Coverage Mission for UAVs Using Differential Evolution and Fast Marching Square Methods

Article *in* IEEE Aerospace and Electronic Systems Magazine · February 2020

DOI: 10.1109/MAES.2020.2966317

Concepción A Monje
University Carlos III de Madrid

Santiago Garrido
University Carlos III de Madrid

Luis E Moreno
University Carlos III de Madrid

Verónica González
University Carlos III de Madrid

Coverage Mission for UAVs using Differential Evolution and Fast Marching Square Methods

V. Gonzalez, C. A. Monje, *Member, IEEE,* S. Garrido, *Member, IEEE,* L. Moreno, *Member, IEEE,* and C. Balaguer, *Member, IEEE,*

***Abstract*—This research presents a novel approach for missions of Coverage Path Planning (CPP) carried out by a UAVs in a 3D environment. These missions are focused on path planning to cover a certain area in an environment in order to carry out tracking, search or rescue tasks. The methodology followed uses an optimization process based on the Differential Evolution (DE) algorithm in combination with the Fast Marching Square (FM^2) planner. The DE algorithm evaluates a cost function to determine what the zigzag path with the minimum cost is, according to the steering angle of the zigzag bands (α). This optimization process allows achieving the most optimal zigzag path in terms of distance traveled by the UAV to cover the whole area. Then, the FM^2 method is applied to generate the final path according to the steering angle of the zigzag bands resulting from the DE algorithm. The approach generates a feasible path free from obstacles, keeping a fixed altitude flight over the ground. The flight level, smoothness and safety of the path can be modified by two adjustment parameters included in our approach. Simulated experiments carried out in this work demonstrate that the proposed approach generates the most optimal zigzag path in terms of distance, safety and smoothness to cover a certain whole area, keeping a determined flight level with successful results.**

***Index Terms*—UAVs, Fast Marching Square, Coverage Path Planning, Fixed flight level, Differential Evolution.**

I. INTRODUCTION

Nowadays, Unmanned Aerial Vehicles (UAVs) are being widely used in a large list of applications, whether civil or military fields. These applications focus mainly on inspections, surveillance or tracking. The majority of these tasks can result very repetitive and costly, just as can be dangerous for the human being. For this reason, the use of UAVs for area coverage applications is emerging in recent years. There are different fields where the Coverage Path Planning (CPP) is being used. For instance, a solution to perform aerial imaging applied to the precision agriculture is presented in [1]; in [2] a path is found in order to completely visit an area for 3D terrain reconstruction application; the area coverage problem is also studied in [3] for a team of aerial mobile robots; besides, the CPP problem is used for in-detail inspection of 3D natural structures on the ocean floor in [4].

Nevertheless, when an area is tracked, it is important to take into account several factors with the aim of saving energy and reducing the flight time. The main factor is that the planned trajectory that covers the whole area has to be an optimal path (in the metrics defined by the gray level matrix W). For this purpose, methods focused on Fuzzy control [5] and Genetics Algorithms (GA) are studied in different situations to evaluate the paths according to a cost function in order to achieve the most optimal result. For instance, in [6] a GA combined with space partitioning methods is used to find the most optimal path to cover a desired area. The Differential Evolution (DE) algorithm is utilized in [7] to optimize a short-term sea trajectory for an underwater glide. Another work that proposes an improved constrained DE algorithm is [8], where an optimal feasible route for UAVs is generated. In [9] the DE algorithm is also used in order to find optimal paths for coordinated UAVs, where the paths are modeled with straight line segments.

The authors are with the RoboticsLab at the Department of Systems and Automation, Carlos III University of Madrid, Spain, e-mail: (vegonzal,cmonje,sgarrido,moreno,balaguer)@ing.uc3m.es

Most techniques of CPP outlined above have been proposed for environments without obstacles. In the case of environments where obstacles are considered, an adequate path planning is required. In this way, the UAV can fly from one point to another avoiding any obstacle in the environment. Many planners have been used to resolve the path planning and obstacle avoidance problem for UAVs, RRT algorithms [10], searching algorithms [11], [12], nature inspired methods such as ant colony [13] or algorithms based on Probabilistic Roadmap Method (PRM) [14], among others.

The path planning requirements depend on the specific task. The majority of the CPP applications require to capture videos or a large number of overlapping images with a high quality and with the same resolution. This implies maintaining a fixed flight level with respect to the ground. This problem has been denoted as Terrain Following Flight (TFF) and has been studied in several works over the last decade. For instance, [15] presents path planning and optimization methods in TFF, though the problem is treated only in 2D. A 3D terrain following/avoidance trajectory optimization is performed in [16], where the problem is solved generating a path and later, optimizating the resulting path. In a previous work by the authors in [17], the path is planned at a constant height among the buildings of a city or above some of the buildings, according to the specific requirements of the mission. However, there are few works which join the CPP, the obstacle avoidance and the TFF problems.

The main objective of this work is to plan a feasible path for a UAV to cover a whole area, keeping a flight level with respect to the ground (TFF). In our approach, the CPP problem is solved using a coverage algorithm. The resulting paths from this algorithm are evaluated by the DE algorithm, choosing the most optimal one in terms of distance cost. Once the most optimal path is obtained, the Fast Marching Square (FM^2)

algorithm is used as the planner.

The FM^2 has been used for different purposes, such as path planning with mobile vehicles [18], [19], [20] and vehicle formations in 2D [21] and 3D [22] environments, providing optimal trajectories in terms of smoothness and safety.

To achieve a feasible path planning for a UAV, the inclusion of certain constraints into the FM^2 is an important factor to take into account. In [23], the authors have proven to generate paths with adaptive smoothness and compatible with UAV kinematic restrictions, where the paths resulting from FM^2 are compared with those resulting from considering the Dubins Model. Thanks to the introduction of two adjustment parameters, the algorithm will allow the path to fulfill both flight level and smoothness constraints so that it can be feasibly executed by a real UAV platform without the need to implicitly include its kinematic or dynamic model into the algorithm.

The contributions of this paper are as follows: (1) we use for the first time a combination of the DE algorithm and the FM^2 method to solve a CPP problem, keeping a fixed flight level over the terrain; (2) the FM^2 method has been modified to introduce two adjustment parameters p_1 and p_2 that allow both changing the smoothness of the path and setting the flight level in a very intuitive way and without adding computational complexity to the approach; (3) the generated path is optimal in terms of distance cost, safety and smoothness, taking into account the environment and running the approach only once for the case of static environments; (4) the low computational cost of the approach makes it suitable for its application in real time for dynamic environments where moving obstacles have to be avoided. This research can be useful in a future when applied for surveillance and tracking tasks or monitoring (videos or overlaying images) of a whole area with a minimum distance cost, among other applications.

The advantages of our approach lie in the following aspects:

- *Easy concept*: The method is based on the natural movement of a wave, so conceptually, it is very easy to understand. Thanks to this, the imposition of the flight level constraint is simple to implement.
- *Feasible trajectories*: The method provides very smooth paths, since it is based on the propagation of a wave, and no later optimization is required in order to respect the kinematics of the UAV.
- *Fast response*: As it is not necessary to include a kinematic model into the algorithm, the computational cost is reduced considerably, allowing it even to be executed in real time for dynamic environments.

