Multi-Robot 3D Coverage of Unknown Areas

Alessandro Renzaglia, Lefteris Doitsidis, Agostino Martinelli, Elias Kosmatopoulos

▶ **To cite this version:**

Alessandro Renzaglia, Lefteris Doitsidis, Agostino Martinelli, Elias Kosmatopoulos. Multi-Robot 3D Coverage of Unknown Areas. The International Journal of Robotics Research, 2012, vol 31 no 6 pp 738-752. 10.1177/0278364912439332 . hal-00751309

HAL Id: hal-00751309
https://hal.science/hal-00751309

Submitted on 13 Nov 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Robot 3D Coverage of Unknown Areas

Alessandro Renzaglia
INRIA Rhône-Alpes, Grenoble, France
alessandro.renzaglia@inrialpes.fr

Lefteris Doitsidis
Department of Electronics,
Technological Educational Institute of Crete, Chania, Greece &
Informatics & Telematics Institute, CERTH, Thessaloniki, Greece
ldoitsidis@chania.teicrete.gr

Agostino Martinelli
INRIA Rhône-Alpes, Grenoble, France
agostino.martinelli@inrialpes.fr

Elias B. Kosmatopoulos
Department of ECE,
Democritus University of Thrace, Xanthi, Greece &
Informatics & Telematics Institute, CERTH, Thessaloniki, Greece
kosmatop@dssl.tuc.gr

October 6, 2011

Abstract

The problem of deploying a team of flying robots to perform surveillance coverage mission over an *unknown* terrain of complex and non-convex morphology is presented. In such a mission, the robots attempt to maximize the part of the terrain that is visible while keeping the distance between each point in the terrain and the closest team member as small as possible. A trade-off between these two objectives should be fulfilled given the physical constraints and limitations imposed at the particular application. As the terrain's morphology is unknown and it can be quite complex and non-convex, standard algorithms are not applicable to the particular problem treated in this paper. To overcome this, a new approach based on the Cognitive-based Adaptive Optimization (CAO) algorithm is proposed and evaluated. A fundamental property of this approach is that it shares the same convergence characteristics as those of constrained gradient-descent algorithms (which require perfect knowledge of the terrain's morphology and optimize surveillance coverage subject to the constraints the team has to satisfy). Rigorous mathematical arguments and extensive simulations establish that the proposed approach provides a scalable and efficient methodology that incorporates any particular physical constraints and limitations able to navigate the robots to an arrangement that (locally) optimizes surveillance coverage.

I Introduction

The use of multi-robot teams has gained a lot of attention in recent years. This is due to the extended capabilities that the teams have to offer comparing to the use of a single robot for the same task. Robot teams can be used in a variety of missions including: surveillance in hostile environments (i.e., areas contaminated with biological, chemical or even nuclear wastes), environmental monitoring (i.e., air quality monitoring, forest monitoring) and law enforcement missions (i.e., border patrol), etc. In all the aforementioned tasks the deployment of limited resources (robots) to optimize the monitoring of the area is the key issue. In order to achieve this purpose, the trajectories of the robots should be designed, in real-time, so that:

(O1) the part of the terrain that is monitored (i.e., is visible) by the robots is maximized;

(O2) for every point in the terrain, the closest robot is as close as possible to that point.

The second objective is necessary for two practical reasons: (a) firstly, the closer is the robot to a point in the terrain the better is, in general, its sensing ability to

monitor this point and (b) secondly, in many multi-robot coverage applications there is the necessity of being able to intervene as fast as possible in any of the points of the terrain with at least one robot.

The majority of existing approaches for multi-robot surveillance coverage, which concentrate mostly on the 2D case of ground robots, deal only with one of the objectives (O1) or (O2) and in most of them the terrain morphology is considered convex and/or known. In such cases the problem of multi-robot surveillance coverage can be seen to be equivalent to a standard optimization problem where the robot trajectories are generated according to a gradient-descent or gradient-descent-like methodology. However, in the case where it is required that both of the objectives (O1) and (O2) are addressed and the terrain's morphology is non-convex and unknown, standard optimization tools are not applicable anymore as these tools require full knowledge of an objective function that depends on the unknown terrain's morphology.

To approach this problem, we propose a new solution that is based on the recently introduced Cognitive-based Adaptive Optimization (CAO) algorithm (Kosmatopoulos, 2009), (Kosmatopoulos and Kouvelas, 2009). The main advantage of CAO as compared to standard optimization tools is that it does not require that the objective function to be optimized is explicitly known; CAO instead requires that at each-time instant a value (measurement) of this objective function is available. Then, if it is possible to define an objective function which is available for measurement for every given team configuration, the CAO methodology will be directly applicable to the problem of surveillance coverage treated in this paper. Rigorous arguments establish that, despite the fact that the terrain's morphology is unknown, the CAO methodology shares the same convergence characteristics as those of constrained gradient-descent algorithms (which require perfect knowledge of the terrain's morphology and optimize surveillance coverage subject to the constraints the robot team has to satisfy). As a result, CAO navigates the robots to an arrangement that locally optimizes the surveillance coverage criterion while satisfying physically-imposed constraints such as that the robots should not leave a prespecified area or they should not hit the terrain.

Both objectives (O1) and (O2) can be characterized by two distinct objective functions. In general, one cannot simultaneously optimize both functions, unless the functions share common optima. Hence, the idea is to optimize a combined objective function that strikes a compromise between maximizing visible area and minimizing the distance of the robots to points in the environment. By introducing such objective function, we achieve to render the CAO algorithm applicable to the particular problem of 3D multi-robot surveillance coverage treated in this paper. This objective function depends on the unknown terrain's characteristics and thus its explicit form is not known. However, for any given team configuration the value of this objective function can be directly computed from the robots' sensor

measurements, and thus the CAO algorithm can be applied to the problem at hand by using such an objective function.

It has to be emphasized that apart from rendering the optimization problem solvable, the CAO-based approach preserves additional attributes that make it particularly tractable: it can handle in a easily fashion a variety of physical constraints and limitations and it is fast and scalable.

We finally mention that the CAO approach extends the popular Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm (Spall, 1992). The difference between the SPSA and the CAO approach is that SPSA employs an approximation of the gradient of an appropriate cost function using only the most recent experiments, while the CAO approach employs linear-in-the-parameters approximators that incorporate information of a user specified time window of the past experiments together with the concept of candidate perturbations for efficiently optimizing the unknown function. Comparative evaluations that were performed on complicated optimization problems have shown that CAO exhibits significantly better convergence properties than SPSA (Kosmatopoulos et al., 2007, Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009). Moreover, CAO was shown to exhibit satisfactory (local) convergence characteristics in particular problems where SPSA failed to provide convergent solutions for any choice of its design parameters, (Kosmatopoulos et al., 2007, Kosmatopoulos and Kouvelas, 2009).

It is mentioned that the CAO or the SPSA do not create an approximation or estimation of the obstacles location and geometry; instead, they on-line produce a local approximation of the unknown cost function the robots are called to optimize. For this reason, they require simple – and thus scalable – approximation schemes to be employed.

I.1 Related Work

The majority of approaches for multi-robot surveillance coverage concentrate on objective (O2) described in the previous section. In (Cortés et al., 2004), the authors present a controller for the coverage with a team of mobile robots of a convex environment, i.e., without obstacles, based on the Voronoi partition. A similar approach, for a convex environment, is proposed in (Schwager et al., 2006), where additionally the robots estimate a function indicating the relative importance of different areas in the environment, using information from the sensors. A possible approach for non-convex regions is proposed in (Pimenta et al., 2008). In this work the Voronoi partition is obtained by using the geodesic distance instead of the Euclidean one taking into account the particular topology of the problem. In (Howard et al., 2002b), the same problem is approached by using the artificial potential field method. Another possible solution for environments which include

obstacles is proposed in (Breitenmoser et al., 2010b): the main idea is to combine the classical Voronoi coverage with the Lloyd algorithm and the local path planning algorithm TangentBug. In all the aforementioned works the regions to cover are in 2D. In (Breitenmoser et al., 2010a) the authors approach also the problem of deploying a team of robots on a non-planar surface in 3D space.

As far as it concerns objective (O1) described in the previous section, different solutions have been proposed in the literature. In (Ganguli et al., 2005) the authors propose a gradient-based algorithm for the case of a single robot case and they prove that the visible area is almost everywhere a locally Lipschitz function of the observer location. In (Ganguli et al., 2007), an approach for the multi-robot problem is presented based on the assumption that the environment is simply connected. The visibility problem is also related with the Art Gallery Problem where the goal is to find the optimum number of guards in a non-convex environment so that each point of the environment is visible by at least one guard (Agarwal and Sharir, 1998), (Shermer, 1992). All the aforementioned solutions are based on the hypothesis that a given point can be monitored regardless of its distance from the robot. An incremental algorithm which takes into consideration also a maximum monitoring distance is presented in (Howard et al., 2002a). In (Schwager et al., 2009), the authors consider the coverage of a 2D region by using a team of hovering robots. In this case, information per pixel is proposed as optimization criterion.

To the best of our knowledge, the problem of considering the two objectives simultaneously to cover a 3D region by using a team of flying robots has never been investigated so far. To do that we propose to use a new stochastic optimization method, the CAO algorithm. The many advantages of using stochastic gradient descent algorithms, like the SPSA algorithm, to approach a sensor-based deployment problem have already been highlighted in (Ny and Pappas, 2010). In this work, the authors proposed applications such as: coverage with heterogeneous sensors and source seeking with stochastic wireless connectivity constraints.

The rest of the paper is organized as follows. In section 2 we describe in detail the cognitive based adaptive optimization approach, while in section 3 we formulate the problem for the 3D multi-robot coverage over unknown terrains. In section 4 extensive experimental results are presented and finally in section 5 concluding remarks and future research are discussed.

