first iteration constructs a convex hull of the set and the points forming the boundary of the hull are then removed from the set. The remaining points are then used as the input for subsequent iterations. The process is repeated until there are no points left. The chain of hulls obtained is then linked together, starting with the outer-most hull. For each pair of neighboring hulls we search the shortest possible link, i.e. a pair of points, through which the hulls can be connected and merged.

The basic algorithm is further improved by relaxing the convex hulls (see function `RelaxConvexHull` in the pseudocode). The idea is to include as many points into each hull as possible while still keeping the hull smooth enough for the UAV to fly through. After constructing each convex hull, we inspect whether there are any points inside the hull that are close enough to the hull’s edge and could be added into the (relaxed) hull without the UAV having to change its current course too much. If such a point is found, the convex hull is relaxed⁵. This relaxation process is executed recursively for each edge of the hull. Further details on the algorithm can be found in [14, 5].

4.4 Zig-zag Algorithm

The input to the zig-zag algorithm is the target area with the list of points of interest. In contrast to the previous two algorithms, the zig-zag algorithm does not require an a priori generated set of vantage points – the consideration

⁴<http://www.tsp.gatech.edu/concorde/>

⁵Note that the resulting polygons are no longer convex

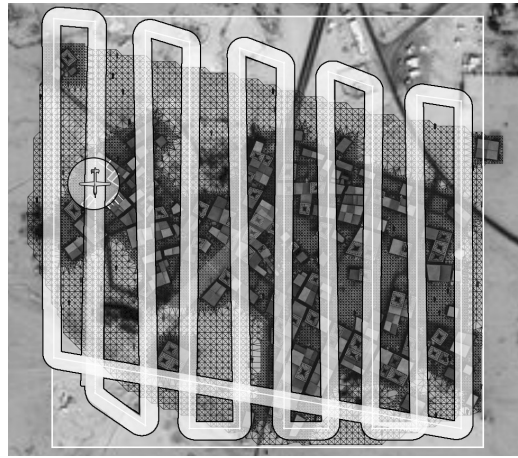


Figure 3: An example trajectory generated by the zig-zag algorithm.

of occlusions is performed *simultaneously* with planning the trajectory.

The algorithm produces zig-zag trajectories (see Figure 3) with the number of rows minimized and the spacing between adjacent rows variable and optimized to minimize the overall trajectory length while ensuring that all ground nodes within the area of interest will be visible to the UAV’s sensors (considering occlusions).

We first describe a single-UAV version of the algorithm. The algorithm takes a rectangular area of interest as its input. A regularly-spaced air navigation grid is created above the area of interest; the grid serves as a basis for occlusion-aware planning of UAV trajectories. The longer of the target area’s two sides is determined; all subsequently generated trajectory rows will be parallel to that side (vertical in the case shown in Figure 3). Next, ground coverage generated by a flight along each such a row of nodes is determined, taking into account occlusions. The algorithm then iterates through individual rows and compares the set of ground nodes covered by the current row with the sets of ground nodes covered by two adjacent rows (previous and next). If all the ground nodes covered by the current row are also covered by its two adjacent rows, the current row is marked as redundant and removed. The process continues until no further row can be eliminated. See the pseudocode of the algorithm in Algorithm 1.

The algorithm produces a set of flight rows which guarantee the coverage of all points of interest. Having found the rows, a feasible trajectory for a UAV with a defined minimum turn radius R can be constructed – the rows are straight segments of the trajectory, and the shortest trajectory connecting the end points of two adjacent rows can be constructed in constant time using the method of Dubins [4].