This article is organized as follows. Section II introduces the problem statement, the mission to be carried out and the environment. A summary of the methods used for this research is presented in Section III. The proposed approach to cover a whole area with a minimum distance cost and maintaining a flight level with respect to the ground is presented in Section IV. Section V presents the simulation results from the application of the proposed approach. Finally, the conclusions and future works are presented in Section VI.

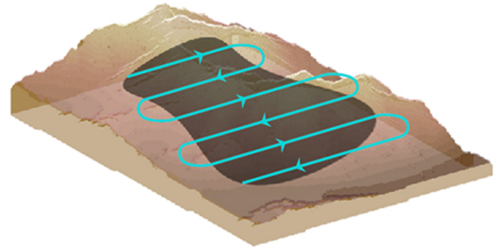

Fig. 1: 3D Environment. The representation of the area to be covered by the UAV and the zigzag path to cover the surface.

II. Problem Statement

In this work, a mission planning for UAVs is carried out, whose main objective is to cover a whole area in the most optimal way in terms of distance cost. The UAV also keeps a fixed altitude with respect to the ground. In this way, for certain applications such as tracking or cartography, the UAV could track a whole area and obtain images of the terrain, maintaining a homogeneous size of the pixel and without overlapping.

Our approach has the following inputs: an environment map where the mission is carried out and a surface which represents the area for the CPP. The environment is represented by an open field map with mountainous terrain, where the surface is rather uneven, as shown in Fig. 1. This map is a 3D grid map whose size is 118 x 87 x 40 cells. The size of each cell is equivalent to 10 x 10 x 10 meters. Here, it is not necessary to consider the size of the UAV, since it is assumed to be smaller than the size of a cell.

An instance of the surface and the zigzag path for that surface is represented in Fig. 1, which is planned with a certain width between the bands and with a determined steering angle α. Here, the steering angle is defined as the direction of each band with respect to the x and y axes (as shown in Fig. 2a), and the width between the bands corresponds with the camera visual field of the UAV. It has to be taken into account that the real camera footprint must be considered when preparing the simulation environment for real applications and that camera requirements must be adequately addressed.

The execution of the path by the UAV can be affected by its kinematics and dynamics. However, large environments are considered in general inspection cases and it is assumed that the turns of the path are compatible with the turning radius of the UAV and do not compromise its dynamic stability.

The length of the trajectory resulting from the coverage algorithm depends mainly on the steering angle. The DE algorithm will evaluate iteratively a cost function defined by the distance of the zigzag path, obtaining as a result the steering angle of the path with minimum distance cost. Then, the FM^2 method is used to plan the final path according to the resulting zigzag path, avoiding any obstacles in the environment and maintaining a certain altitude with respect to the ground. The final path is composed of a set of consecutive points ($Q = P_0, P_1, ..., P_n$) and its smoothness and safety distance can be modified by two adjustment parameters used in the FM^2 method. In this way, the path could be adapted to the kinematics of the UAV.

III. Method for Coverage Path Planning

Our proposal for CPP is divided into three different parts: the Zigzag Path method, the DE algorithm and the FM^2 method. Next all these parts will be explained in detail. All algorithms and simulations presented have been implemented and run using Matlab R2018a software in a standard MacBook Pro with processor 2,4 GHz Intel Core i5.

A. Zigzag Path Method

The area to be covered is a surface, which can be represented by an irregular form (for instance, see Fig. 2a). To begin the zigzag path planning in the area, the tangent point to the surface is determined according to a certain angle α. This angle corresponds with the steering angle of the bands. The tangent point is found within the enclosed area shown in Fig. 2a, and the angle α is delimited between [1, 89] degrees.

A parallel band (b_1) to the tangent line is calculated with a distance of $\frac{D}{2}$ (see Fig. 2b), where D is the distance corresponding to the visual field of the UAV camera. Here, b_1 cuts the surface in two points denoted as cut points Pc_1 and Pc_2. In our case, the direction of the zigzag path will always be from the greatest values of x to the smallest values. For this reason, Pc_1 will be the initial point of the trajectory and Pc_2 will be the point that connects with the arc towards the second band.

The next band (b_2) is calculated with a distance D from b_1, where the cut points Pc_3 and Pc_4 are obtained (see Fig. 2c). The cut points Pc_2 and Pc_3 will be connected through an arc, so both points must be on a perpendicular line to any band. For this reason, two perpendicular lines are calculated for these points to check which one has to be modified (see Fig. 2d). The point outside the surface is replaced by the new one, in this case Pc_2.

To ensure that all area is covered in the opposite side of the arc according to the vision field, a parallel line to b_1 is calculated with a distance of $\frac{D}{2}$, bv_1 (see Fig. 2e). Then, a perpendicular line is calculated above the cut point of bv_1 with the surface. If the points where this perpendicular line cuts with b_1 and b_2 cover a larger area, these will be considered. Here, Pc_1 would be modified.

All this process is repeated for the next bands, as shown in Fig. 2f. In this figure it can be observed that Pc_3 is modified because of the perpendicular line above the cut point of bv_2, which entails a modification of Pc_2.

To finish with the process, two cases can be given. On the one hand, a last band outside the surface, b_{n+1}, may be necessary, where n is the total number of bands, to cover the whole area of the surface. In this case, a parallel band to b_n is calculated outside the surface with a distance D. According to Fig. 3a, a perpendicular line is calculated above the last cut point (Pc_m). The cut point of this perpendicular line with b_{n+1} will be the initial point of this band (Pc_{m+1}). Then, a parallel line bv_n is calculated with a distance $\frac{D}{2}$ from b_n. A new perpendicular line is calculated above the point where bv_n cuts with the surface (this points depends on the flight direction). The point when this perpendicular line cuts with b_{n+1} will be the final point of the path (Pc_{m+2}).

On the other hand, bv_n could be outside the surface. Here, a perpendicular line is calculated above the cut point of bv_{n-1} with the surface (on the opposite side of the arc). The cut point of this perpendicular line with b_n will be the final point of the path.

B. Differential Evolution Algorithm

Once the zigzag path is generated, the DE algorithm evaluates iteratively a cost function defined by the distance of the path, according to its steering angle (α) defined as the direction of each band with respect to x and y axes, as described before. In this way, the most optimal zigzag path in terms of distance cost is obtained, and therefore the best steering angle.

The DE algorithm was proposed in 1996 by Storn and Price [24] and it is a relatively new population-based stochastic global optimization algorithm. It is based on a genetic algorithm and combines concepts of $differential\ mutation$, $crossover$ and $selection$. In order to optimize a certain function, this method uses n D-dimensional parameter vectors, $x_i^G = \{x_{i,1}^G, x_{i,2}^G, ..., x_{i,d}^G\}$; i = 1, 2, ..., n, where d is the number of real parameters of the function and G the generation number.

The initial population is chosen randomly, defining initial limits $x_j^{min} \leq x_{j,i,1} \leq x_j^{max}$ and must cover the entire parameter space. A uniform probability distribution is assumed for all random decisions. DE generates new parameter vectors through the addition of the weighted difference vector between two population vectors to a third vector. This process is denoted as $mutation$. The resulting vector is called $trial$ vector and is defined by

$$v_i^{G+1} = x_{r1}^G + Ft(x_{r2}^G - x_{r3}^G), \tag{1}$$

where $Ft \in [0, 2]$ is a factor which controls the amplification of the differential variations $x_{r2}^G - x_{r3}^G$, which is determined by the user, and $r1$, $r2$, $r3$ $\in\{1, 2, ..., n\}$, being $r1 \neq r2 \neq r3$. The vector v_i^{G+1} is called $donor$ vector.