II The Cognitive-based Adaptive Optimization Approach

The Cognitive-based Adaptive Optimization (CAO) approach (Kosmatopoulos et al., 2007, Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009) was originally developed and analyzed for the optimization of functions for which an explicit form is unknown but their measurements are available as well as for the adaptive fine-tuning of large-scale nonlinear control systems. In this section, we will describe how the CAO approach can be appropriately adapted and extended so that it is applicable to the problem of multi-robot coverage. More explicitly, let us consider the problem where M robots are involved in a coverage task, attempting to optimize a given *coverage criterion*. The coverage criterion is a function of the robots' positions or poses (positions and orientations), i.e.,

$$J_k = \mathcal{J}\left(x_k^{(1)}, \ldots, x_k^{(M)}\right) \tag{2.1}$$

where $k = 0, 1, 2, \ldots$ denotes the time-index, J_k denotes the value of the coverage criterion at the k-th time-step, $x_k^{(1)}, \ldots, x_k^{(M)}$ denote the position/pose vectors of robots $1, \ldots, M$, respectively, and $\mathcal{J}$ is a nonlinear function which depends – apart from the robots' positions/poses – on the particular environment where the robots live; for instance, in the 2D case the function $\mathcal{J}$ depends on the location of the various obstacles that are present, while in the 3D case with flying robots monitoring a terrain, the function $\mathcal{J}$ depends on the particular terrain morphology.

Due to the dependence of the function $\mathcal{J}$ on the particular environment characteristics, the *explicit form of the function $\mathcal{J}$ is not known* in most practical situations; as a result, standard optimization algorithms (e.g., steepest descent) are not applicable to the problem in hand. However, in most practical cases, like the one treated in this paper, the current value of the coverage criterion can be estimated from the robots' sensor measurements. In other words, at each time-step k, an estimate of J_k is available through robots' sensor measurements,

$$J_k^n = \mathcal{J}\left(x_k^{(1)}, \ldots, x_k^{(M)}\right) + \xi_k \tag{2.2}$$

where J_k^n denotes the estimate of J_k and ξ_k denotes the noise introduced in the estimation of J_k due to the presence of noise in the robots' sensors. Please note that, although it is natural to assume that the noise sequence ξ_k is a stochastic *zero-mean* signal, it is not realistic to assume that it satisfies the typical Additive White Noise Gaussian (AWNG) property even if the robots' sensor noise is AWNG: as $\mathcal{J}$ is a nonlinear function of the robots' positions/poses (and thus of the robots' sensor measurements), the AWNG property is typically lost.

Apart from the problem of dealing with a criterion for which an explicit form is not known but only its noisy measurements are available at each time, efficient robot coverage algorithms have additionally to deal with the problem of restricting the robots' positions so that obstacle avoidance as well as robot formation constraints are met. In other words, at each time-instant k, the vectors $x_k^{(i)}, i = 1, \ldots, M$ should satisfy a set of constraints which, in general, can be represented as follows:

$$\mathcal{C}\left(x_k^{(1)}, \ldots, x_k^{(M)}\right) \leq 0 \tag{2.3}$$

where $\mathcal{C}$ is a set of nonlinear functions of the robots' positions/poses. As in the case of $\mathcal{J}$, the function $\mathcal{C}$ depends on the particular environment characteristics (e.g., location of obstacles, terrain morphology) and an explicit form of this function may be not known in many practical situations; however, it is natural to assume that the coverage algorithm is provided with information whether a particular selection of robots' positions/poses satisfies or violates the set of constraints (2.3).

Given the mathematical description presented above, the multi-robot coverage problem can be mathematically described as the problem of moving $x_k^{(1)}, \ldots, x_k^{(M)}$ to a set of positions/poses that solves the following constrained optimization problem:

$$\begin{array}{ll} \text{minimize} & (2.1) \\ \text{subject to} & (2.3)\,. \end{array} \tag{2.4}$$

As already noticed, the difficulty in solving, in real-time and in real-life situations, the constrained optimization problem (2.4) lies in the fact that explicit forms for the functions $\mathcal{J}$ and $\mathcal{C}$ are not available. To circumvent this difficulty, the CAO approach, appropriately modified to be applicable to the problem in hand, is adopted. Indeed this algorithm is capable of efficiently dealing with optimization problems for which the explicit forms of the objective function and constraints are not known, but noisy measurements/estimates of these functions are available at each time-step. In the following, we describe the CAO approach as applied to the multi-robot coverage problem described above.

It has to be emphasized that the CAO algorithm presented here is an extension of the CAO versions presented and analyzed in (Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009). The main difference is that while (Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009) address the unconstrained version of the problem (2.4), in the present paper the CAO approach of (Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009) has to be extended so that it efficiently takes care of the constraint (2.3). In order to do so, the CAO approach of (Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009) is augmented by a special – yet simple – projection mechanism. Theorem 1 establishes that the introduction of such a projection mechanism does not destroy the nice properties of the CAO

approach of (Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009); as a matter of fact, according to Theorem 1 presented below, the CAO algorithm used in this paper is proven to be approximately a *projected gradient-descent* algorithm, while the ones of (Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009) have been established to be approximate unconstrained *gradient-descent* algorithms.

As a first step, the CAO approach makes use of function approximators for the estimation of the unknown objective function $\mathcal{J}$ at each time-instant k according to

$$\hat{J}_k\left(x_k^{(1)},\ldots,x_k^{(M)}\right)=\vartheta_k^{\tau}\phi\left(x_k^{(1)},\ldots,x_k^{(M)}\right). \tag{2.5}$$

Here $\hat{J}_k\left(x_k^{(1)},\ldots,x_k^{(M)}\right)$ denotes the approximation/estimation of $\mathcal{J}$ generated at the k-th time-step, ϕ denotes the nonlinear vector of L *regressor terms*, ϑ_k denotes the vector of *parameter estimates* calculated at the k-th time-instant and L is a positive user-defined integer denoting the *size* of the function approximator (2.5). The vector ϕ of regressor terms must be chosen so that it satisfies the so-called *Universal Approximation Property* (Polycarpou and Ioannou, 1991), i.e. it must be chosen so that the approximation accuracy of the approximator (2.5) is an increasing function of the approximator's size L. Polynomial approximators, radial basis functions, kernel-based approximators, etc, are known to satisfy such a property, see (Polycarpou and Ioannou, 1991) and the references therein.

The parameter estimation vector ϑ_k is calculated according to

$$\vartheta_k=\underset{\vartheta}{\operatorname{argmin}}\frac{1}{2}\sum_{\ell=\ell_k}^{k-1}\left(J_\ell^n-\vartheta^{\tau}\phi\left(x_\ell^{(1)},\ldots,x_\ell^{(M)}\right)\right)^2 \tag{2.6}$$

where $\ell_k=\max\{0,k-L-T_h\}$ with T_h being a user-defined nonnegative integer. Standard least-squares optimization algorithms can be used for the solution of (2.6).

Remark 1 *In order for the proposed methodology to guarantee with efficient performance, special attention has to be paid in the selection of the regressor vector ϕ. Polynomial or polynomial-like regressor vectors as well as sigmoidal regressor vectors can be employed for the construction of ϕ. The particular choice adopted for the application treated in this paper is described in section III. See (Kosmatopoulos et al., 2007, Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009) for more details on the design considerations for the regressor vector.* $\diamond$

As soon as the estimator $\hat{J}_k$ is constructed according to (2.5), (2.6), the set of new robots' positions/poses is selected as follows: firstly, a set of N candidate

robots' positions/poses is constructed according to[1]

$$x_k^{i,j} = x_k^{(i)} + \alpha_k \zeta_k^{i,j}, i \in \{1, \ldots, M\}, j \in \{1, \ldots, N\}\,, \tag{2.7}$$

where $\zeta_k^{i,j}$ is a zero-mean, unity-variance random vector with dimension equal to the dimension of $x_k^{(i)}$ and α_k is a positive real sequence which satisfies the conditions:

$$\lim_{k\to\infty} \alpha_k = 0, \quad \sum_{k=1}^{\infty} \alpha_k = \infty, \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty\,. \tag{2.8}$$

Among all N candidate new positions $x_k^{1,j}, \ldots, x_k^{M,j}$, the ones that correspond to non-feasible positions/poses – i.e., the ones that violate the constraints (2.3) – *are neglected* and then the new robots' positions/poses are calculated as follows:

$$\left[x_{k+1}^{(1)}, \ldots, x_{k+1}^{(M)}\right] = \underset{\substack{j \in \{1,\ldots,N\} \\ x_k^{i,j} \text{not neglected}}}{\operatorname{argmin}} \hat{J}_k\left(x_k^{1,j}, \ldots, x_k^{M,j}\right)$$

The idea behind the above logic is simple: at each time-instant a set of many candidate new robots' positions/poses is generated. The candidate, among the ones that provide with a feasible solution, that provides the best estimated value $\hat{J}_k$ of the coverage criterion is selected as the new set of robots' positions/poses. The random choice for the candidates is essential and crucial for the efficiency of the algorithm, as such a choice guarantees that $\hat{J}_k$ is a reliable and accurate estimate for the unknown function $\mathcal{J}$; see (Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009) for more details. On the other hand, the choice of a slowly decaying sequence α_k, a typical choice of adaptive gains in stochastic optimization algorithms (see e.g., (Bertsekas and Tsitsiklis, 2000)) is essential for filtering out the effects of the noise term ξ_k [cf. (2.2)]. The next theorem summarizes the properties of the CAO algorithm described above; as the proof of this theorem is among the same lines as the main results of (Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009), only a sketch of the proof is provided.

Theorem 1 *Let $x^{(1^*)}, \ldots, x^{(M^*)}$ denote any – local – minimum of the constrained optimization problem (2.4). Let $N \geq 2M \times \dim\left(x_k^{(i)}\right)$ and, moreover, the vector ϕ satisfy the Universal Approximation Property. Assume also that the functions*

[1] According to (Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009) it suffices to choose N to be any positive integer larger or equal to $2\times$[the number of variables being optimized by CAO]. In our case the variables optimized are the robot positions/poses $x_k^{(1)}, \ldots, x_k^{(M)}$ and thus it suffices for N to satisfy $N \geq 2M \times \dim\left(x_k^{(i)}\right)$.

