Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms

Ivan Maza and Anibal Ollero

{imaza,aollero}@cartuja.us.es
Grupo de Robotica, Vision y Control
Escuela Superior de Ingenieros. University of Seville
Camino de los Descubrimientos, s/n
41092 Seville, SPAIN

This paper focuses on the problem of cooperatively searching a given area to detect objects of interest, using a team of heterogenous unmanned air vehicles (UAVs). The paper presents algorithms to divide the whole area taking into account UAV's relative capabilities and initial locations. Resulting areas are assigned among the UAVs, who could cover them using a zigzag pattern. Each UAV has to compute the sweep direction which minimizes the number of turns needed along a zigzag pattern. Algorithms are developed considering their computational complexity in order to allow near-real time operation. Results demonstrating the feasibility of the cooperative search in a scenario of the COMETS multi-UAV project are presented.

1 Introduction

This paper addresses cooperative search problems for UAVs. Cooperative coverage of *a priori* unknown rectilinear environments using mobile robots is discussed in [5]. Ref. [10], uses neural networks to direct robots for complete coverage in complex domains with dynamically moving obstacles. For execution and coordinated control of a large fleet of autonomous mobile robots, Alami et al. propose a Plan Merging Paradigm [1] . The robots incrementally merge their plans into a set of already coordinated plans, through exchange of information about their current state and their future actions.

In recent years, there has been a great deal of work on cooperative control for UAVs. The cooperative control problem that has received the most attention is formation flying [8, 12]. In formation flight, the UAV trajectories are dynamically coupled through the physics of close flight. By exploiting the physical structure of the problem, path planning for formation flying applications can be reduced to path planning algorithms for single vehicles [13].

Unfortunately, there are many other cooperative control problems that do not admit solutions that are extensions of single vehicle solutions. These include cooperative rendezvous [11], coordinated target assignment and intercept [3], multiple task allocation [4], and ISR scenarios [6].

The full solution to many of these cooperative control problems are NP-hard. While formation flight problems can be solved efficiently using numerical methods, there is a need to identify others classes of cooperative control problems that can also be solved efficiently.

Research presented in this paper has been carried out in the framework of the COMETS Project (Real-time coordination and control of multiple heterogeneous unmanned aerial vehicles). In this EU Project, several missions have been considered: detection, aerial mapping, alarm confirmation, fire monitoring, object/person tracking, communications relay, etc. In the mission considered in this paper, a team of heterogeneous UAVs has to cooperatively search an area to detect objects of interest (fire, cars, etc). The problem has been decomposed into the subproblems of (1) determine relative capabilities of each UAV, (2) cooperative area assignment, and (3) efficient area coverage.

The paper is organized as follows: In Section 2 an algorithm based on a divide-and-conquer, sweep-line approach is applied to solve the area partition problem. In Section 3 we introduce the sensing capabilities considered on board the UAVs and the implications with respect to the following sections. A discussion about the covering algorithm that each UAV should use is presented in Section 4. In Section 5 the flexibility in case of re-planning and the complexity of the method outlined is analyzed. Simulations are presented in Section 6 and finally conclusions are given in Section 7.

2 Area decomposition for UAV workspace division

In [9] it was presented a polygon decomposition problem, the *anchored area partition problem*, which has applications to our multiple-UAV terrain-covering mission. This problem concerns dividing a given polygon $\mathcal{P}$ into n polygonal pieces, each of a specified area and each containing a certain point (site) on its boundary. In our case, there are n UAVs U_i, $i = 1, \dots, n$, each placed at a distinct starting point S_i on the boundary of the polygonal region $\mathcal{P}$ (see Figure 1). The team of UAVs has the mission of completely covering the given region, and to do this most efficiently, the region $\mathcal{P}$ should be divided among the UAVs accordingly with their relative capabilities. Within its assigned region, each vehicle will execute a covering algorithm which is discussed in Section 4.