A $crossover$ is introduced with the aim of increasing the diversity of the new generation of parameter vectors.

The process of $mutation$, $crossover$ and $selection$ is repeated depending on the criteria established by the user, such as the maximum number of iterations and the population number.

C. Fast Marching Square Method

When the best zigzag path in terms of distance cost is obtained, the FM^2 algorithm is used to plan the final path at a fixed altitude with respect to the terrain, avoiding any obstacles found in the environment.

The FM^2 algorithm has been widely used in various fields, such as path planning for mobile robots [20], path planning for UAVs [23] or for robot learning [25]. All these applications require very smooth trajectories, which, moreover, have to maintain an adequate safety distance from the obstacles of the environment. The FM^2 method satisfies these conditions

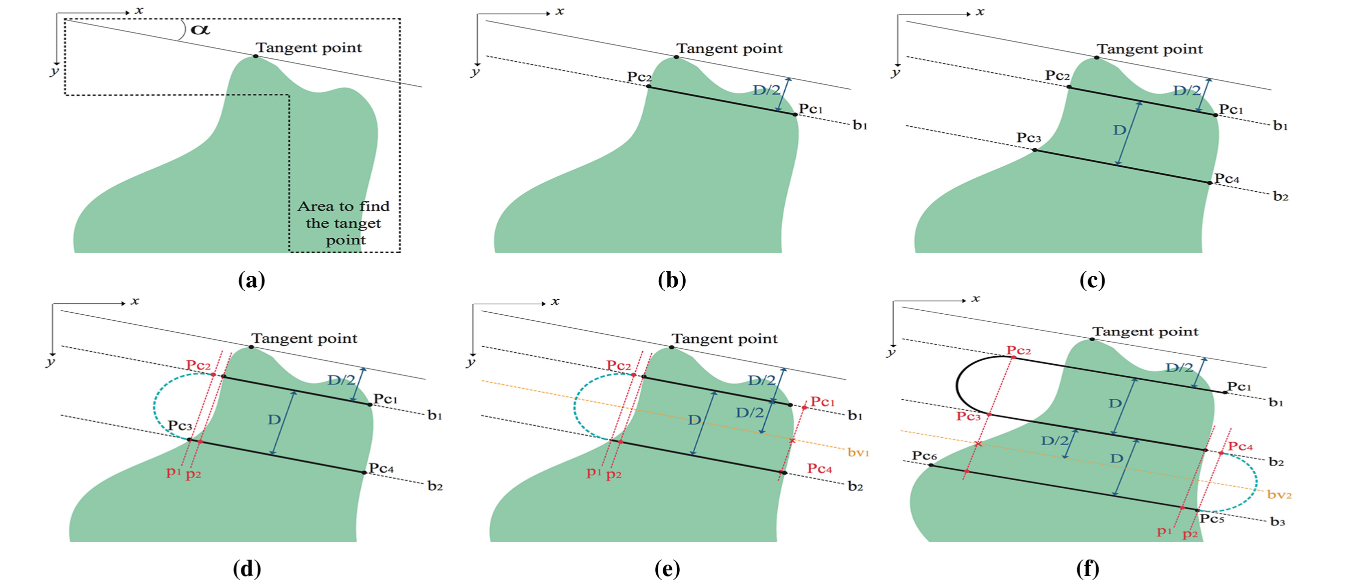

Fig. 2: Process to cover all the surface with a zigzag path. (a) Area to find the tangent point. (b) The first band is calculated. (c) The next band is calculated. (d) Creation of the cut points for the arc. (e) Adaptation of the cut points for the UAV visual field. (f) Repetition of the process for the rest of the bands.

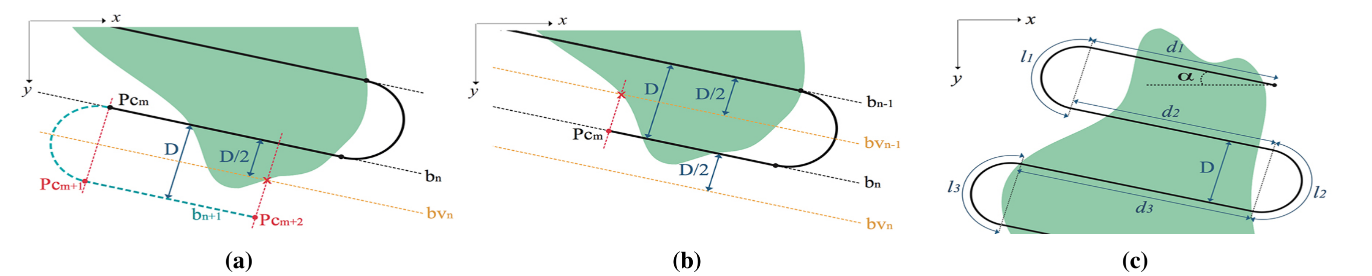

Fig. 3: Two final cases to cover all the surface: (a) a new band is necessary to cover the whole area; (b) no further bands are needed to cover the whole area. (c) Illustration of the final zizag bands over the surface.

and is based on applying the Fast Marching Method (FMM) twice. The FM^2 method and the FMM are explained in more detail below.

1) Fast Marching Method: The FMM is an efficient numerical algorithm to calculate the expansion of a wave on a media with varying refraction index or inverse of the expansion speed (Eikonal or light equation) (2). It was introduced by Sethian [26] and is a planning method of order O(Nlog(N)), where N is the total number of sampling points, that is, the number of grid points in which the map is distributed.

The method works in the following way: ρ = (x, y) is a point of the space, $W(\rho)$ is the expansion speed of the wave front and $D(\rho)$ is the arrival time of the wave front to each point of the space. The eikonal equation is

$$1 = W(\boldsymbol{\rho})|\nabla D(\boldsymbol{\rho})|. \tag{2}$$

As explained in [27], the FMM works by iterations solving the value $D_{i,j}$ for each cell of the grid map. Later, the gradient descent is applied over any point of the time of arrival map up to the point where the wave has been originated. In this way we can obtain the path between a start point and the goal point.

A main feature of this method is that it provides only a local minima. Consequently, the resulting path is the most optimal path in distance.

2) Fast Marching Square Method: The FMM generates optimal trajectories in terms of distance. However, this is not the only thing to take into account when a path planning for a robot is carried out. The trajectory must be smooth without sharp curves, always respecting a certain turning radius. Also, the trajectory must have safety margins with respect to the obstacles to prevent accidents with the environment. These two deficiencies are solved by applying the FM^2 method.