$\mathcal{J}, \mathcal{C}$ are either continuous or discontinuous with a finite number of[2] discontinuities. Then, the CAO-based multi-robot coverage algorithm as described above guarantees that the robots' positions/poses $x_k^{(1)}, \ldots, x_k^{(M)}$ will converge to one of the local minima $x^{(1^*)}, \ldots, x^{(M^*)}$ almost surely, provided that the size L of the regressor vector ϕ is larger than a lower bound $\bar{L}$.

Sketch of the Proof: Let X_k denotes the augmented vector of all robots' positions/poses at time-instant k, i.e., the entries of X_k are the entries of all vectors $x_k^{(1)}, \ldots, x_k^{(M)}$. Using similar arguments as those of the main results of (Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009) it can be seen that at each iteration of the CAO-based algorithm described above, the new vector X_{k+1} satisfies

$$X_{k+1} = \Pi_C \left\{ X_k - \alpha_k \left(c \nabla J(X_k) + e_k + b_k \right) \right\}$$

where c is a positive constant; $\Pi_C\{\cdot\}$ denotes the projection operator onto the set $G = \{X : \mathcal{C}(X) \leq 0\}$ defined as follows: for any X not satisfying the constraint $\mathcal{C}(X) \leq 0$, the point $\tilde{X} = \Pi_C\{X\}$ is the nearest point to X on G, where the norm is defined in the usual Euclidean norm; and e_k, b_k are two terms that are defined similarly to the respective terms in section III of (Spall, 1992). By using the same arguments as those in the proof of Proposition 1 of (Sadegh, 1997), it can be established that the above equation converges almost surely to one of the local minima of the constraint minimization problem (2.4).

Remark 2 *Strictly speaking, Theorem 1 is valid as long as the zero-mean, unity variance vectors $\zeta_k^{i,j}$ satisfy some extra technical conditions (which are satisfied if e.g., $\zeta_k^{i,j}$ are Bernoulli random vectors). However, extensive simulation investigations have shown that, in practice, Theorem 1 is still valid even if the random vectors $\zeta_k^{i,j}$ are Gaussian random vectors, despite the fact that such a choice does not satisfy the aforementioned technical conditions.* $\diamond$

Remark 3 *As already noticed in section I, the CAO algorithm requires only a local approximation of the unknown function $\mathcal{J}$ and as a result the lower bound $\bar{L}$ has not to be large (as opposed to methods that construct a global approximation of the unknown function $\mathcal{J}$). Although, there exist no theoretical results for providing the lower bound $\bar{L}$ for the size of the regressor vector ϕ, practical investigations on many different problems indicate that for the choice of the regressor vectors according to Remark 1 such a bound is $2\times$[number of variables being optimized by CAO]; see (Kosmatopoulos et al., 2007, Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009) for more details.* $\diamond$

[2]Please note that the family of "discontinuous functions with a finite number discontinuities" corresponds to the family of functions that can be approximated with arbitrary accuracy by continuous ones (Jin et al., 1995). For instance, terrains with discontinuities along e.g., a closed or open curve belong to this family of functions and so do the corresponding functions $\mathcal{J}$ and $\mathcal{C}$.

Remark 4 *As an alternative to the CAO approach, the SPSA approach (Spall, 1992) may be employed in multi-robot coverage applications. According to the SPSA approach, the robot positions/poses are updated according to*

$$\begin{cases} x_{k+1}^{(i)} = x_k^{(i)} + \beta_k \zeta_k^i \,, & \text{if } k \text{ is even} \\ x_{k+1}^{(i)} = x_k^{(i)} + \gamma_k \frac{J_k^n - J_{k-1}^n}{\zeta_{k-1}^{(i)}} \,, & \text{if } k \text{ is odd} \end{cases} \tag{2.9}$$

where $\zeta_k^{(i)}$ are zero-mean, unity-variance random vectors and β_k, γ_k are slowly decaying sequences (similar as the sequence α_k). The SPSA algorithm is computationally simpler than the CAO one, but it does not perform as efficient as the CAO approach as have been demonstrated in a variety of approaches, see (Kosmatopoulos et al., 2007, Kosmatopoulos, 2009, Kosmatopoulos and Kouvelas, 2009). However, extensive simulation experiments have demonstrated that a hybrid scheme which uses SPSA at the first 10-20 time-steps and then switches to the CAO algorithm can have significant improvements over schemes that employ only the CAO algorithm. This is due to the fact that CAO, at its initial steps, may preserve a poor performance because it takes some iterations for the CAO estimator (2.5) in order to come up with a reliable estimate $\hat{J}_k$ of the unknown coverage function $\mathcal{J}$. ◇

Remark 5 *We close this section by mentioning that similarly to the proposed approach, global optimization methods such as simulated annealing and genetic algorithms do not require that the explicit form of the function $\mathcal{J}$ is known. However, simulated annealing, genetic algorithms and other similar global optimization methods require that a large amount of different combinations of robots' positions is being evaluated all over the robots' application area. Such a requirement renders these methods practically infeasible as a huge amount of time and energy would have to be spent in order for the robots to visit many different locations all over their application area. Nevertheless, attempting to globally optimize surveillance coverage is practically infeasible as it is an NP-hard problem whose solution requires dense discretization over the space of all possible team configurations and evaluation of all points of the discretized space.* ◇

III CAO for 3D Multi - Robot Coverage over Unknown Terrains

III.1 Problem Definition

In our previous work (Renzaglia et al., 2010, 2011) we have extensively described the case of using the CAO approach for maximizing the monitored area in a given

region by using a team of mobile robots in the 2D plane, without any assumption on the topology of the environment. In this section we will extend our approach to the case of robots living in a 3D environment and having fixed orientation.

Consider a team of M flying robots that is deployed to monitor an unknown terrain $\mathcal{T}$. Let z denote the unknown height of the terrain at the point (x, y) and assume for simplicity that the terrain $\mathcal{T}$ is rectangular along the (x, y)-axes, i.e., $x_{min} \leq x \leq x_{max}, y_{min} \leq y \leq y_{max}$. Let $\mathcal{P} = \{x^{(i)}\}_{i=1}^{M}$ denote the configuration of the robot team, where $x^{(i)}$ denotes the position of the i-th robot. We will say that a point $q = (x, y, z)$ of the terrain is visible if there exists at least one robot so that

- the robot and the point q are connected by a line-of-sight;
- the robot and the point q are at a distance smaller than a given threshold value (defined as the maximum distance the robot's sensor can "see").

Given a particular team configuration $\mathcal{P}$, we let $\mathcal{V}$ denote the *visible area* of the terrain, i.e., $\mathcal{V}$ consists of all points $q \in \mathcal{T}$ that are visible from the robots. We will assume that the robots are equipped with visual sensors together with inertial sensors and/or range sensors; in other words, for each visible point we will assume that the team is able to estimate the terrain's height at this point. A possible way to deploy a robot team satisfying the above, is by using the down-looking-camera-equipped flying robots of (Bloesch et al., 2010, Weiss et al., 2010) which employ the monocular SLAM algorithm of (Klein and Murray, 2007).

The main objective for the robot team is to maximize the visible area $\mathcal{V}$. However, this cannot be the only objective for a robot team in a coverage task: trying to maximize the visible area will simply force the robots to climb as high as[3] their visibility threshold allows for. For this reason, while maximizing the visible area is the primary goal of the mission, the team members should be deployed so that for every point in the terrain, the closest robot is as close as possible to that point. In other words, *among all possible configurations that maximize the visible area $\mathcal{V}$, the robot team should converge to the one that keeps as small as possible the average distance between each of the robots and the terrain subarea the particular robot is responsible for,* where the subarea of the terrain the i-th robot is responsible for is defined as the part of the terrain that (a) is visible by the i-th robot and (b) each point in this subarea is closer to the i-th robot than any other robot of the team. Attempting to keep the robots as close as possible to the subarea they are responsible for is necessary for two practical reasons: (a) firstly, the closer is the robot to a point in the terrain the better is, in general, its sensing ability to monitor

[3]Note also that in the ideal case where there are no limits for the robot's maximum height and the robot has unlimited sensing capabilities, it suffices to have a single robot at a very high position to monitor the whole terrain.

this point and (b) secondly, in many multi-robot coverage applications there is the necessity of being able to intervene as fast as possible in any of the points of the terrain with at least one robot.

Having the above reasoning in mind, we define the following combined objective function the robot team has to minimize:

$$J(\mathcal{P}) = \int_{q\in\mathcal{V}} \min_{i\in\{1,\ldots,M\}} \|x^{(i)} - q\|^2 dq + K \int_{q\in\mathcal{T}-\mathcal{V}} dq \tag{3.10}$$

where K is a user-defined positive constant and $\|\cdot\|$ denote the Euclidean norm. The first of the terms in above equation is the usual cost function considered in many coverage problem for known 2D environment related to the second objective (minimize the average distance between the robots and the subarea they are responsible for, see (Cortés et al., 2004)). The second term is related to the invisible area in the terrain ($\int_{q\in\mathcal{T}-\mathcal{V}} dq$ is the total part of the terrain that is not visible by any of the robots).

The positive constant K serves as a weight for giving less or more priority to one of the objectives. For instance, in the case where $K = 0$, we will have that the robots, in their attempt to minimize their average distance to the subarea they are responsible for, may also seek to minimize the total visible area. It has to be emphasized that the positive constant K should be chosen sufficiently large so that the second term in (3.10) dominates the first term unless no or a negligible part of the terrain remains invisible. In this way, minimization of (3.10) is equivalent to firstly making sure that all, or almost all, of the terrain is visible and then to locate the robots so that their average distance to the subarea they are responsible for is minimized. However, choosing a value for K so that the second term in (3.10) dominates the first term is not straightforward unless the terrain is known. Later in this section we will comment further on how to choose the parameter K for the particular setup considered in this paper. Please also note that whether the CAO-based algorithm employing a large K converges to negligible or non-negligible invisible areas depends on the number, the sensing capabilities and maximum height constraints of the robots as well as the terrain's complexity.