4.5 Extension to Multiple UAVs

There are three comparably effective ways in which the single-UAV algorithms can be extended for multiple UAVs, each with its specific advantages:

1. All UAVs travel along the whole trajectory with equal spacing between the consecutively following UAVs. The advantage of this approach is that it decreases the in-

```

input : area, ground_nodes, air_grid
output: asetofflightrows
1 begin
2   rows ← GetGridRowsForArea(air_grid, area);
3   foreach row in rows do
4     coverage(row) ←
5     CalculateRowCoverage(row,
6     ground_nodes, area);
7   end
8   foreach curr in rows do
9     prev ← GetPreviousRow(curr, rows);
10    next ← GetNextRow(curr, rows);
11    c ← coverage(curr) \ (coverage(prev) ∪
12    coverage(next))
13    if c = ∅ then
14      remove curr from rows;
15    end
16  end
17  return rows
18 end

```

Algorithm 1: Occlusion-aware zig-zag algorithm

formation age consistently across the whole area. A disadvantage is the initial coordination of the UAVs required to ensure their equal time spacing.

2. The rows covering the target area are divided between the UAVs. This approach is less effective than the first one because it is not always possible to divide the rows evenly. On the other hand, once the rows are distributed, no inter-UAV coordination process is required.
3. The target area is divided evenly between the UAVs and each UAV is then left to plan the zig-zag trajectory on its dedicated part of the area. This scheme has the advantage of simplicity and robustness against communication failures as it does not rely on any coordination except for the straightforward division of the area.

Option 3 has been adopted in the presented work.

5. MULTI-AREA UAV ALLOCATION

We now describe the second building block of the two-stage surveillance – the multi-area allocation algorithm which distributes the available UAVs to the areas of interest, where they are subsequently controlled by the single-area zig-zag algorithm presented in the previous section (the other two algorithms could be used too).

5.1 Multi-Area UAV Allocation Problem

As discussed in Section 3, we assume there are more UAVs than areas. For this case, we define the *multi-area UAV allocation problem*.

Let $a : \mathcal{A} \rightarrow \mathbb{N}^0$ be the assignment function that specifies how many UAVs are allocated to a particular area:

DEFINITION 1. *The estimated age function of an assignment a is the function $E : a \rightarrow \mathbb{R}$ defined as:*

$$E(a) = \sum_{i=1}^n \frac{I(A_i)}{a(A_i)} \quad (2)$$

where $I(A_i)$ is the estimated average information age for area A_i if the area was assigned one UAV only.

The average information age of an area $I(A)$ can be estimated using a lower bound calculated from the dimensions of the area. For an area of proportions w, h , in which points of interest are regularly distributed, and a UAV which travels at constant velocity v in altitude a having a sensor with ground radius $\rho = 2 \cdot a \cdot \sin(\frac{\theta}{2})$, the lower bound on the quality of solution of the information age problem is

$$\underline{I}(A) = \frac{1}{v} \left(\frac{h \cdot w}{2\rho} - \frac{\rho}{\pi} \right) \quad (3)$$

The first term inside the parenthesis is proportional to the number of zig-zag passes that the UAV has to make to see the whole rectangle without any occlusions. The second term compensates for the fact that due to sensor's circular ground footprint, points closer to the center of UAV's trajectory are visible longer.

The multi-area UAV allocation problem then amounts to find such an assignment a^* that the corresponding estimated age $E(a^*)$ is minimal.

5.2 Allocation Algorithm

In this section, we show that an optimum assignment a^* can be found using a greedy algorithm with $O(n^2)$ time complexity. The algorithm Algorithm 2 chooses the next step based on the best improvement of the information age estimate $E(a)$ (line 8).

```

input :  $\mathcal{A}, U$ 
output: a
1 begin
2   // initialization
3   for  $i = 1$  to  $|\mathcal{A}|$  do
4      $a(A_i) \leftarrow 1$ 
5   end
6   // main loop
7   for  $i = 1$  to  $|U| - |\mathcal{A}|$  do
8      $j =$ 
9      $\arg \min_k (E(a) - E(a_{[a(A_k) \leftarrow a(A_k) + 1]}))$ 
10     $a(A_j) \leftarrow a(A_j) + 1$ 
11  end
12  return a

```

Algorithm 2: A greedy algorithm for the multi-area UAV allocation problem.