The FM^2 method was introduced by Garrido et al. [28] in 2009 and, as mentioned above, is based on applying the FMM twice. The first time that FMM is applied, a potential map W is generated and then, the FMM is applied again to generate the path between two points. The procedure to obtain a path between two points is as follows:

- *Environment (W_0)*: The input of the method is a 3D grid map, which is read as a binary map (see Fig. 4a). The obstacles are identified with value 0 (black) and the free space is identified with value 1 (white).
- *Velocities map (W)*: Each cell of the grid map labeled

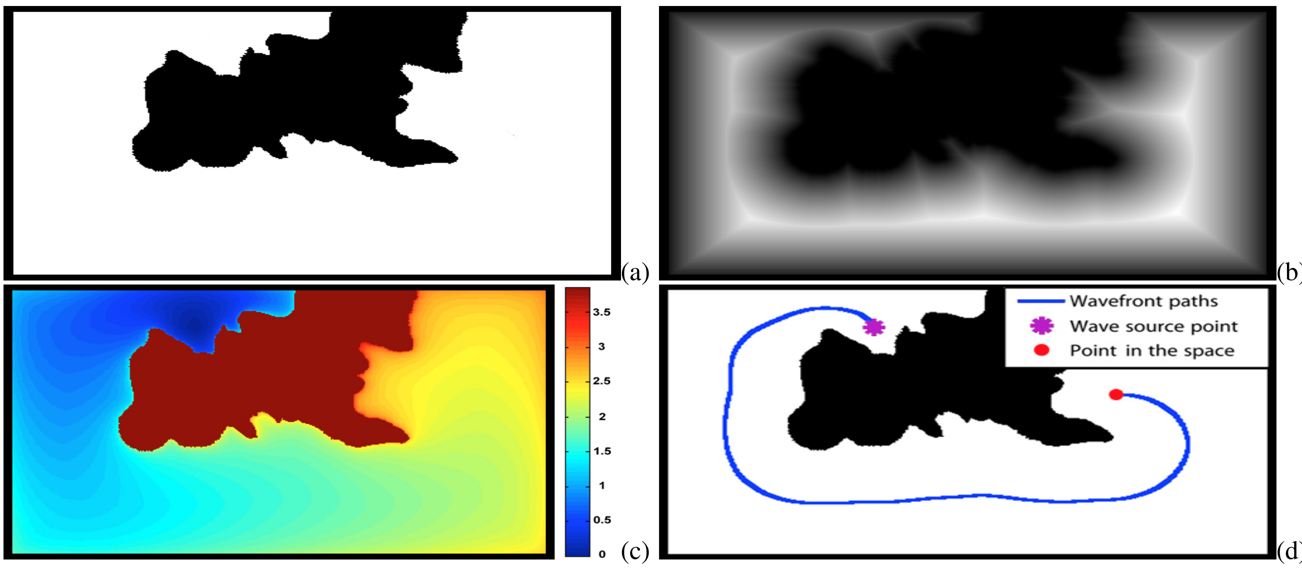

Fig. 4: The Fast Marching Square method (FM^2). (a) Binary map W_0. (b) Velocities map W. (c) Time of arrival map D. (d) Resulting path.

as obstacle is used as source point of the FMM. In this way, a potential map is generated as shown in Fig. 4b. This map in grayscale is rescaled to fix the maximum and minimum values as 1 and 0, respectively. The value of each cell is proportional to the distance from the obstacles, in other words, now the free space keeps a certain distance from the obstacles. This map is also called *Velocities map* because the value of each cell can be interpreted as the speed of the vehicle, that is to say, the speed is faster when the vehicle is far from the obstacles (clear areas) and the speed is slower when the vehicle is approaching obstacles (obscured areas). But in this work, the speed has not been considered when carrying out the path.

- *Time of arrival map (D)*: The FMM is applied again over the map W, where the wave is expanded from the goal point until the start point. The result of this process is the *Time of arrival map D* shown in Fig. 4c.
- *Resulting path*: The resulting path (see Fig. 4d) is obtained applying the gradient descent over D from the start point to the goal point. The resulting path is the most optimal in terms of smoothness and safety (in the metrics defined by the gray level matrix W).

However, many times the resulting paths are not optimal in terms of safety margins or smoothness, since the path does not befit the requirements of the mission. Thus, W can be modified according to certain specifications, such as security margins and kinematic of the vehicle.

As a novel contribution of this research, each cell of the map W can be raised to a value specified by the user. This value is called adjustment parameter. This procedure causes a lightening or darkening of W, as shown in Fig. 5a. If the cells are raised to a value a greater than 1 the map is darkened, producing paths with sharp curves and further away from the obstalces. By contrast, if the cells are raised to a value smaller than 1, the paths are smoother and closer to the obstacles (see Fig. 5b).

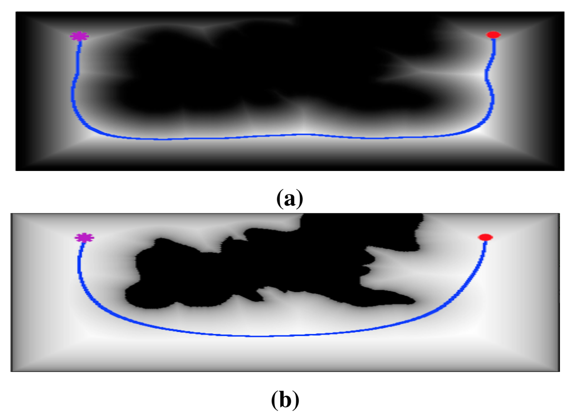

Fig. 5: W raised to different values. (a) W raised to $\frac{3}{2}$. (b) W raised to $\frac{1}{4}$.

It is noteworthy that the form of the path depends mainly on the environment and, sometimes, a different result can be generated. For instance, there may be cases where the path keeps a great safety distance from the obstacle with a great smoothness. It could happen if the environment is composed exclusively of one obstacle, since the trajectory is totally dependent on the environment. However, this is not the case in this work.

IV. Coverage Path Planning Approach

This section presents our approach, whose main objective is to generate a feasible zigzag path with the minimum distance cost in order to cover a certain area of an environment. The path is planned according to the visual field of the UAV, maintaining a fixed flight level with respect to the ground. The area to be covered, the flight level and the distance between the bands are chosen by the user. The environment where the CPP is carried out is the one described in Section II.

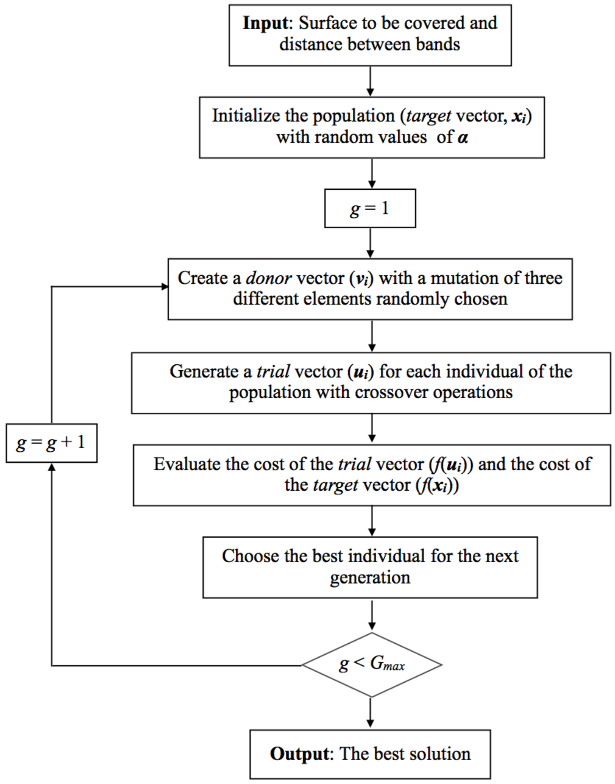

Fig. 6: Flowchart of the DE algorithm for our approach.