The second term $\int_{q\in\mathcal{T}-\mathcal{V}} dq$ in (3.10) cannot be, in general, computed in practice; as this term involves the part of the terrain that is not currently visible, its computation requires that the geometry of this part is known or equivalently – as the invisible part changes with the evolution of the team's configuration – that the whole terrain is known. To overcome this problem, instead of minimizing (3.10) the following performance index is actually minimized by the CAO approach:

$$\bar{J}(\mathcal{P}) \;=\; \int_{q\in\mathcal{V}} \min_{i\in\{1,\ldots,M\}} \|x^{(i)} - q\|^2 dq - K \int_{q\in\mathcal{V}} dq \tag{3.11}$$

To see that minimization of (3.11) and (3.10) is equivalent, please note that $\int_{q\in\mathcal{T}-\mathcal{V}} dq = \int_{q\in\mathcal{T}} dq - \int_{q\in\mathcal{V}} dq$ and the integral $\int_{q\in\mathcal{T}} dq$ is constant.

Throughout the above analysis, the assumption that K is "sufficiently large" was made. If such an assumption does not hold, the arguments presented above do not hold either. Therefore, the question that naturally arises is what is a value for K that is sufficiently large so that these arguments hold. As a very large choice for K (e.g., $K = 10^{10}$) can lead to numerical instability problems [switching-like performance for the algorithm when the numerically-computed invisible area switches from small values to zero], a guide on how to choose K so that such instability problems do not occur should be provided. Extensive simulations with all set-ups considered in the next section (IV. Simulation Results) and with different values for K indicate that it suffices to choose K to be $3-50$ times the parameter $\hbar_{max}$ in order to get an efficient performance, where the parameter $\hbar_{max}$ can be calculated as follows: let

$$\begin{aligned} f(\mathcal{P}) &= \int_{q\in\mathcal{V}} \min_{i\in\{1,\ldots,M\}} \|x^{(i)} - q\|^2 dq, \\ \bar{g}(\mathcal{P}) &= \int_{q\in\mathcal{T}-\mathcal{V}} \mathcal{I}(x,y)dxdy \end{aligned}$$

where $\mathcal{I}(x,y)$ denotes the indicator function that is equal to 1 if the point q belongs to the invisible area of the terrain and is zero, otherwise. In other words, the term $\bar{g}(\mathcal{P})$ would correspond to the total invisible area, if the unknown terrain points (x,y,z) were replaced by $(x,y,1)$, i.e., if the whole invisible area were flat. Then the parameter $\hbar_{max}$ is calculated according to

$$\hbar_{max} \approx \frac{\sup f(\mathcal{P})}{\sup \bar{g}(\mathcal{P})}$$

i.e., $\hbar_{max}$ corresponds to the maximum possible value for $f(\mathcal{P})$ (over all possible feasible team configurations) divided by the maximum value the invisible-area-term $\bar{g}(\mathcal{P})$ can take. The $\sup \bar{g}(\mathcal{P})$ is equal to the terrain's area along the (x,y)-axes, i.e. $\sup \bar{g}(\mathcal{P}) = (x_{max} - x_{min})(y_{max} - y_{min})$. On the other hand an estimate of the term $\sup f(\mathcal{P})$ can be produced by running extensive simulations with randomly generated terrains and randomly generated team's configurations. Figure 1 shows the time-histories of the terms $f(\mathcal{P})$ and $\bar{g}(\mathcal{P})$ for different choices for K and for one of the scenarios described in the simulations section (more precisely, for the scenario described in section IV.2.2). For this particular scenario, we have that $\sup f(\mathcal{P}) \approx 1000$ and $\sup \bar{g}(\mathcal{P}) = 100$ and thus the parameter $\hbar_{max}$ can be estimated to be around 10. As exhibited in Figure 1, the CAO-based algorithm converges to negligible invisible areas for all values of K satisfying $K \in [3\hbar_{max}, 50\hbar_{max}] \equiv [30, 500]$.

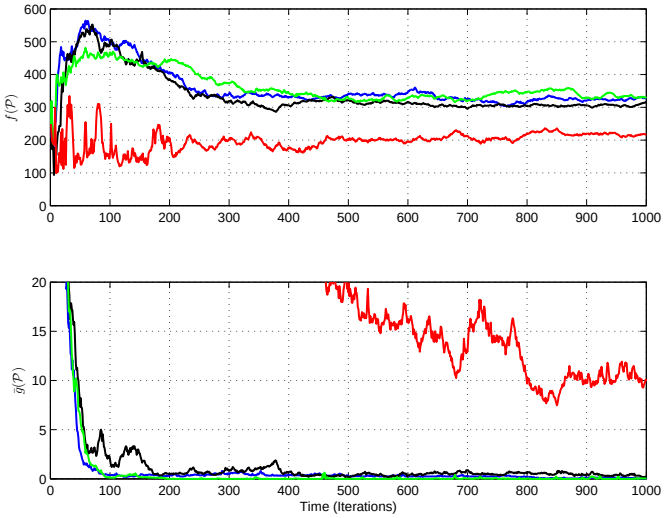


Figure 1: Time-histories for the terms $f(\mathcal{P}) = \int_{q\in\mathcal{V}} \min_{i\in\{1,\ldots,M\}} \|x^{(i)} - q\|^2 dq$ (upper plot) and $\bar{g}(\mathcal{P}) = \int_{q\in\mathcal{T}-\mathcal{V}} \mathcal{I}(x,y)dxdy$ (lower plot) for different values of the parameter K: $K = 5$ (red), $K = 30$ (black), $K = 100$ (blue) and $K = 500$ (green).

III.2 Algorithm Implementation

An efficient trajectory generation algorithm for optimal coverage must make sure that the physical constraints are also met throughout the whole multi-robot coverage application. Such physical constraints include, but are not limited to, the following ones:

- The robots remain within the terrain's limits, i.e., within $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$ in the $x-$ and y-axes, respectively.
- The robots satisfy a maximum height requirement while they do not hit the terrain, i.e., they remain within $[z + d, z_{max}]$ along the z-axis, where d denotes the minimum safety distance (along the z-axis) the robots' should be from the terrain and z_{max} denotes the maximum allowable height for the robots.
- The robots do not come closer to the other ones than a minimum allowable safety distance d_r.

It is not difficult for someone to see that all the above constraints can be easily cast in the form (2.3) and thus can be handled by the CAO algorithm.

Having defined the optimization problem, a fundamental point for a good behavior of the CAO algorithm is an appropriate choice of the form of the regressor

vector ϕ, introduced in equation (2.5). As mentioned in remark 1, several different choices for its explicit expression are admissible. However, in all different tests for the particular application treated in this paper, it was found that it suffices to choose the regressor vector as follows:

1. choose the size of the function approximator L to be an odd number;
2. select the first term of the regressor vector ϕ to be the constant term;
3. select randomly the next $(L-1)/2$ terms of ϕ to be any 2nd-order terms of the form $x_a^{(i)} \cdot x_b^{(j)}$ [with $a, b \in \{1, \ldots, \dim(x^{(i)})\}, i, j \in \{1, \ldots, M\}$ randomly-selected positive integers];
4. select the last $(L-1)/2$ terms of ϕ to be any 3rd-order terms of the form $x_a^{(i)} \cdot x_b^{(k)} \cdot x_c^{(j)}$ [with $a, b, c \in \{1, \ldots, \dim(x^{(i)})\}, i, k, j \in \{1, \ldots, M\}$ randomly-selected positive integers].

Once the regressor vector ϕ has been set and once the values of the cost function (3.11) are available for measurement at each time step, it is possible to find at each time step the vector of parameter estimates θ_k and thus the approximation of the cost function $\hat{J}_k$. The other important choice in order to assure the convergence of the algorithm is the expression of the sequence α_k, defined in equation (2.7). A typical choice for such a sequence is given by

$$\alpha_k = \frac{\alpha}{(k+1)^\eta}\,, \tag{3.12}$$

where α is a positive user-defined constant and $\eta \in (0, 0.5)$.

Remark 6 *Please note that the CAO algorithm's computational requirements are dominated by the requirement for solving the least-squares problem (2.6). As the number of free parameters in this optimization problem is L, most popular algorithms for solving least-squares problems have, in the worst case, $\mathcal{O}(L^3)$ complexity (polynomial complexity with respect to L). Furthermore, as it suffices to choose L around $2M \times \dim\left(x_k^{(i)}\right)$ (see Remark 3) we have that the computational requirements of the proposed algorithm are, at most, $\mathcal{O}(M^3)$.* $\diamond$

IV Simulation Results

To evaluate the efficiency of the proposed approach, several scenarios were considered with the use of a simulated robot team which was able to move freely at the 3D plane. In all cases studied, the team was homogeneous with exactly the same

monitoring capabilities. This assumption has been made for simplification purposes and easier comprehension of the results. The main constraints imposed to the robots are that they remain within the terrain's limits, i.e., within $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$ in the $x-$ and y-axes, respectively. At the same time they have to satisfy a maximum height requirement while they do not hit the terrain, i.e., they remain within $[z+d, z_{max}]$ along the z-axis. The scenarios considered are terrains with obstacles with same or uneven heights, while for each scenario different values of the parameter α which is responsible for the convergence of the algorithm were tested. Apart from the simulated terrains, a realistic scenario was considered by using a map of a real area (Weiss et al., 2010), extracted with the methodology described in detail in (Bloesch et al., 2010).

In all experiments reported next, the following choices were made for the algorithm's implementation:

- The CAO parameters N (number of next candidate robots' positions) and L (size of approximator ϕ) were set equal to $6M$ and $6M+1$, respectively, where M denotes the robot team's size, while the approximator ϕ was calculated as described in the previous section. Please note that the above choices for N, L and ϕ are in accordance to Theorem 1 and Remark 3; moreover, L was set equal to $6M+1$ as it has to be an odd number.

- The parameter K in the cost criterion (3.11) was set equal to 30 which satisfies $K \in [10\hbar_{max}, 50\hbar_{max}]$ for all terrains and team sizes considered in the simulations (see section III.1 for more details on the parameter $\hbar_{max}$).

- The parameter d (minimum allowable distance from the terrain) was set equal to 0.1, while the robot's were assumed to have unlimited visibility.

- Different choices for the parameters z_{max} (maximum allowable height) and α (magnitude of next candidate robots' positions) were made as these parameters are the most crucial for the algorithm's performance.

- Finally, with the exception of the experiments reported in IV.4 that involve teams of 10 and 20 robots, in all other cases the team comprised 4 robots.