The algorithm applied in this paper solves the case when $\mathcal{P}$ is convex and contains no holes (no obstacles), which is a preliminar scenario considered in COMETS. A generalized version that handles nonconvex and nonsimply connected polygons is also presented in [9], but computational complexity increases in this case.

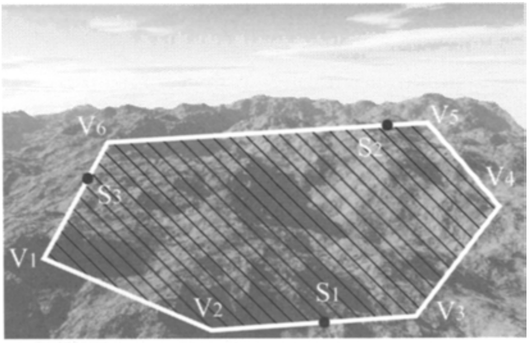

Fig. 1. Initial scenario considered.

2.1 Relative capabilities of the UAVs

The low cost UAVs currently involved in the COMETS system are strongly constrained in flying endurance and range. Then, in a first approximation, maximum range of the UAVs seems to be a good measure of their capabilities to perform the mission considered. As UAVs are heterogeneous, range information should be scaled taking into account factors like flight speed and altitude required for the mission, sensitivity to wind conditions, sensing width (due to different camera's fields of view), etc.

Based on the relative capabilities of the vehicles, it is determined what proportion of the area of the region $\mathcal{P}$ should be assigned to each of them. These proportions are represented by a set of values c_i, $i = 1, \dots, n$, with $0 < c_i < 1$ and $\sum_{i=1}^{n} c_i = 1$. Therefore, the problem considered is as follows: Given a polygon $\mathcal{P}$ and n points (sites) $S_1, \dots, S_n$ on the polygon, divide the polygon into n nonoverlapping polygons $\mathcal{P}_1, \dots, \mathcal{P}_n$ such that $\text{Area}(\mathcal{P}_i) = c_i \text{Area}(\mathcal{P})$ and S_i is on $\mathcal{P}_i$.

2.2 Algorithm

Let $S_1, \dots, S_n$ be a set of sites (start positions of the UAVs), each of them with an *area requirement*, denoted $\text{AreaRequired}(S_i)$, which specifies the desired area of each polygon $\mathcal{P}_i$.

A polygon $\mathcal{P}$ which contains q sites is called a *q-site polygon*, and is called *area-complete* if $\text{AreaRequired}(S(\mathcal{P})) = \text{Area}(\mathcal{P})$ where $\text{AreaRequired}(S(\mathcal{P}))$ is the sum of the required areas by the sites in $\mathcal{P}$.

As it has been stated before, it is assumed a polygon $\mathcal{P}$ convex and with no holes (no obstacles). In this case, it has been shown (see Ref. [9]) that the desired area partition can be achieved using $n-1$ line segments, each of which divides a given q-site ($q > 1$) area-complete polygon $\mathcal{P}$, into two smaller convex polygons — a q_1-site area-complete polygon and a q_2-site area-complete polygon with $q_1 + q_2 = q$ and $q_1, q_2 > 0$. The computation of each segment can be done using an algorithm based on a divide-and-conquer, sweep-line approach presented in [9]. This procedure should be called exactly $n-1$ times to partition a convex, n-site area-complete polygon into n convex, 1-site area-complete polygons.

3 Sensing capabilities

A team of UAVs has to perform a cooperative search operation over an area to detect objects of interest. Consider a Base Coordinate System (BCS), fixed in the environment (x-axis towards north, y-axis west, z-axis up) and UAVs equipped with sensors and cameras. Sensors allow the vehicles to determine their own coordinates relative to BCS and those of any point detected in its sensing region.

The UAVs are assumed to have cameras without orientation devices. In fact, light UAVs have strong payload constraints that may preclude the use of ginbals and other devices to change the orientation of the on-board cameras.