THEOREM 1. *Algorithm 2 is optimal.*

Proof. The algorithm first assigns one UAV to each task. This is done in $|\mathcal{A}|$ number of steps. Such an assignment is possible because $|U| \geq |\mathcal{A}|$. Let us call l the number of UAVs left unassigned after the first phase. We treat the two cases (i) $l = 0$ and (ii) $l > 0$ separately.

(i) If $l = 0$ the assignment is optimal. Because there is one UAV per task, the estimated age of the assignment is finite. For any other assignment, the estimated age would be infinite, i.e., bigger.

(ii) If $l > 0$. We consider the following construction: For each task A_i , we define its *age improvement function* $E_{A_i}^+$:

$\mathbb{N} \rightarrow \mathbb{R}^+$ as

$$E_{A_i}^+(n) = I(A_i) \cdot \left(\frac{1}{n-1} - \frac{1}{n} \right) \quad (4) \text{ for } n \geq 2 \text{ and}$$

$$E_{A_i}^+(n) = 0$$

for $n = 1$. The age improvement function represents the improvement of the estimated age function E if the number of UAVs assigned to A_i is increased from $n - 1$ to n . Let us note that, in each step, the allocation algorithm increases the number of UAVs assigned to the task with the highest current value $E_{A_i}^+(a(A_i) + 1)$ of the age improvement function. Finally, let us note that the final estimated age of the whole assignment is equal to

$$\sum_{i=1}^{|\mathcal{A}|} \left(I(A_i) - \sum_{j=1}^{a(A_i)} E_{A_i}^+(j) \right). \quad (5)$$

From this fact and from the fact that the age improvement function is strictly decreasing, it follows that choosing the best improvement at each step leads to the assignment with the minimum estimated age. \square

The time complexity of the algorithm is $O(n^2)$ where n is the number of UAVs. More precisely, the complexity is $O(|U| \cdot |\mathcal{A}|)$ because the main loop is executed exactly $|U|$ times and in each execution the maximum improvement of the cost function is calculated, which requires $|\mathcal{A}|$ atomic computations.

6. EXPERIMENTAL RESULTS

To evaluate the performance of the two-stage approach and the quality of produced trajectories, we have carried out a number of experiments, presented in the following two subsections. The first Section 6.1 presents results for single-area surveillance; the second Section 6.2 presents results for the integrated two-stage method – combining the multi-area allocation and the zig-zag algorithms – used for multi-area surveillance.

All evaluation was performed using the AGENTFLY framework [10] for UAV flight and air traffic simulation. The core framework – consisting of UAV flight model, accelerated flight path planning and collision avoidance – has been extended with a realistic on-board sensor model which accurately simulates the effect of occlusions. A screenshot of the simulation testbed in operation is given in Figure 4.

6.1 Single-Area Experiments

We first describe the experimental settings and then present two sets of experimental results. The first set evaluates the performance for a single UAV; all three single-area algorithms are tested. The second set evaluates the multiple UAV case, specifically the dependence of the average information age on the number of employed UAVs; only the best performing zig-zag algorithm is used for this experiment.

Experimental Setting

The specific scenario used for the evaluation is modeled after a real-world settlement with surroundings located in a flat 1500m-by-1500m square area. Buildings are modeled as non-overlapping but possibly adjacent quadrilateral prisms with bases on the $z = 0$ plane. There is a total of 300 buildings with heights in the 6-to-22 m range; the width of