IV.1 Areas with same height obstacles

The first case considered, studies an area sizes 10 by 10 meters, which includes a surface with seven same height randomly placed obstacles. For this area, several scenarios were tested with the robot team starting from different initial positions and heights. In all cases considered the robots had to satisfy a maximum flight height requirement while they did not hit the terrain.

IV.1.1 Scenario 1

In the first scenario studied, all the team members were placed at starting points adjunct to each other, with initial height 0.6 meters. The maximum allowed flight height was 1 meter for all robots. Different values of the expression α were tested for the case of $\alpha = 0.3, 0.5, 1$ and the respective cost functions are presented in Fig. 2. The initial position for *Robot 1* was $(0.18, 0.2, 0.6)$, of *Robot 2* was $(0.19, 0.2, 0.6)$, of *Robot 3* was $(0.2, 0.2, 0.6)$ and of *Robot 4* was $(0.21, 0.2, 0.6)$. In Fig. 3 successive snapshots of different positions of the robot team for the case of $\alpha = 0.3$ are presented (different color corresponds to different team member). The final configuration in all three test cases is presented in Fig. 4. In Table 1 the percentage of the initial and final coverage of the area monitored in all three cases, is presented. It's worth mentioning that the coverage percentage is depended on several factors apart the optimization algorithm i.e. the sensors that might be used in a real implementation. It should be noted that CAO does not converge always to the same swarm configuration, but it converges always to a swarm configuration with similar coverage characteristics which corresponds to similar final J value.

Table 1: Coverage percentage in the case described in IV.1.1.

	(% of Coverage)		
α	0.3	0.5	1
Initial Configuration	34.58		
Final Configuration	97.06	97.49	98.59

IV.1.2 Scenario 2

In the second scenario studied, the initial positions of the robots were for *Robot 1* $(9.18, 0.2, 0.4)$, for *Robot 2* $(9.19, 0.2, 0.4)$, for *Robot 3* was $(0.2, 0.2, 0.4)$ and for *Robot 4* was $(0.21, 0.2, 0.4)$, while the maximum allowed flight height remained the same (1 meter). Different values of the expression α were tested for the case of $\alpha = 0.3, 0.5$ and the respective cost functions are presented in Fig. 6. In table 2 the percentage of the initial and final coverage of the area monitored in both cases is presented.

Table 2: Coverage percentage in the case described in IV.1.2.

	(% of Coverage)	
α	0.3	0.5
Initial Configuration	48.71	
Final Configuration	98.56	97.04

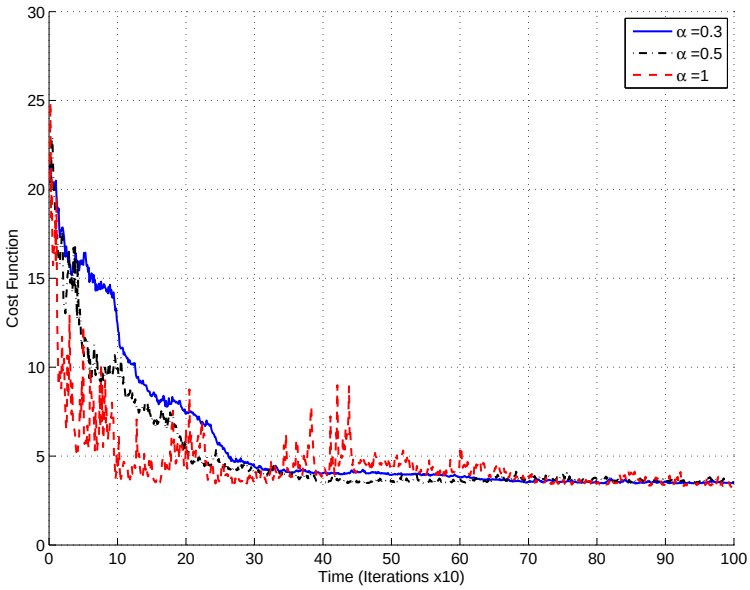


Figure 2: Cost Functions for $\alpha = 0.3, 0.5, 1$, in the case described in IV.1.1.

IV.2 Areas with uneven obstacle height

The second case considered, studies an area sizes 10 by 10 meters, which includes a surface with seven randomly placed obstacles with uneven height, with maximum value 2 meters. In this test case we have tested several scenarios with the robot team starting from different initial positions and heights. In all cases considered the robots had to satisfy a maximum flight height requirement while they did not hit the terrain.

IV.2.1 Scenario 1

In the first scenario studied for the case of areas with uneven obstacle heights, all the team members were placed at starting points adjunct to each other, with initial height 0.2 meters. The maximum allowed flight height was 1 meter for all robots. The initial positions of *Robot 1* was $(0.18, 0.2, 0.2)$, of *Robot 2* was $(0.19, 0.2, 0.2)$, of *Robot 3* was $(0.2, 0.2, 0.2)$ and of *Robot 4* was $(0.21, 0.2, 0.2)$. Different values of the expression α were tested for the case of $\alpha = 0.3, 0.5, 1$ and the respective cost functions are presented in Fig. 7. The final configuration in all three test cases is presented in Fig. 8, while in Table 3 the percentage of the initial and final coverage of the area monitored in all cases, is presented.

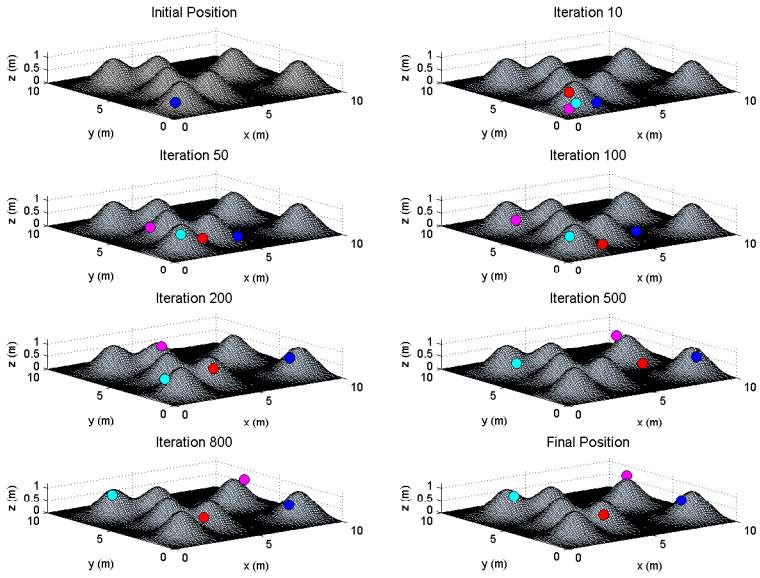


Figure 3: Successive snapshots of different positions of the robot team for $\alpha = 0.3$, in the case described in IV.1.1.

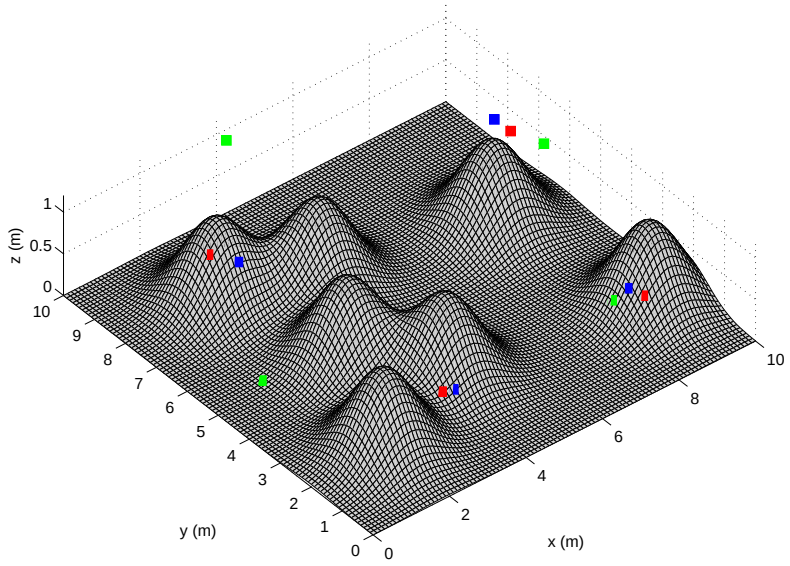


Figure 4: Final positions of the robotic teams for $\alpha = 0.3$ (blue markers), $\alpha = 0.5$ (red markers), $\alpha = 1$ (green markers), in the case described in IV.1.1.

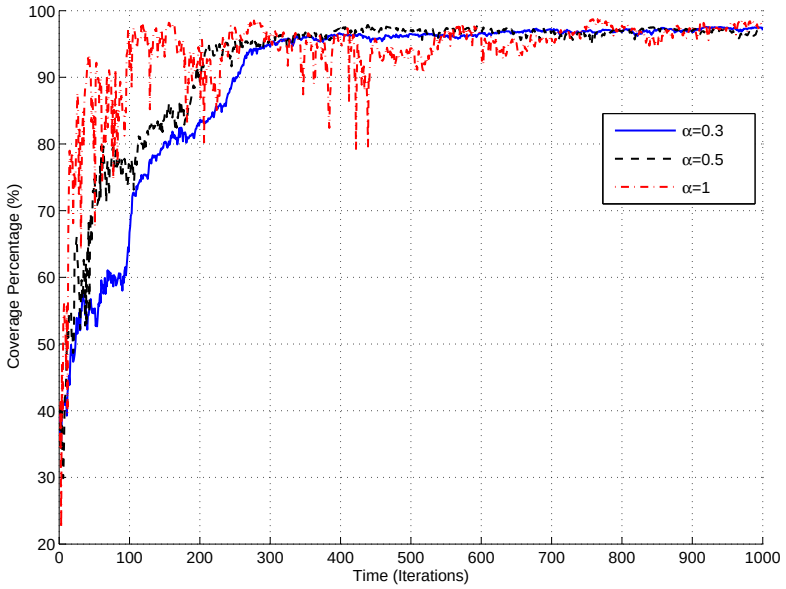


Figure 5: Coverage percentage for $\alpha = 0.3, 0.5, 1$, in the case described in IV.1.1.