Each UAV has associated an UCS (UAV Coordinate System) that changes its point of origin and its orientation with the movement of the vehicle (x-axis forward, y-axis left, z-axis up). On board cameras are fixed, oriented in the x-z plane of the UCS and defined by the angle $(-\alpha)$ with respect to the x-axis.

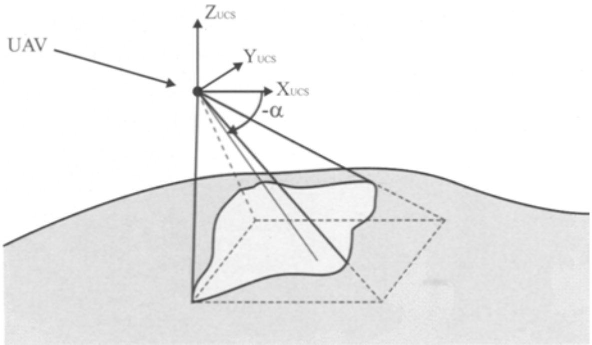

Fig. 2. The imaged area is the intersection of the image pyramid and the terrain.

As the UAV moves along a straight line path between waypoints taking shots, the *image piramid* of the camera defines an *imaged area* on the terrain (see Figure 2). Considering a plain terrain, it can be shown that the sensing width of an UAV moving in the x-z plane of the UCS is given by:

$$w = 2z_{BCS}\tan\gamma\left[\sin\alpha + \cos\alpha\tan\left(\frac{\pi}{2} - \alpha - \beta\right)\right] \tag{1}$$

where z_{BCS} is the altitude of the UAV, and angles β and γ determine the field of view of the camera.

As the planar algorithm for covering a given area is based on a zigzag pattern, the spacing of the parallel lines will be determined in first approximation by equation (1). To be able to generalize this planar algorithm directly to the three-dimensional environment considered, the nonplanar surface (area) to be covered must be a *vertically projectively planar surface.* That is, a vertical line passing through any point on the surface intersects it at only one point.

4 Individual areas coverage algorithm

Once each UAV has an area assigned (corresponding to a convex polygon $\mathcal{P}_i$), an algorithm is needed to cover this area searching for objects of interest. Those convex areas can be easily and efficiently covered by back and forth motion along rows perpendicular to the sweep direction (simulations have shown that in general this pattern is faster than the spiral pattern). The time to cover an area in this manner consists of the time to travel along the rows plus the time to turn around at the end of the rows. Covering an area for a different sweep direction results in rows of approximately the same total length; however, there can be a large difference in the number of turns required as illustrated in Figure 3. In the COMETS Project, autonomous helicopters are included in the heterogeneous team of UAVs. Helicopter turns take a significant amount of time: the helicopter must slow down, stay in hovering, make the turn, and then accelerate.

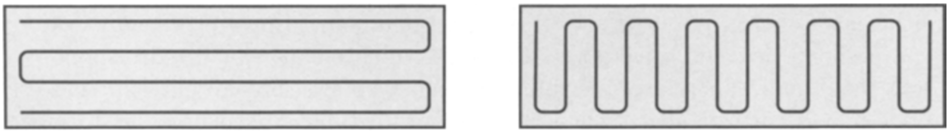

Fig. 3. The number of turns is the main factor in the cost difference of covering a region along different sweep directions.

We therefore wish to minimize the number of turns in an area, and this is proportional to the altitude of the polygon measured along the sweep direction. The *altitude* of a polygon is just its height. We can use the *diameter function* $d(\theta)$ to describe the altitude of a polygon along the sweep direction. For a given angle θ, the diameter of a polygon is determined by rotating the polygon by $-\theta$ and measuring the height difference between its highest and lowest point. The altitude of a polygon $\mathcal{P}_i$ for a sweep direction at an orientation of α is $d_{\mathcal{P}_i}\left(\alpha - \frac{\pi}{2}\right)$.