First, the Zigzag Path method is used to generate the zigzag path that will cover all the surface, and then the DE algorithm is used to determine what the best steering angle of the zigzag bands (α) is, providing the most optimal zigzag path in terms of distance cost. Thanks to the evolutionary optimization technique of this algorithm, a minimization of the cost function is possible.

The bands of the zigzag are disposed to cover the whole area according to a certain distance (D) between each band, as shown in Fig. 3c. This distance is equivalent to the vision field of the UAV. Each strech of the bands is composed of the longitude of a straight line (d_i) and the longitude of an arc (l_i). The cost of the path is determined by the sum of each of the longitudes. So, the cost function of this problem to be minimized is described as follows:

$$Ct(\alpha) = \sum_{i=1}^{n}(d_i + l_i), \tag{3}$$

for $i = 1, 2, ..., n$, n being the maximum number of stretches in the planning. Next, it is explained how the DE algorithm is used in our approach, whose procedure is described in Fig. 6.

- *Initialization*: The population is formed by a specific number of individuals N_p = 20, which represents each possible solution. For the initial generation $g = 0$, the initial population is calculated using a random number distribution according to $x^0_{i,j} = \alpha_{max} \cdot \tau$, where α_{max} = 89° is the maximum permissible angle and τ is a random value between [0, 1].
- *Mutation*: The mutation process generates the $donor$ vector from the linear combination of different members of the population according to $v^g_i = x^g_{r_1} + F \cdot (x^g_{r_2} - x^g_{r_3})$, where the parameter related to the speed of convergence is a positive constant F = 0.8, whose value gives a good precision of the solution, and values r_1, r_2 and r_3 are random individuals, being $r_1 \neq r_2 \neq r_3 \neq i$. In this way, the mutation is based on the randomness.
- *Crossover*: The crossover process is used to increase the diversity of the new generation.
- *Selection*: The selection mechanism compares the individuals of the cost of the $trial$ vector and the cost of the $target$ vector. Here, the best result is chosen. If the value of the $trial$ vector cost is smaller than the value of the $target$ vector cost, the result is the value of u^g. On the contrary, the best result will be the value of the individual of the population x^g, maintaining in both cases this value for the next population.

Fig. 7a shows the specific surface to track in the map and Fig. 7b shows the resulting zigzag path after the application of the DE algorithm to cover the whole area of the surface with a minimum cost.

Once the zigzag path has been obtained in Fig. 7b, an enlargement of the zigzag path is applied, obtaining W_0 matrix as shown in Fig. 7c. Over this new enlarged path, the FM method is now applied, obtaining W matrix as shown in Fig. 7d. Now, FM algorithm is used again to plan the final path at a fixed altitude with respect to the terrain (using p1 and p2 adjustment parameters), avoiding any obstacles found in the environment. The resulting path is shown in Fig. 7e and Fig. 7f in top and 3D views, respectively.

As an original contribution to the FM^2 algorithm, in order to impose a fixed flight level for the UAV, W is modified according to the parameters p_1 and p_2, whose main purpose is to clarify and obscure each cell of W (see Fig. 8). As explained in III-C, the wave front of the FM^2 tends to expand through the clearer areas. For this purpose, the path is forced to be planned through certain areas of the map clarifying the areas belonging to the desired altitude and obscuring the rest of the cells. This process is explained with Algorithm 1.

The terrain elevation for each cell $Surface(i, j)$ of the map is saved in the variable $SurfaceValue$ (see Algorithm 1, line 4). This elevation is computed taking as reference the layer 0 of the environment, which can be identified as sea level. Then, the value of the given flight level is added to the $SurfaceValue$ (Algorithm 1, line 5). These cells are clarified raising their value to p_1 (Algorithm 1, line 6). Thus, the planning through these zones will be easier. The rest of the cells are obscured raising them to p_2 (Algorithm 1, line 8), making the planning harder through them.

In Fig. 7f the 3D perspective of the resulting path from our approach is shown, where it is appreciated that a flight level with respect to the ground is maintained. The influence of the values of parameters p_1 and p_2 on the smoothness and feasibility of the path are discussed in [17].

Thanks to the introduction of the two adjustment parameters p_1 and p_2, the algorithm will allow the path to fulfil both flight level and smoothness constraints so that it can be feasibly

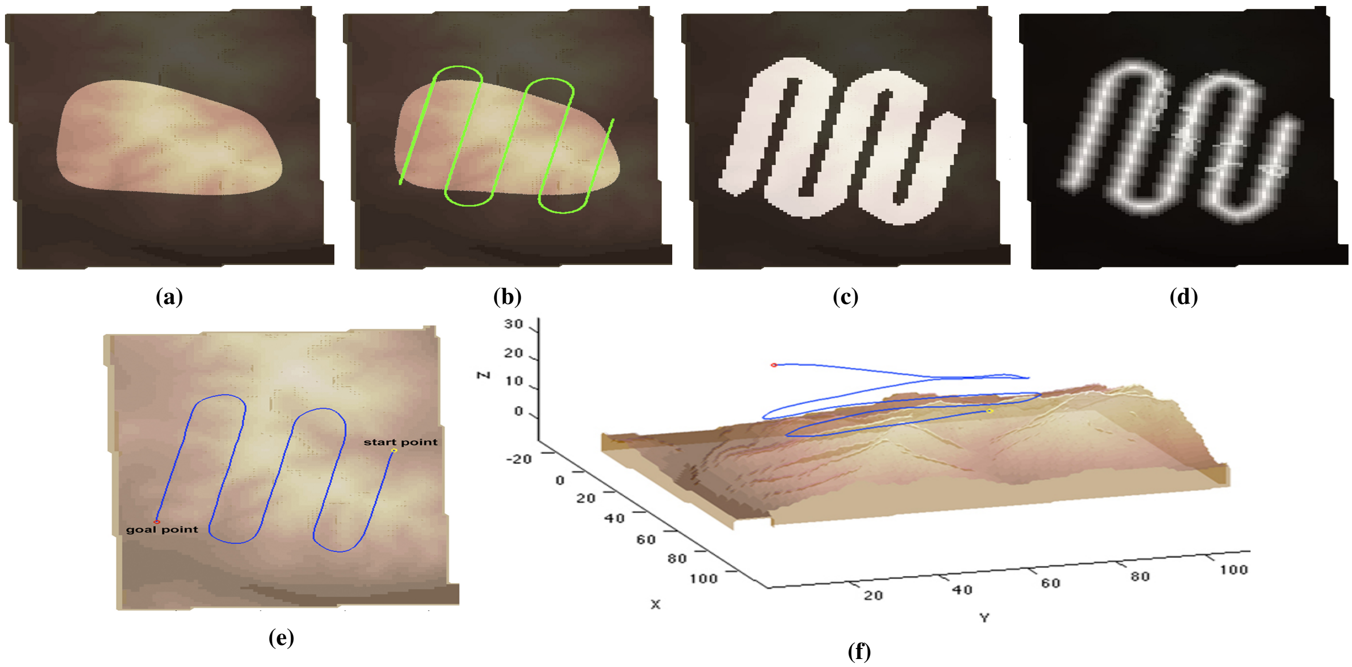

Fig. 7: Process to obtain the path with minimum cost. (a) Surface to be covered. (b) Result from the DE algorithm. (c) Enlargement of the zigzag path. (d) Applying FM to obtain W matrix. (e) Aplying FM again to obtain the final path. (f) 3D perspective of the resulting path from our approach.