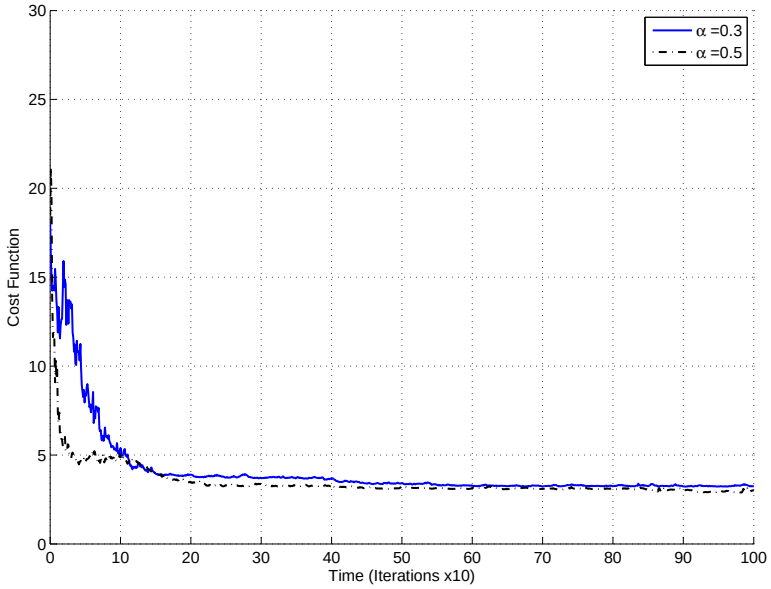


Figure 6: Cost Functions for $\alpha = 0.3, 0.5$, in the case described in IV.1.2.

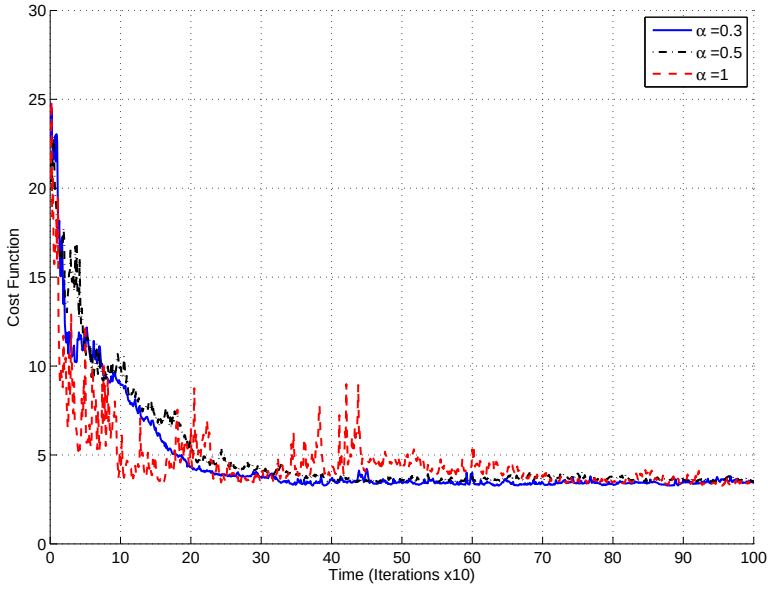


Figure 7: Cost Functions for $\alpha = 0.3, 0.5, 1$, in the case described in IV.2.1.

Table 3: Coverage percentage in the case described in IV.2.1.

	(% of Coverage)		
α	0.3	0.5	1
Initial Configuration	29.78		
Final Configuration	98.29	97.76	96.35

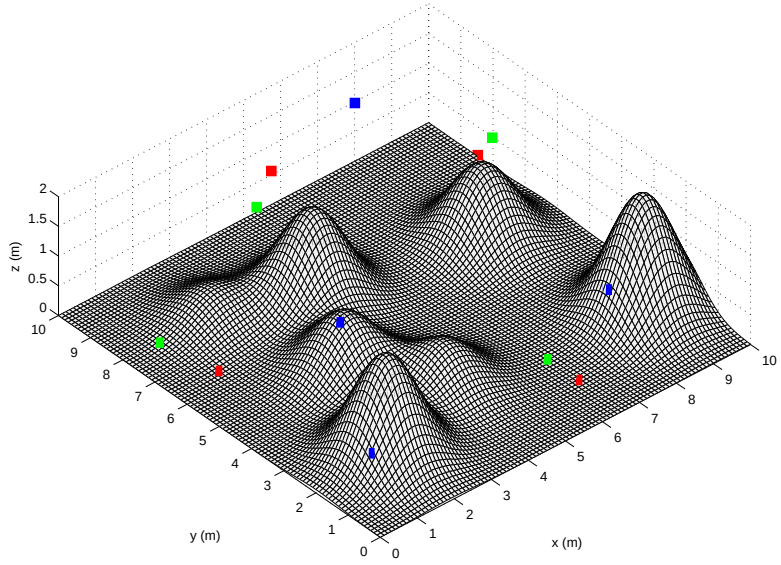


Figure 8: Final positions of the robotic teams for $\alpha = 0.3$ (blue markers), $\alpha = 0.5$ (red markers), $\alpha = 1$ (green markers), in the case described in IV.2.1.

IV.2.2 Scenario 2

In the second scenario studied for uneven surfaces, the maximum allowed flight height was 5 meters for all robots. Different values of the expression α were tested for the case of $\alpha = 0.3, 0.5, 1$ and the respective cost functions are presented in Fig. 9. In Fig. 10 successive snapshots of different positions of the robot team for the case of $\alpha = 0.3$ are presented (different color corresponds to different team member). The initial position of *Robot 1* was $(0.18, 0.2, 0.4)$, of *Robot 2* was $(0.19, 0.2, 0.4)$, of *Robot 3* was $(0.2, 0.2, 0.4)$ and of *Robot 4* was $(0.21, 0.2, 0.4)$. In table 4 the percentage of the initial and final coverage of the area monitored in all cases, is presented.

Table 4: Coverage percentage in the case described in IV.2.2.

	(% of Coverage)		
α	0.3	0.5	1
Initial Configuration	29.78		
Final Configuration	99.14	98.69	98.36

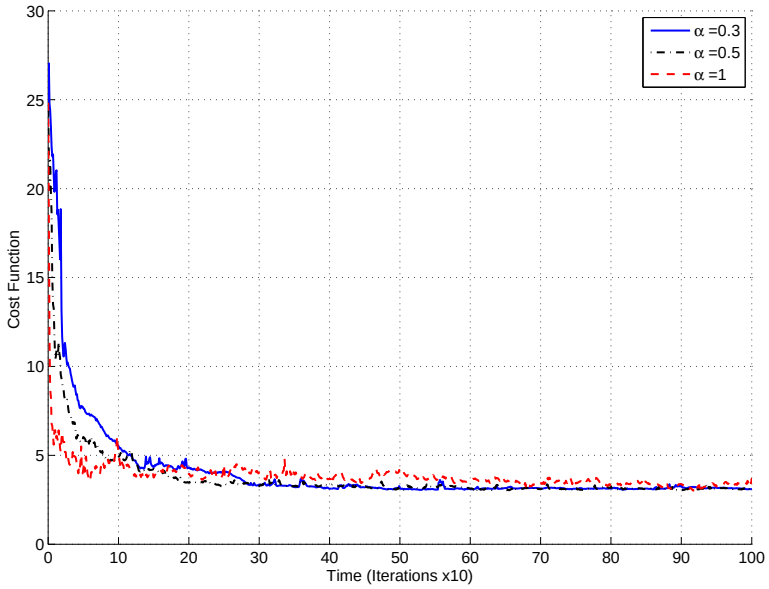


Figure 9: Cost Functions for $\alpha = 0.3, 0.5, 1$, in the case described in IV.2.2.

IV.3 Cave-like Surface

In the above described set-up and proposed simulations, for simplicity's sake we assumed that the unknown terrain is defined as a set of unique triplets (x, y, z), that is, for each (x, y) the terrain is defined by a unique z-point, i.e. $z = f(x, y)$. In realistic applications there exist cases where there may be more than one z-points (e.g., cases of terrains with buildings, overhangs, ledges, caves, etc). Here we present a similar scenario to show how our method can be applied also for these cases. The simulated environment is a gaussian with a cave (fig. 11). The robots start their mission on the other side with respect to the cave, so at the beginning the it is not visible. Start and final robots' positions are shown in fig. 12. In fig. 13(a) the behavior of the cost function is presented and in fig. 13(b) it is shown also the percentage of the invisible surface during the task. It is possible to see

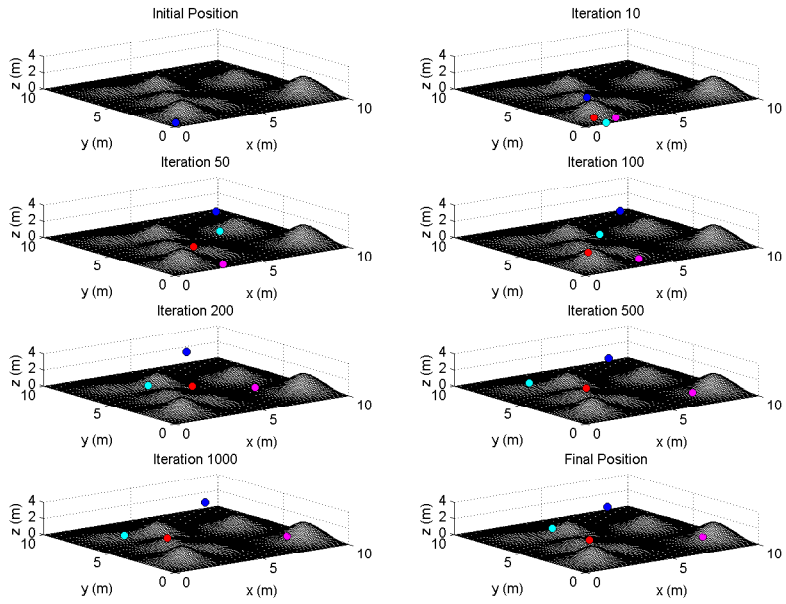


Figure 10: Successive snapshots of different positions of the robot team for $\alpha = 0.5$, in the case described in IV.2.2.