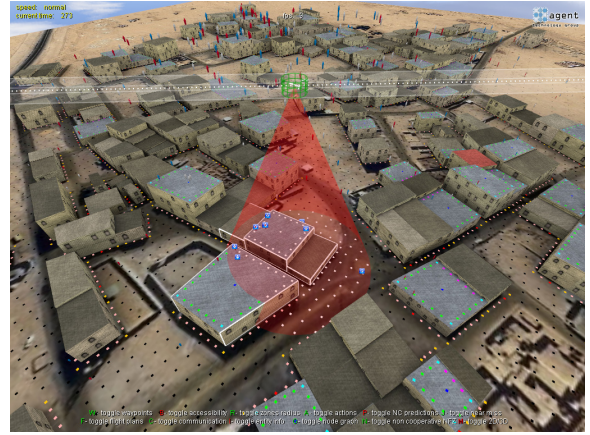


Figure 4: AgentFly UAV simulation testbed with an occlusion-aware sensor model.

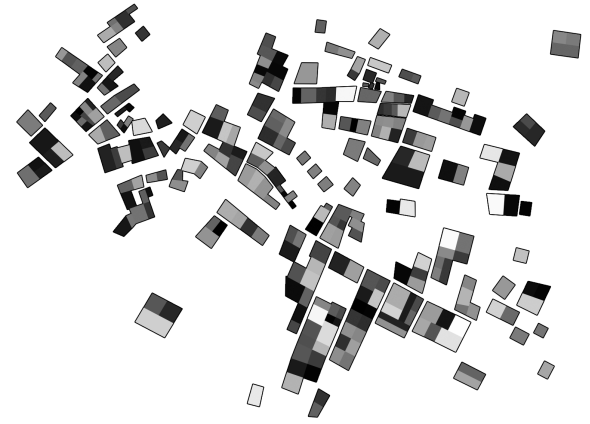


Figure 5: Height map of the urban area used in the empirical evaluation. Highest buildings depicted in very light gray (22m), lowest buildings in black (6m).

the streets range from 3 to 10 meters. The whole 1500m-by-1500m village area is to be surveilled. A visualization of the height map used for the experiments is depicted in Figure 8.

There are a number of configurable parameters of the scenario, summarized in Table 6.1 including the range within which they were varied.

Number of UAVs	1–12
UAV altitude	50–300 m
UAV minimal turning radius	$R = 0\text{m}-50\text{m}$
Sensor field of view angle	$\varphi = 47^\circ$
UAV speed	$v = 25 \text{ m/s}$

The points of interest consisted of a uniform grid covering all ground stretches (streets, intersections, and open areas) of the target area. The distance between the points in the grid was 5m.

Single-Area Single UAV

The average information age for a single UAV surveillance was evaluated for all three occlusion-aware methods (Section 4). The results for different UAV flight altitude and

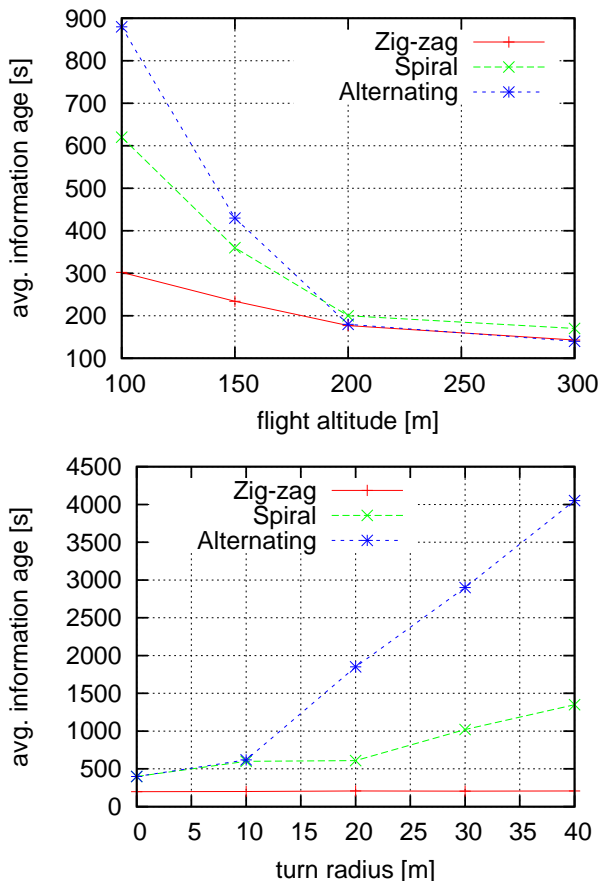


Figure 6: Performance of single-UAV single-area surveillance for different UAV’s flight altitudes (top) and UAV’s minimum turn radii (bottom).

minimum turn radius are given in Figure 6, respectively.