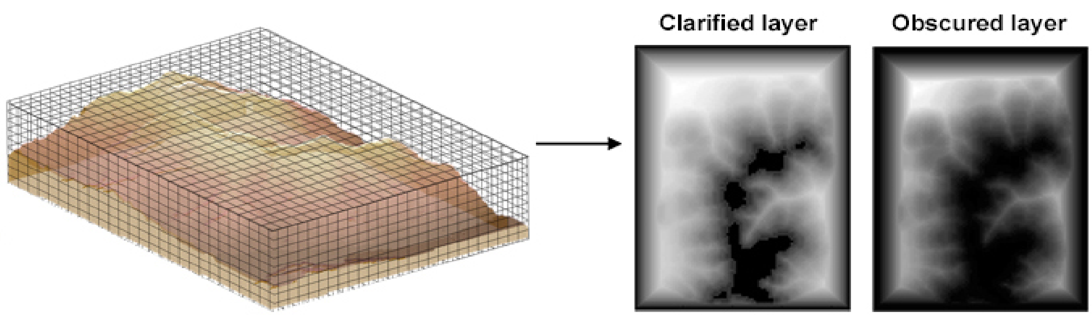

Fig. 8: The left part of the image shows the 3D map of the environment as a grid. The right part shows clarified and obscured layers of the map W.

executed by a real UAV platform without the need to implicitly include its kinematic or dynamic model into the algorithm. This fact reduces considerably the computational cost of the approach, allowing it even to be executed in real time for dynamic environments. Besides, the approach provides a desired path just in one run, and no later optimization is required for static environments, unlike other works cited previously, which also contributes to the reduction of the computational cost. Though this work does not deal with the uncertainty problem, our approach could be used in case of terrain uncertainties or even dynamic obstacles avoidance. A previous work by the authors [21] focuses on this topic in 2D for mobile robots.

V. Results and Discussion

Having defined the environment, the approach based on the DE and FM^2 algorithms takes a certain surface and plans a feasible zigzag trajectory to cover the whole area with the lowest possible cost in terms of distance at a fixed flight level with respect to the ground, avoiding any static obstacle in the environment.

Here, different paths are calculated for four different irregular surfaces. The experiment show how the smoothness of the trajectory can affect the compliance of the flight level constraint, since the terrain is highly non-uniform.

For the experiment, the DE algorithm runs over 30 iterations with a population of 20 individuals and the fixed flight level is 10 meters, with values of p_1 and p_2 of 0.5 and 1.3, respectively.

The resulting computation time of the DE part is 3.79 s and the computation time of the FM^2 part (including environment preparation) is 0.36 s. It is important to mention that once the environment is treated and the UAV is flying on its path, the FM^2 algorithm takes around 0.36 s to recalculate the trajectory, which makes this approach very suitable for dynamic environments, as discussed previously.

TABLE I: Costs and steering angles of the bands for different surfaces.

	Minimum		Maximum		Median		Average	
Surface	Cost (m)	α (^{o})	Cost (m)	α (^{o})	Cost (m)	α (^{o})	Cost (m)	α (^{o})
S1	5446.504	37.134	6104.040	11.499	5464.316	37.684	5616.718	37.214
S2	6748.057	2.503	8373.462	71.434	6768.528	3.389	6870.913	8.597
S3	5908.607	1.012	7493.887	65.506	5980.716	2.086	6061.788	9.452
S4	7176.987	1.349	8223.283	44.610	7215.164	2.581	7290.188	7.953

Algorithm 1 Imposition of a fixed flight level with respect to the ground.

Require: The Velocities map W of a gridmap G of size m x n x l.
Require: Flight level L_w with respect to the ground.
Require: Adjustment parameters p_1 and p_2.
Ensure: The Velocities map W with the clarified flight level cells.

for k **to** l **do**
  for j **to** n **do**
    for i **to** m **do**
      $SurfaceValue \leftarrow Surface(i, j)$
      if $k = (L_w + SurfaceValue)$ **then**
        $w_{i,j,k} \leftarrow (w_{i,j,k})^{p_1}$
      else
        $w_{i,j,k} \leftarrow (w_{i,j,k})^{p_2}$
      end if
    end for
  end for
end for

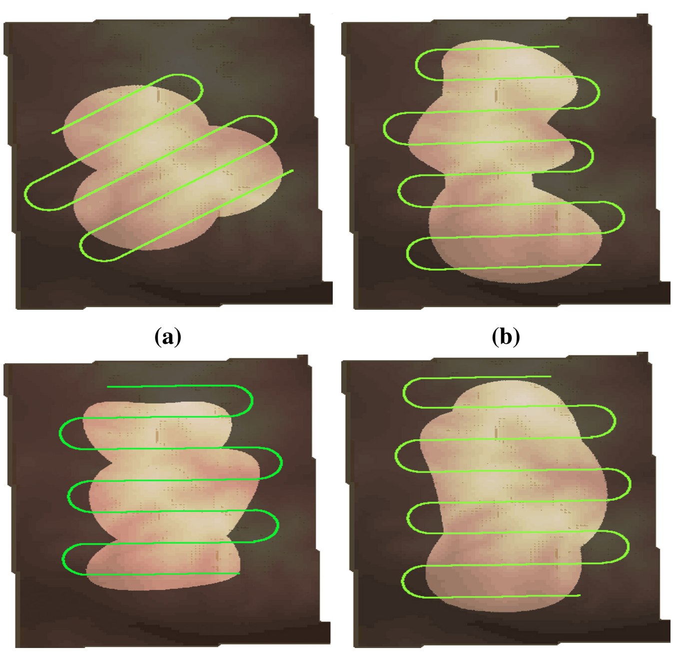

Fig. 9: Zigzag bands resulting from the DE algorithm for different surfaces: (a) zigzag bands for S1; (b) zigzag bands for S2; (c) zigzag bands for S3; (d) zigzag bands for S4.

A previous work by the authors [29] presents a deeper discussion on how the computation time changes with the number of cells of the environment. The study shows that the FM method and its variants are very competitive to this respect.

A. Different Areas to be Covered

The aim of this experiment is to plan the optimal trajectory in terms of distance cost to cover a whole area. The path has to be feasible by an UAV avoiding any obstacle in the environment and keeping a fixed altitude with respect to the ground. Four different cases are presented in this experiment, evaluating in each one a different area. The different areas to be covered are presented in Fig. 9, which have irregular forms. The chosen distance between the bands is 80 cells, because it is assumed that the visual field of the UAV is 800 meters. The size of each cell of the map is equivalent to 10 x 10 x 10 meters.