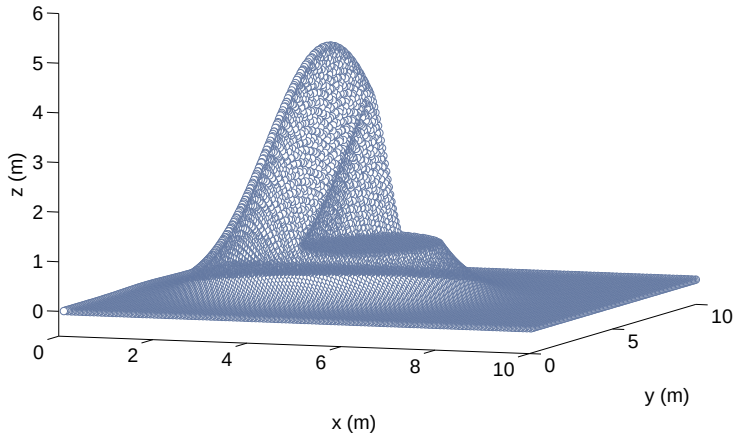


Figure 11: A different scenario: the environment is a gaussian with a cave.

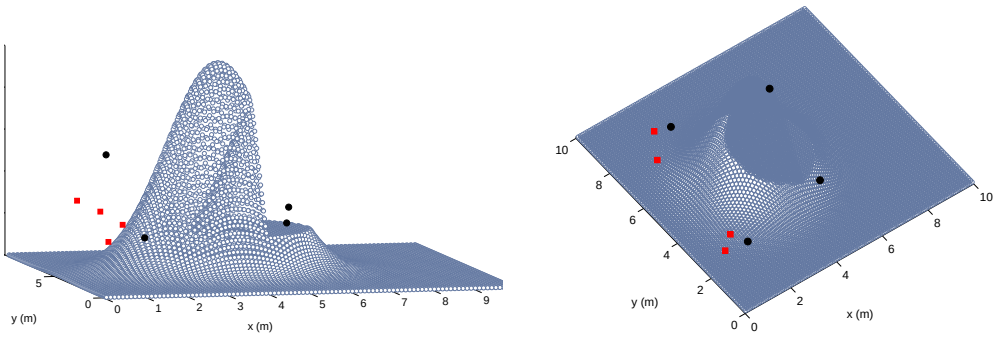
that it is minimized until everything is visible.

IV.4 Scalability Issues

To validate the efficiency of our approach in the case of bigger robot teams, we have performed experiments with teams consisting of 10 and 20 members. Our experiments were performed in an area sizes 20 by 20 meters, which includes a surface with fifteen uneven height randomly placed obstacles. The maximum flight height was set to be 2. The basic difference as far as it concerns the computational requirements in the experiments conducted with the teams of 10 and 20 robots, was that the parameters L and N increase linearly according to Theorem 1 and Remark 3; therefore $L = 61$ and $N = 60$ in the case of the team with 10 members and $L = 121$ and $N = 120$ in the case of the team with 20 members. In the case of 4 robots the best values of $\mathcal{J}$ are around 15 which is significantly larger than the values obtained with the bigger teams. In Fig. 14 the cost functions for the case of 10 and 20 robots are presented, while in Fig. 15 we present their final configuration. In table 5 the percentage of the initial and final coverage of the area monitored for a team with 10 & 20 members is presented.

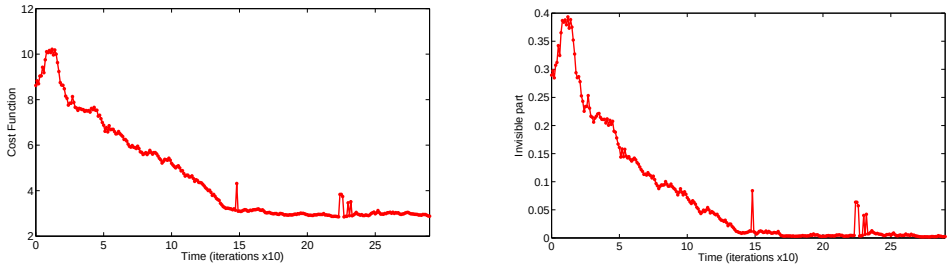
IV.5 Birmensdorf area

To validate our approach in a realistic environment, we used the data which were collected with the use of a miniature quadrotor helicopter specially designed for



(a) (b)

Figure 12: The team is composed by four robots. Red squares and black circles represent initial and final positions respectively.



(a) (b)

Figure 13: Fig. (a) shows the behavior of the cost function during the task, fig. (b) the percentage of invisible surface. At the end everything is visible.

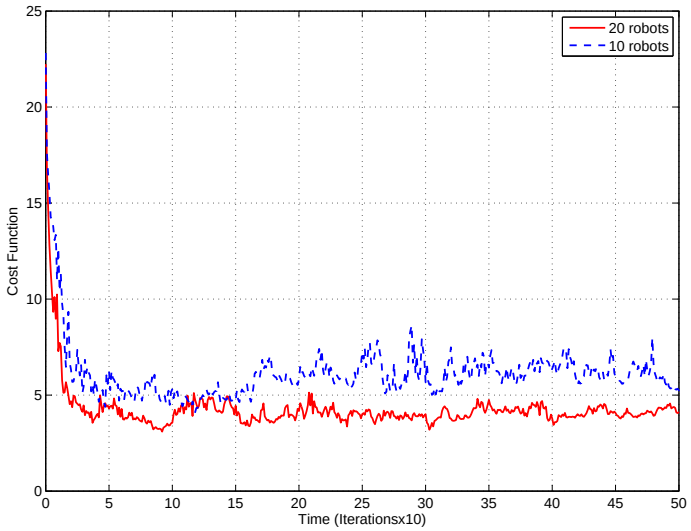


Figure 14: Comparative cost functions for the case of a robot team with 10 and 20 members.

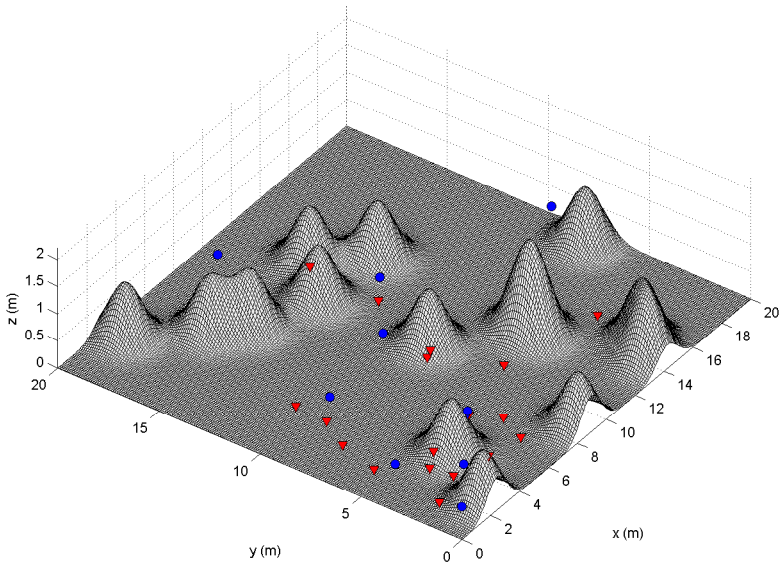


Figure 15: Final configuration of the team with 10 robots (blue circle markers) and the team with 20 robots (red triangle markers).

Table 5: Coverage percentage with teams of 10 & 20 members.

	(% of Coverage)	
Team size	10 Members	20 Members
Initial Configuration	39.62	42.93
Final Configuration	86.33	90.78

the needs of the European project sFLY (www.sfly.org). These data correspond to the Birmensdorf area presented in Fig. 16. The area was mapped using a state-of-the-art visual-SLAM algorithm which tracks the pose of the camera while, simultaneously, autonomously, building an incremental map of the surrounding environment. The area mapped using the aforementioned methodology corresponds to the bounded part of Fig. 16. More details about the data and the methodology used, are presented in (Bloesch et al., 2010) and (Weiss et al., 2010). The main constraints imposed to the robots are that they remain within the terrain's limits, i.e., within $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$ in the $x-$ and y-axes, respectively. At the same time they have to satisfy a maximum height requirement while they do not hit the terrain, i.e., they remain within $[z + d, z_{max}]$ along the z-axis. The value of α was equal to 0.3. Several initial configurations for the robot team were tested. In Fig. 17 the cost function of an illustrative scenario is presented, while the final configuration of the team (for the same scenario) is displayed in Fig. 18 (3D view) and in Fig. 19 (2D side view). The initial coverage was 44.49% while the final coverage was 98.55%. In table 6 the final coverage percentage for different initial configuration in the Birmensdorf area, is presented.

Table 6: Coverage percentage for different initial configuration in the Birmensdorf area.

(% of Coverage)					
Test Case	1	2	3	4	5
Initial Configuration	44.49	40.49	21.41	57.88	56.81
Final Configuration	98.55	99.52	98.53	98.94	99.56

V Discussion and Conclusions

A new method for dealing with the problem of performing surveillance coverage in *unknown* terrain of complex and non-convex morphology has been proposed. The proposed approach has the following key advantages:

- it does not require any a priori knowledge on the environment;

Figure 16: Outdoor flight path through the Birmensdorf area.

- it works in any given environment, without the necessity to make any kind of assumption about its topology;
- it can incorporate any kind of constraints;
- it does not require a knowledge about these constraints since they are learnt during the task execution;
- its complexity is low allowing real time implementations.