The shape of a diameter function can be understood by considering the height of the polygon as it rolls along a flat surface (Figure 4). Starting with one edge resting on the surface, we can draw a segment from the pivot vertex to another vertex of the polygon, and the height of the polygon will be determined by this vertex. Whenever the polygon has rolled on to the next side

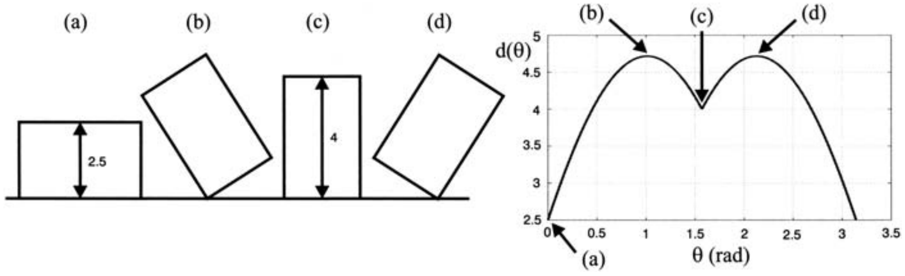

Fig. 4. Example of a simple diameter function.

or when an edge at the top of the polygon becomes parallel to the surface, we will change to a different segment (from a different pivot vertex or to a different top vertex). Therefore, a diameter function has the following form for an n sided convex polygon:

$$d(\theta) = \begin{cases} k_1 \sin(\theta + \phi_1) & \theta \in [\theta_0, \theta_1) \\ k_2 \sin(\theta + \phi_2) & \theta \in [\theta_1, \theta_2) \\ \vdots & \\ k_{2n} \sin(\theta + \phi_{2n}) & \theta \in [\theta_{(2n-1)}, \theta_{2n}) \end{cases} \tag{2}$$

where $\theta_0 = 0$ and $\theta_{2n} = 2\pi$. The diameter function is piecewise sinusoidal; its "breakpoints" θ_i occur when an edge of the rotated polygon is parallel to the horizontal. The minimum of the diameter function must lie either at a critical point ($d'_{\mathcal{P}_i} = 0$) or at a breakpoint. However, for any critical point in between breakpoints $d''_{\mathcal{P}_i} < 0$, which means that it corresponds to a maximum. Therefore, the minimum must lie at a breakpoint and these breakpoints correspond to when the sweep direction is perpendicular to an edge of the perimeter. Testing each of these sweep directions, the minimum can be determined.

A similar approach can also be applied when obstacles are present inside the areas. In this case, the altitude to be minimized is the sum of the diameter function of the perimeter plus the diameter functions of the obstacles.

5 Complexity analysis and reconfiguration process

Special attention has been focused to this issue due to the real time operation required in the COMETS Project.

The computation of the full partition of a convex n-site polygon $\mathcal{P}$ with v vertices, requires $\mathcal{O}(n-1)(n+v)$ time in the worst case. Resulting polygons are assigned to the UAVs, and each of them has to compute the sweep direction which minimizes the number of turns needed along the zigzag pattern. It only implies that each UAV has to test a number of directions equal to the number of edges of its assigned polygon $\mathcal{P}_i$.

Therefore, the whole process has a low computational cost which could also be shared easily among a control centre and the UAVs:

- Each UAV computes its relative capabilities (simple algebraic expressions).
- The control centre computes the complete partition and assigns the resulting areas to each UAV ($\mathcal{O}(n-1)(n+v)$ time in the worst case).
- Each UAV determines its more efficient sweep direction (test a number of directions equal to the number of edges of its polygon).

The functionality of this control centre could also be performed by an UAV with enough computational capability.

If system reconfiguration is needed, the system can quickly adapt to the new scenario. For example, if an UAV is lost, remaining UAVs have to perform the detection mission properly. It implies that a new area partition process must be triggered. In that case, initial locations of the UAVs are not in the boundary of the given area and the algorithm described in Section 2.2 is not valid. A different algorithm described in [9] should be applied, but computational complexity remains bounded and low. In the next section, this re-planning has been handled with a minor modification of the algorithm described in Section 2.2, due to the low number of vehicles involved.