The results show the improving performance of the algorithm with the increasing UAV’s altitude. In addition, the alternating and spiral algorithms show strong sensitivity on UAV’s turn radius – higher values significantly worsen their performance, in particular for the alternating algorithm due to decreased maneuverability of the UAV. In contrast, thanks to composing the flight trajectories from straight row segments, the zig-zag algorithm is virtually unaffected (there is a slight increase if the turn radius is higher than the row spacing).

Single-Area Multiple UAVs

Next we evaluated the performance of single-area surveillance using multiple UAVs, employing the solution proposed in Section 4.5. We only present results for the best performing zig-zag algorithm.

Our literature review has not identified a suitable candidate for comparison for the case of multiple UAVs in a single area. In order to give at least some indication of performance, we compare the information age with the information age estimate $\frac{1}{n}I(A)$ (3). For multiple UAVs in one area, the estimate is divided by the number of UAVs. Figure 7 shows the measured performance of the multi-UAV

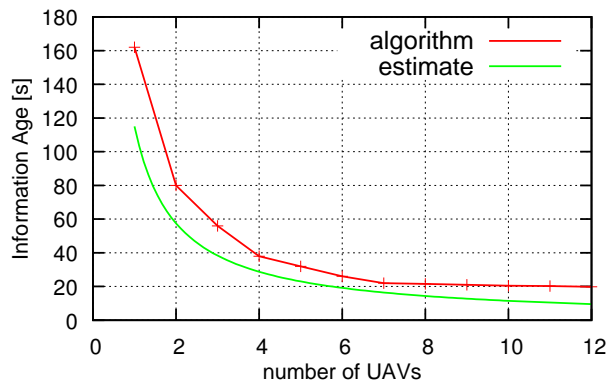


Figure 7: Information age, measured and estimated, for multi-UAV single-area surveillance.

algorithm and the corresponding estimate $\frac{1}{n}I(A)$ ⁶.

6.2 Multi-Area Experiments

Next we evaluated the two-stage multi-area surveillance algorithm employing the zig-zag algorithm as its base single-area occlusion-aware planner. Our main objective was to understand the dependency of the algorithm’s performance on the number and size of the target areas. Similarly to the multi-UAV single-area surveillance, there is no existing algorithm suitable for direct comparison, and we therefore use a theoretical baseline for comparison.

Experimental Setting

The urban environment used was the same as for the single-area experiments except that more areas are defined. Several different combinations of the areas were considered. All the areas are outlined in Figure 8 with their dimensions given in Table 1. All configurations consisted of exactly four areas. The UAVs were flying at 100m with constant velocity of 25 m/s, had minimal turning radius $R = 20m$ and ground radius of sensors $\rho = 76.5m$.

nr.	I	II	III	IV	nr.	I	II	III	IV
1	200	200	200	200	2	400	200	200	200
3	400	100	100	100	4	400	400	200	200
5	400	400	400	100	6	400	300	200	100

Table 1: Lengths of the sides of the square areas used in the experiments. All areas were squares. Organization of the table is the same as the organization of graphs in Figure 9. Arabic numerals denote experiment numbers; roman numerals denote the respective quadrants from Figure 8.

The UAVs were initially placed at the border of the whole area. Nonetheless, because each experiment ran for an extended period of time, the effect of initial conditions was eliminated.