Fig. 9a shows the resulting zigzag path for the surface S1. Table I shows the results obtained from the algorithm DE, where the minimum (best), maximum (worst), average and median of the cost and angles are listed. The first line of the table shows the values for the surface S1. In this case, the best angle to obtain the most optimal path is 37.134 degrees with a trajectory cost of 5446.504 meters. The worst cost is 6104.040 meters for an angle of 11.499 degrees. The median and average are maintained in values closer to minimum. The result of applying FM^2 for a path with a steering angle of 37.134 degrees is shown in Fig. 10a. Here, the 3D perspective and x-y view of the path are shown. In Fig. 11a it can be appreciated how the planned path maintains the flight level in the majority of the trajectory, the difference between the planned path and the ideal path being minimal.

The result for the surface S2 is shown in Fig. 9b. The second row of Table I corresponds to the values of the cost and steering angles for S2. Here, the minimum cost of the trajectory is 6748.057 meters with an angle of 2.503 degrees. The worst cost is 8373.462 meters with an angle of 71.434 degrees. Fig. 10b. shows the resulting zigzag path when FM^2 is applied with a steering angle of 2.503. Fig. 11b shows the altitude profile of the trajectory for the surface S2. There are several path points where the UAV is not capable of keeping the flight level due to the values of p_1 and p_2, fixed for a certain kinematics. For a major fulfilment of the fixed altitude, parameters p_1 and p_2 must be changed so that a balance between the kinematics of the UAV, the fulfilment of the imposed flight altitude and the corresponding cost function (see Table I) is achieved, as explained in [17].

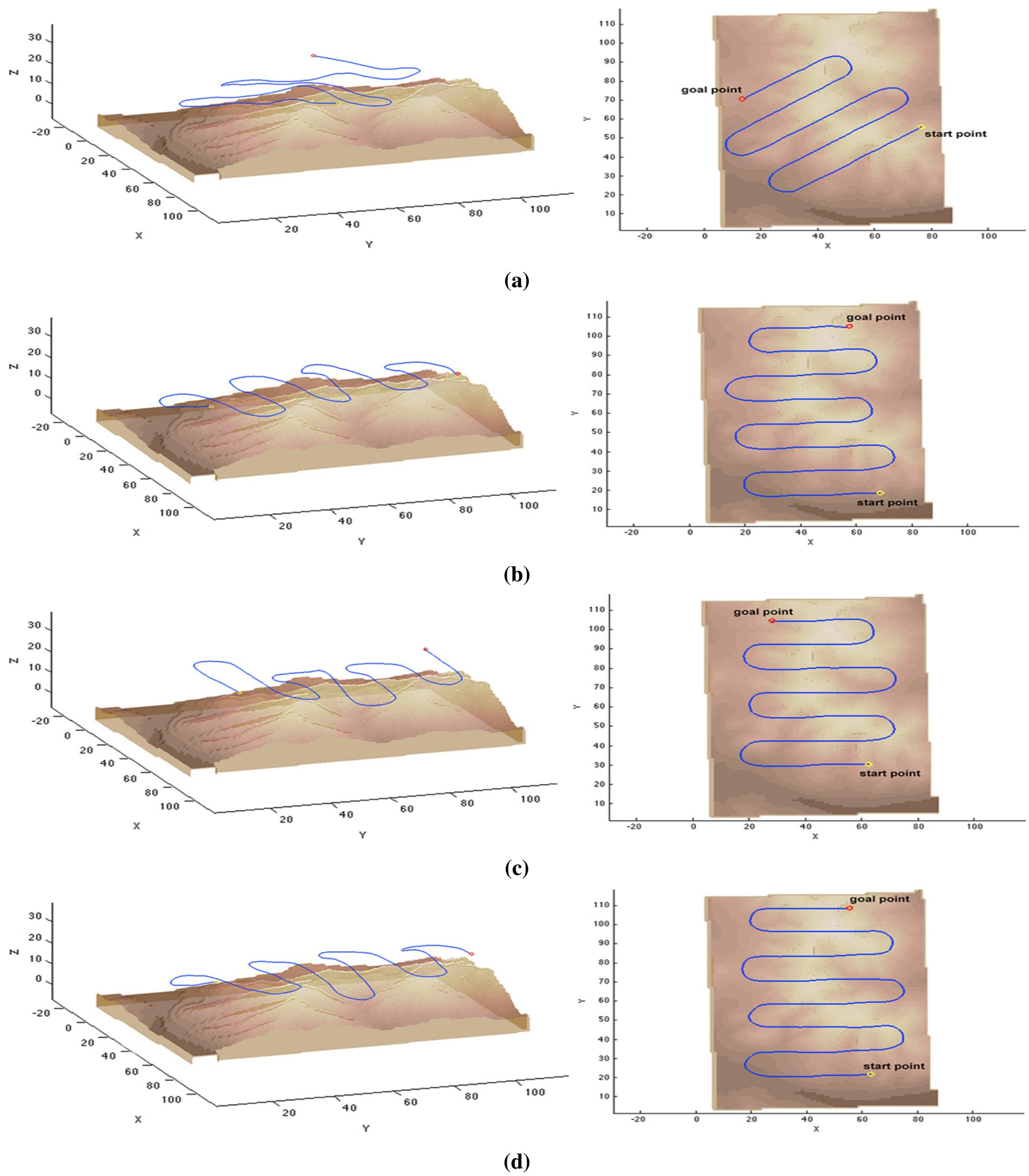

Fig. 10: 3D perspective (left) and x-y view (right) of the resulting path: (a) path for S1; (b) path for S2; (c) path for S3; (d) path for S4.

In the case of the surface S3, the result is shown in Fig. 9c. Here, it is appreciated that one last band is necessary to cover the whole area. The best angle for the path is 1.012 degrees, providing a cost of 5908.607 meters. This values can be found in the third row of Table I. The worst angle is 65.506 degrees, providing a cost of 7493.887 meters. In Fig 11c the altitude profile of the path can be appreciated. As in the previous case, the UAV is unable to reach several points of the trajectory, due to its kinematic limitations, fixed by p_1 and p_2.

Finally, the case for the surface S4 is presented in Fig. 9d, where it is appreciated that a final band of the zigzag is necessary to cover the whole area, as in the previous case. The zigzag bands have an angle of 1.349 degrees, whose trajectory has a cost of 7176.987 meters. The worst angle for this case is 44.610 degrees with a cost of 8223.283 meters. The maintained altitude for the path is shown in Fig. 11d. As in the two cases described above, the terrain has significant irregularities, being impossible for the UAV to maintain a constant level with respect to the ground.