6 Implementation details and simulations results

Algorithms have been implemented in C++ using the CGAL library [7] for computational geometry support.

Table 1. Initial coordinates, camera angles, sensing width and relative capabilities of the UAVs.

	x_{BCS}(m)	y_{BCS}(m)	z_{BCS}(m)	α_i(rad)	β_i(rad)	γ_i(rad)	w_i(m)	c_i(%)
UAV1	190.00	0.00	29.00	$\pi/2$	$\pi/8$	$\pi/8$	24.02	24.92
UAV2	550.00	100.00	34.00	$\pi/3$	$\pi/7$	$\pi/8$	25.45	41.81
UAV3	225.38	412.69	20.00	$\pi/3$	$\pi/6$	$\pi/6$	20.00	33.27

In this simulation, three UAVs have to search an area defined by a convex polygon with seven edges. We assume different cameras on board the UAVs, each of them defined by different values of the angles α, β and γ. In Table 1, initial coordinates of the UAVs and their relative capabilities (c_i – see Section 2.1) are listed. Those values for c_i have been obtained via an estimation of the maximum range in function of parameters like remaining fuel, specific consumption, flight speed, etc. (see Ref. [2]) Using equation (1), and assuming constant altitudes during the mission, sensing width (w_i) of each UAV can be easily derived (see also Table 1).

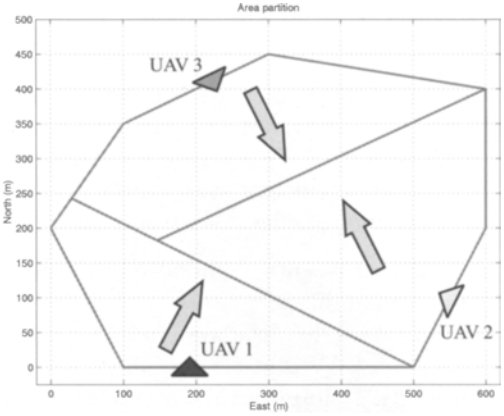

Fig. 5. Area partition simulation results. Optimal sweep directions have been represented by arrows.

Area partition has been computed using the algorithm presented in Section 2.2. The resulting assignment is shown in Figure 5. It can be seen that each UAV has been assigned an area (convex polygon) according with its relative capabilities.

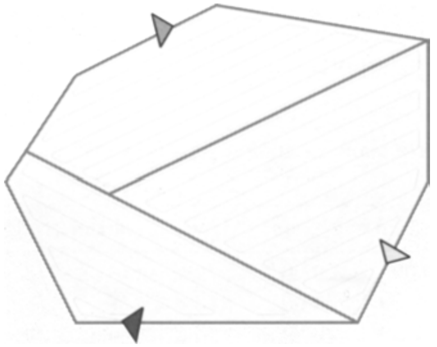

Fig. 6. Resulting zigzag patterns minimizing the number of turns required.

Each UAV has to find the optimal sweep direction which minimizes its assigned polygon's altitude. As it has been explained in Section 4, only the directions which are perpendicular to the edges of each polygon have to be tested. Resulting directions have been represented by arrows in Figure 5. Then, each UAV has to compute the waypoints needed to follow a zigzag pattern perpendicular to those directions (see Figure 6). Distance between parallel lines depends on the sensing width of the UAV.

Finally, a reconfiguration process has been simulated. When UAV3 is lost, remaining UAVs have to cover the whole area. A new area partition process has to be triggered and new sweep directions are followed (see Figure 7).

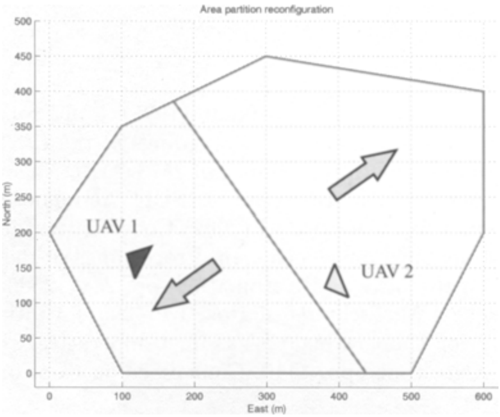

Fig. 7. UAV3 is lost and remaining UAVs have to reconfigure their flight plans to cover the whole area.