Results

⁶The estimate is no longer a lower bound in general; it can be shown, however, that it *is* a lower bound for vast majority of practical cases, except for degenerate cases where the total footprint of all UAV sensors is close to the size of the area surveilled. We therefore use it for comparison.

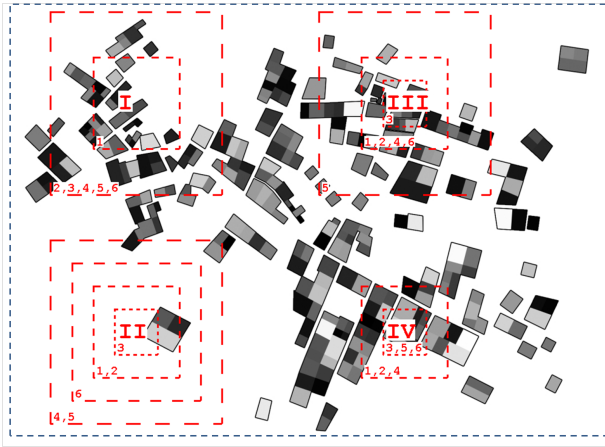


Figure 8: The areas used in the multi-area experiments. The sizes of the numbered areas are given in Table 1.

Before presenting the experimental results we extend the lower bound from Section 5.1 to the case of multiple areas. The extension is based on the fact that surveilling a certain spatial area is most effective if the whole area is made up by a single rectangle rather than by multiple rectangles. To obtain the estimate, we therefore substitute the term $w \cdot h$ in (3) representing the area of a single area with the sum of the areas of all the surveilled areas. The resulting estimate

$$\mathbb{I}(\mathcal{A}, n) = \frac{1}{nv} \left(\frac{\sum_{A_i \in \mathcal{A}} A_i}{2\rho} - \frac{\rho}{\pi} \right) \quad (6)$$

is displayed alongside the empirically measured values in Figure 9. Even though there is a notable difference in the absolute values, the trends are identical. On average, when there was the same number of UAVs and areas (4 in our case) the approach performed worse by 100%. However, for twice as many UAVs as the areas (8 UAVs) it performed worse only by 30%. For three times as many UAVs (12) it performed worse by 27%.

6.3 Discussion

On rectangular areas with uniform distribution of points of interest, the zig-zag algorithm (significantly) outperforms the other two single-area algorithms for most combinations of UAV's altitude and turn radius. In addition, the regular shape of the generated trajectories makes the zig-zag algorithm more predictable and thus easier to work with; the zig-zag trajectories can also be easily split into multiple disjoint segments. Both properties are particularly useful when the trajectory is to be divided between multiple UAVs.

For high flight altitudes, nevertheless, the performance of the zig-zag algorithm is matched by the alternating algorithm. This is because with the increasing sensor altitude, occlusions become less critical, resulting in a decreased number of vantage points and larger distances between them. This in turn renders UAV's motion constraints relatively less critical and optimum flight paths tend to be very close to the solutions of Euclidean TSP, which is the basis of the alternating algorithm. In fact, the alternating algorithm might outperform the zig-zag algorithm for non-convex target ar-

reas or for areas with strongly non-uniform distribution of points of interest – although these cases might be handled by first segmenting each such area into a number of subareas and then employing the multi-area algorithm.

The results for multi-UAV single-area surveillance show that the simple division scheme performs surprisingly well and its performance approaches optimum. The performance for multiple areas depends on the ratio between the number of available UAVs and the total number of target areas. With the increasing ratio, the relative performance with respect to a baseline estimate improves. The requirement to have at least as many UAVs as the areas can be restrictive. A simple solution would be to alter the allocation algorithm so that the absence of UAVs in an area is not penalized by an infinite penalty but only by a finite one corresponding to the current information age of the area, and then run the modified algorithm periodically.