The advantages of the proposed methodology make it suitable for real implementations and the results obtained through numerical simulations give us the motivation to adopt the CAO also in other frameworks. We are interested into formulating the same problem in a distributed manner by using different cost functions for the robots in the team. This approach is closer to real world applications since it will not depend into a centralized scheme with all the known disadvantages. Apart from that a decentralized approach will allow us to include communications constraints. We are also interested in incorporating more realistic constraints including sensor limitations. Our aim is to develop a strategy for the surveillance of an unknown urban-like environment with a real MAV swarm. Furthermore, we are interested in enhancing the proposed methodology in order to be able to deal with cases where the team converges to a configuration that fails to cover the entire terrain. It is worth noticing, that although in the $3D$ case treated

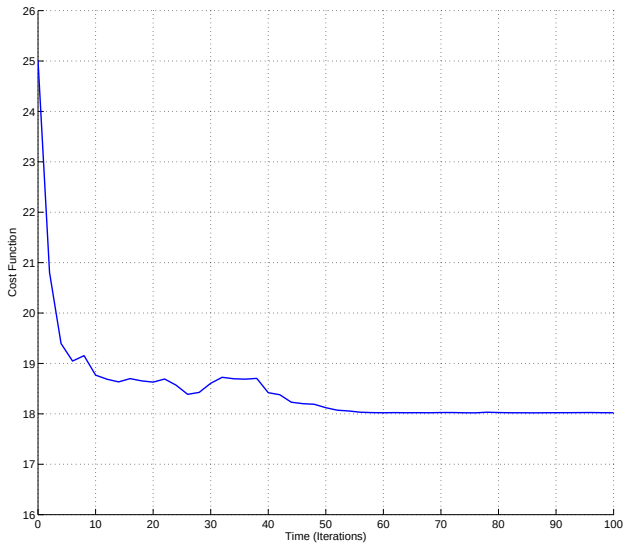


Figure 17: Cost function for a robotic team performing surveillance coverage in Birmensdorf area.

here we have never encountered such cases there is always the possibility (due to the local convergence properties of the algorithm) for the methodology to fail to cover the overall area. As a matter of fact, such cases have been encountered in the $2D$ version of the algorithm reported in (Renzaglia et al., 2010, 2011) and an extension/enhancement of the proposed methodology is required to efficiently deal with such cases.

Furthermore, we expect that many important tasks in mobile robotics can be approached by CAO-based algorithms: for example coordinated exploration, optimal target tracking, multi-robot localization, and so on. This is basically due to the fact that the CAO approach does not require an a priori knowledge of the environment and it has low complexity. Both these issues are fundamental in mobile robotics.

Acknowledgments

The research leading to these results has received funding from the European Communities Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 231855 (sFly). The authors would like to acknowledge S.Weiss and D. Scaramuzza from ETHZ/ASL for fruitful discussions and for providing the data set from Birmensdorf area. The authors would like to thank S. A. Chatzichristofis for his help in code optimization for the experimental results. The authors would

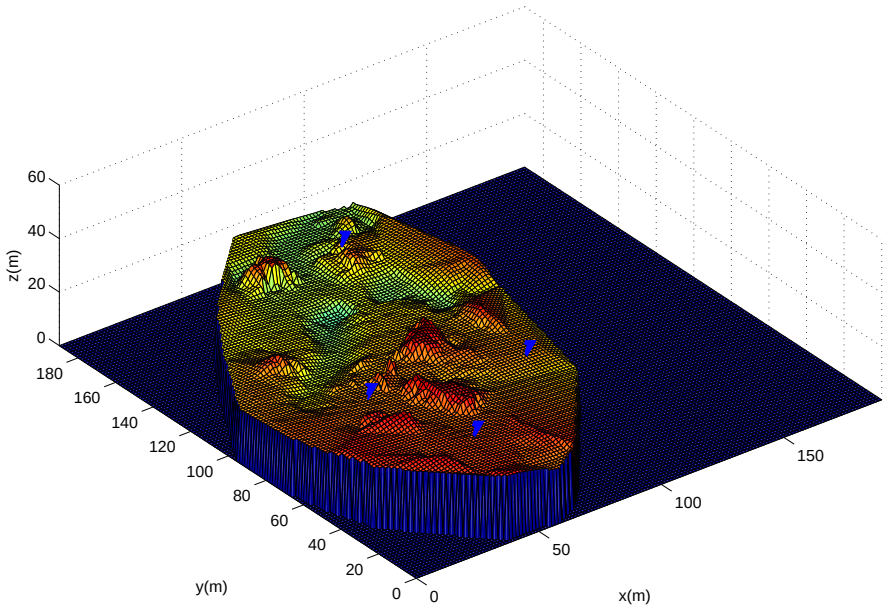


Figure 18: Final configuration of a robot team in a coverage scenario in Birmensdorf area (3D View).

also like to thank the reviewers for their constructive comments and criticism.

References

P.K. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Computing Surveys*, 30(4):412–458, 1998.

D.P. Bertsekas and J.N. Tsitsiklis. Gradient convergence in gradient methods with errors. *SIAM Journal in Optimization*, 10(3):627–642, 2000.

M. Bloesch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based mav navigation in unknown and unstructured environments. In *Proceedings of the IEEE In-*

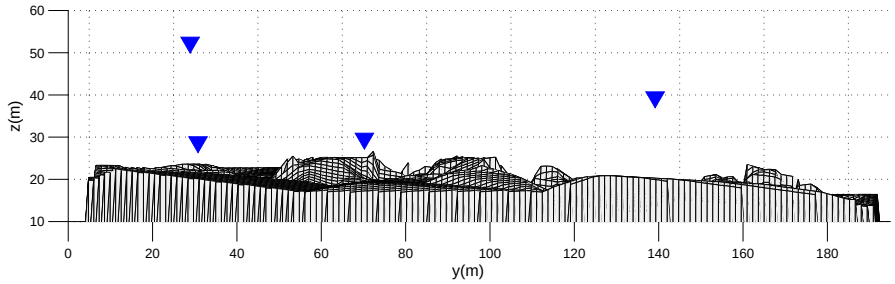


Figure 19: Final configuration of a robot team in a coverage scenario in Birmensdorf area (2D View).

ternational Conference on Robotics and Automation (ICRA), Anchorage, USA, 2010.

A. Breitenmoser, J. Metzger, R. Siegwart, and D. Rus. Distributed coverage control on surfaces in 3d space. In *Proceedings of the IEEE International Conference on Robotics and Intelligent System (IROS)*, Taipei, Taiwan, 2010a.

A. Breitenmoser, M. Schwager, J. Metzger, R. Siegwart, and D. Rus. Voronoi coverage of non-convex environments with a group of networked robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, 2010b.

J. Cortés, S. Martínez, T. Karataş, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

A. Ganguli, J. Cortés, and F. Bullo. Maximizing visibility in nonconvex polygons: nonsmooth analysis and gradient algorithm design. In *Proceedings of the American Control Conference*, volume 2, pages 792–797, 2005.

A. Ganguli, J. Cortés, and F. Bullo. Visibility-based multi-agent deployment in orthogonal environments. In *Proceedings of the American Control Conference*, pages 3426–3431, 2007.

A. Howard, M.J. Matarić, and G.S. Sukhatme. Distributed coverage control on surfaces in 3d space. In *Proceedings of the IEEE International Conference on Robotics and Intelligent System (IROS)*, pages 2849–2854, Lausanne, Switzerland, 2002a.

A. Howard, M.J. Matarić, and G.S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *6th International Conference on Distributed Autonomous Robotic System (DARS)*, pages 299–308, 2002b.

L. Jin, M.M. Gupta, and P.N. Nikiforuk. Neural networks and fuzzy basis functions for functional approximation. *Fuzzy Logic and Intelligent Systems*, 3: 17–67, 1995.

G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Proceedings of the Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR07)*, 2007.

E.B. Kosmatopoulos. An adaptive optimization scheme with satisfactory transient performance. *Automatica*, 45(3):716–723, 2009.

E.B. Kosmatopoulos and A. Kouvelas. Large-scale nonlinear control system fine-tuning through learning. *IEEE Transactions Neural Networks*, 20(6):1009–1023, 2009.

E.B. Kosmatopoulos, M. Papageorgiou, A. Vakouli, and A. Kouvelas. Adaptive fine-tuning of nonlinear control systems with application to the urban traffic control strategy tuc. *IEEE Transactions on Control Systems Technology*, 15(6): 991–1002, 2007.

J. Le Ny and G.J. Pappas. Sensor-based robot deployment algorithms. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, pages 5486–5492, Atlanta, GA, USA, 2010.

L. Pimenta, V. Kumar, R.C Mesquita, and G. Pereira. Sensing and coverage for a network of heterogeneous robots. In *47th IEEE Conference on Decision and Control*, pages 3947–3952, Cancun, Mexico, 2008.

M.M. Polycarpou and P.A. Ioannou. Identification and control of nonlinear systems using neural network models: Design and stability analysis. Technical Report 91 -09-01, University of Southern California, Dept. Electrical Engineering - Systems, 1991.

A. Renzaglia, L. Doitsidis, A. Martinelli, and E.B. Kosmatopoulos. Cognitive-based adaptive control for cooperative multi-robot coverage. In *Proceedings of the IEEE International Conference on Robotics and Intelligent System (IROS)*, pages 3314–3320, Taipei, Taiwan, 2010.

A. Renzaglia, L. Doitsidis, A. Martinelli, and E.B. Kosmatopoulos. Adaptive-based distributed cooperative multi-robot coverage. In *American Control Conference (ACC)*, pages 468–473, San Francisco, CA, USA, 2011.

P. Sadegh. Constrained optimization via stochastic approximation with a simultaneous perturbation gradient approximation. *Automatica*, 33:889–892, 1997.

M. Schwager, J. McLurkin, and D. Rus. Distributed coverage control with sensory feedback for networked robots. In *Proceedings of the of Robotics: Science and Systems*, Philadelphia, USA, 2006.

M. Schwager, B.J. Julian, and D. Rus. Optimal coverage for multiple hovering robots with downward facing camera. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3515–3522, Kobe, Japan, 2009.

T.C. Shermer. Recent results in art galleries. In *IEEE Proceedings*, volume 80, pages 1384–1399, 1992.

J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3): 332–341, 1992.

S. Weiss, M. Achtelik, L. Kneip, D. Scaramuzza, and R. Siegwart. Intuitive 3d maps for mav terrain exploration and obstacle avoidance. In *International Conference on Unmanned Aerial Vehicles (UAV)*, Dubai, 2010.