7 Conclusions and future research

The problem of cooperative searching a given area (convex polygon) by a team of UAVs taking into account their different sensing and range capabilities has been considered. The solution adopted in this paper is fully adapted to a simple scenario inspired by experiments developed in the COMETS Project (convex polygons and no obstacles), but all the algorithms could be extended to more complex problems with bounded (and relatively low) computational load. It provides a spectrum of solutions useful for real-time implementation (experiments are expected for next year).

It would be very interesting to modify the altitude of the UAVs during the mission execution to maximize the capabilities of their cameras (increase the altitude in sectors with low detection probability).

The methods presented in this paper can be easily extended to the cooperation of autonomous aerial and ground vehicles, which is being addressed in the framework of the CROMAT Spanish project.

Acknowledgements

This work was partially funded by European Union COMETS Project (IST-2001-34304). The authors would like to thank LAAS (Laboratory for Analysis and Architecture of Systems) for their technical reports in COMETS Project, which have been very useful to focus this research. Partial funding of the CROMAT Project of the Spanish Research and Development Program (DPI2002-04401-C03-03) is also acknowledged.

References

1. Alami R, Robert F, Ingrand F, Suzuki S (1995) Multi-robot cooperation through incremental plan-merging. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2573–2579. Nagoya, Japan
2. Barcala M, Rodriguez A (1998) Helicopteros. EUIT Aeronautica, Madrid
3. Beard R W, McLain T W, Goodrich M (2002) Coordinated target assignment and intercept for unmanned air vehicles. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2581–2586. Washington
4. Bellingham J, Tillerson M, Richards A, How J P (2001) Multi-task allocation and path planning for cooperating UAVs. In: Cooperative Control: Models, Applications and Algorithms, pp. 1–19, Conference on Coordination, Control and Optimization.
5. Butler Z J, Rizzi A A, Hollis R L (2000) Cooperative coverage of rectilinear environments. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2722–2727. San Francisco, CA
6. Chandler P R, Pachter M, Swaroop D, Fowler J M, Howlett J K, Rasmussen S, Schumacher C, Nygard K (2002) Complexity in UAV cooperative control. In: Proceedings of the American Control Conference. Anchorage, AK
7. Computational Geometry Algorithms Library (CGAL). Web address: `http://www.cgal.org/`
8. Giulietti F, Pollini L, Innocenti M (2000) Autonomous formation flight. IEEE Control Systems Magazine 20:34–44
9. Hert S, Lumelsky V (1998) Polygon area decomposition for multiple-robot workspace division. International Journal of Computational Geometry and Applications, 8(4):437-466.
10. Luo C, Yang S X, Stacey D A, Jofriet J C (2002) A solution to vicinity problem of obstacles in complete coverage path planning. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 612–617. Washington DC
11. McLain T, Beard R (2000) Cooperative rendezvous of multiple unmanned air vehicles. In: Proceedings of the AIAA Guidance, Navigation and Control Conference, paper no. AIAA 2000–4369. Denver, CO
12. Pachter M, D'Azzo J J, Proud A W (2001) Tight formation flight control. AIAA Journal of Guidance, Control and Dynamics 24:246–254
13. Pledgie S T, Hao Y, Ferreira A M, Agrawal S K, Murphey R (2002) Groups of unmanned vehicles: Differential flatness, trajectory planning, and control. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3461–3466. Washington DC
14. Wesley H., Huang W H (2001) Optimal line-sweep-based decompositions for coverage algorithms. In: Proceedings of the 2001 IEEE International Conference on Robotics and Automation, 1:27–32. Seoul, Korea