7. RELATED WORK

The problem of multi-UAV surveillance has received some attention lately and a variety of approaches from reactive policies to deliberative search-based methods have been proposed. However, none of the reported approaches explicitly deals with occlusions.

In [9] the authors present an approach for constructing a semi-heuristic control policy for multiple UAVs performing a surveillance task. [1] proposes a class of semi-distributed stochastic navigation algorithms based on the minimization of artificial potentials.

The more deliberative approaches view the surveillance problem as a *routing problem* as e.g. in [11] where the resulting problem is the *traveling salesman problem with time windows*. In some routing problems, the aircraft trajectory constraints have been explicitly considered adopting the Dubins vehicle aircraft model. Important work on routing problems with Dubins vehicles is [13], which presents several algorithms for single- and multi-UAV routing problems including the traveling repairman problem, which can be viewed as a special case of multi-UAV area surveillance problem. Again, occlusions are not considered, though.

8. CONCLUSIONS

We have formulated the problem of multi-UAV surveillance of multiple areas with sensor occlusions as the minimization of the average information age of a set of points of interest. Noting that the problem is intractable in its optimum formulation, we have proposed an approximate two-stage approach which first allocates the UAVs to the areas and then solves the resulting single-area surveillance problems for each of the areas with the allocated UAVs. We have proved the optimality of the allocation algorithm, assuming that there are more UAVs than areas. For the single-area surveillance, we have presented three occlusion-aware algorithms with different performance characteristics.

We have evaluated all methods empirically on a realistic UAV simulation testbed which accurately models UAVs and their on-board sensors. For the most typical rectangular areas, the zig-zag algorithm performs the best for most UAV operational conditions. For multi-UAV single-area and for multi-UAV multi-area problems, we have derived theoretical performance estimates and shown that our two-stage approach performs favorably with respect to these estimates while guaranteeing 100% coverage of the points of interest.

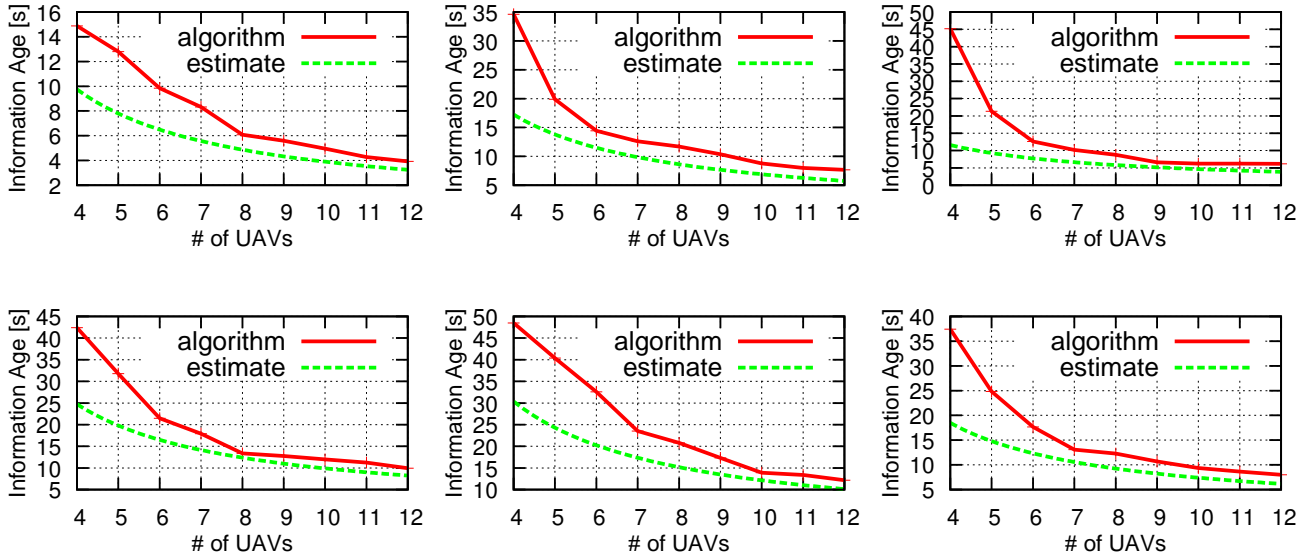


Figure 9: Performance of the multi-area surveillance, both measured and estimated. Configurations of the respective experiments is described in Figure 8 and Table 1. (Configurations 1-3 in the upper row; configurations 4-6 in the lower row).

Acknowledgements

The presented research has been funded by the U.S. Army (grant no. W911NF-08-1-0521.1312AM01) and by the Czech Ministry of Education, Youth and Sports under grant "Decision Making and Control for Manufacturing IIP" (grant no. MSM 6840770038).

9. REFERENCES

- [1] L. Caffarelli, V. Crespi, G. Cybenko, I. Gamba, and D. Rus. Stochastic Distributed Algorithms for Target Surveillance. *Intelligent Systems Design and Applications*, page 137, 2003.
- [2] T. Cox, C. Nagy, M. Skoog, and I. Somers. Civil UAV capability assessment. Technical report, NASA Dryden Research Center, Edwards, CA, 2004.
- [3] M. De Berg, D. Halperin, M. Overmars, and M. Van Kreveld. Sparse arrangements and the number of views of polyhedral scenes. *International Journal of Computational Geometry and Applications*, 7:175–196, 1997.
- [4] L. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, pages 497–516, 1957.
- [5] M. Jakob, E. Semsch, D. Pavlíček, V. Eliáš, and M. Pěchouček. Intelligent Software Agent Control of Combined UAV Operations for Tactical Missions – Month 12 Report. Technical report, Czech Technical University, Prague, December 2009.
- [6] J. Kim and Y. Kim. Moving ground target tracking in dense obstacle areas using UAVs.
- [7] S. LaValle. *Planning algorithms*. Cambridge University Press, 2006.
- [8] M. Marengoni, B. Draper, A. Hanson, and R. Sitaraman. A system to place observers on a polyhedral terrain in polynomial time. *Image and Vision Computing*, 18(10):773–780, 2000.
- [9] N. Nigam and I. Kroo. Persistent Surveillance Using Multiple Unmanned Air Vehicles. In *2008 IEEE Aerospace Conference*, pages 1–14, 2008.
- [10] M. Pechoucek and D. Sislak. Agent-based approach to free-flight planning, control, and simulation. *IEEE Intelligent Systems*, 24(1):14–17, Jan./Feb. 2009.
- [11] J. Ryan, T. Bailey, J. Moore, and W. Carlton. Reactive tabu search in unmanned aerial reconnaissance simulations. In *Proceedings of the 30th conference on Winter simulation*, pages 873–880. IEEE Computer Society Press Los Alamitos, CA, USA, 1998.
- [12] A. Sarmiento, R. Murrieta-Cid, and S. Hutchinson. A multi-robot strategy for rapidly searching a polygonal environment. *Lecture notes in computer science*, pages 484–493, 2004.
- [13] K. Savla. *Multi UAV Systems with Motion and Communication Constraints*. PhD thesis, UNIVERSITY of CALIFORNIA, 2007.
- [14] E. Semsch, M. Jakob, D. Pavlíček, and M. Pěchouček. Autonomous UAV Surveillance in Complex Urban Environments. In *Proceedings of 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2009)*, 2009.
- [15] D. Sislak, P. Volf, and M. Pechoucek. Accelerated A* Path Planning (Extended Abstract). In *AAMAS 2009: Proceedings of the eighth international conference on Autonomous agents and multiagent systems*, 2009.
- [16] H. Vollmer. Computational complexity of constraint satisfaction. In S. B. Cooper, B. Löwe, and A. Sorbi, editors, *CiE*, volume 4497 of *Lecture Notes in Computer Science*, pages 748–757. Springer, 2007.