VI. Conclusions and Future Works

This work presents a novel approach based on the DE algorithm and the FM^2 method to plan a feasible zigzag path for UAVs in order to cover a certain area of a environment. The method generates an optimal path in terms of smoothness and safety with a minimum distance cost, maintaining a fixed flight

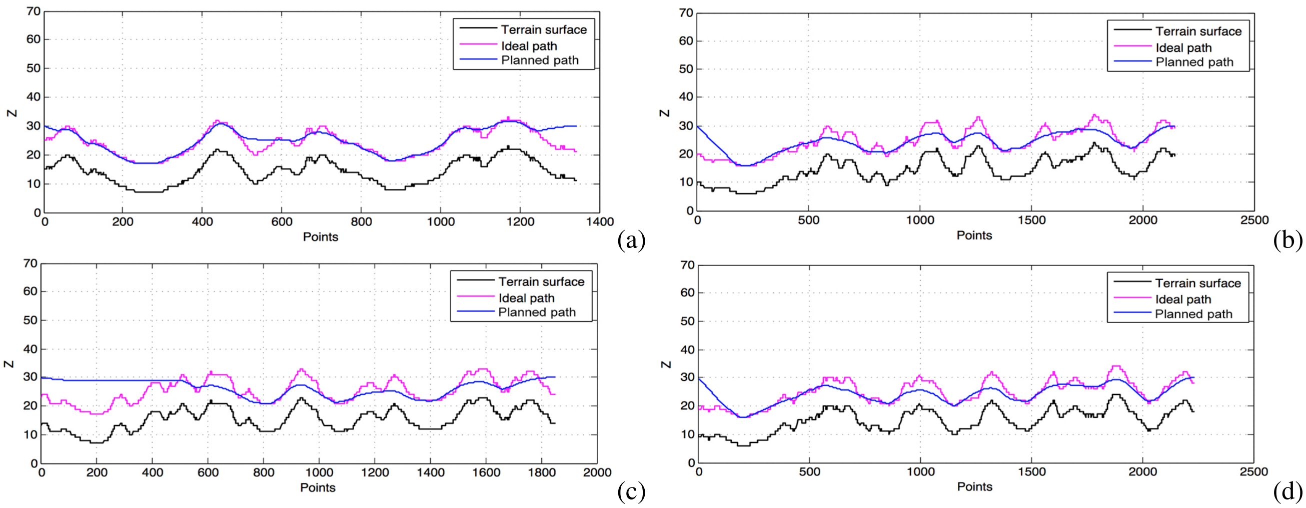

Fig. 11: Comparison of the resulting path against the ideal path and the terrain profile: (a) path for S1; (b) path for S2; (c) path for S3; (d) path for S4.

level with respect to the ground and avoiding any obstacle in the terrain. The cost of the path corresponds with its length.

First, the Zigzag Path method is used for the generation of the zigzag bands, and then the DE algorithm optimizes them so that the steering angle is optimal, ensuring a minimal distance cost. Later, the map W_0 is modified according to the obtained zigzag path and the FM^2 method is applied over this map. Two adjustment parameters p_1 and p_2 have been used to adjust the path to the kinematics of the UAV and to force the path planning in specific areas of the 3D map determined by a given flight level. The generated path is optimal not only in terms of distance cost, but also in terms of safety and smoothness, and therefore, feasible for the UAV.

Our approach has been proven with successful results. The approach works in different irregular surfaces and for different vision fields of the UAV, obtaining always the feasible path with minimum cost. This results demonstrate that our approach generates paths to save energy or fuel, and in addition, generates trajectories compatibles with the kinematics of the UAV.

In a further research, the explicit relationship between the adjustment parameters p_1 and p_2 and the kinematic and dynamic constraints of the UAV will be studied.

Acknowledgments

The research leading to these results has received funding from the RoboCity2030-III-CM project (Robótica aplicada a la mejora de la calidad de vida de los ciudadanos, fase III; S2013/MIT-2748), funded by Programas de Actividades I+D en la Comunidad de Madrid and cofunded by Structural Funds of the EU.

References

[1] A. Barrientos, J. Colorado, J. del Cerro, A. Martınez, C. Rossi, D. Sanz, and J. Valente, "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots," *J. Field Robotics*, vol. 28, no. 5, pp. 667–689, 2011.

[2] M. Torresa, D. A. Pelta, J. L. Verdegaya, and J. C. Torres, "Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction," *Expert Systems with Applications*, vol. 55, p. 441–451, 2016.

[3] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, "Distributed approach for coverage and patrolling missions with a team of heterogeneous aerial robots under communication constraints," *International J. Advanced Robotic Systems*, vol. 10, pp. 1–13, 2013.

[4] E. Galceran and M. Carreras, "Planning coverage paths on bathymetric maps for in-detail inspection of the ocean floor," in *Proc. 2013 IEEE International Conference on Robotics and Automation (ICRA), IEEE*, Ed., Karlsruhe, Germany, 2013, pp. 4159–4164.

[5] J. Y. Xixia Sun, Chao Cai and X. Shen, "Route evaluation for unmanned aerial vehicle based on type-2 fuzzy sets," *Engineering Applications of Artificial Intelligence*, vol. 39, p. 132–145, 2014.

[6] M. M. Trujillo, M. Darrah, K. Speransky, B. DeRoos, and M. Wathen, "Optimized flight path for 3D mapping of an area with structures using a multirotor," in *Proc. International Conference on Unmanned Aircraft Systems (ICUAS), IEEE*, Ed., Arlington, VA USA, 2016, pp. 905–910.

[7] A. Zamuda and J. D. H. Sosa, "Differential evolution and underwater glider path planning applied to the short-term opportunistic sampling of dynamic mesoscale ocean structures," *Applied Soft Computing*, vol. 24, p. 95–108, 2014.

[8] X. Zhanga and H. Duana, "An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning," *Applied Soft Computing*, vol. 26, pp. 270–284, 2015.

[9] I. K. Nikolos, E. S. Zografos, and A. N. Brintaki, "UAV path planning using evolutionary algorithms," *Innovations in Intelligent Machines-1*, vol. 70, pp. 77–111, 2007.

[10] D. Lee and D. H. Shim, "RRT-based path planning for fixed-wing UAVs with arrival time and approach direction constraints*," in *Proc. 2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, Orlando, FL, USA, 2014, pp. 317–328.

[11] Q. Xue, P. Cheng, and N. Cheng, "Offline path planning and online replanning of uavs in complex terrain*," in *Proc. 2014 IEEE Chinese Guidance, Navigation and Control Conference*, Yantai, China, 2014, pp. 2287–2292.

[12] B. Zhang, L. Tang, and M. Roemer, "Probabilistic weather forecasting analysis for unmanned aerial vehicle path planning," *J. Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 309–312, 2014.

[13] U. Cekmez, M. Ozsiginan, and O. K. Sahingoz, "Multi colony ant optimization for uav path planning with obstacle avoidance," in *Proc. 2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, Arlington, VA USA, 2016, pp. 47–52.

[14] A. Babiarz and K. Jaskot, "The concept of collision-free path planning of UAV objects," *Advanced Technologies for Intelligent Systems of National Border Security*, vol. 440, pp. 81–94, 2013.

[15] K. Amirreza, M. Hossein, and F. Mehdi, "A new method for trajectory optimization in terrain following flight," in *Proc. 22nd Iranian Conference on Electrical Engineering (ICEE 2014)*, Tehran, Iran, 2014, pp. 1203–1208.

[16] R. Kamyar and E. Taheri, “Aircraft optimal terrain/threat-based trajectory planning and control,” *J. Guidance, Control and Dynamics,* vol. 37, no. 2, pp. 466–483, 2014.

[17] V. Gonzalez, C. A. Monje, L. Moreno, and C. Balaguer, ”UAVs mission planning with imposition of flight level through fast marching square,” *J. Cybernetics and Systems*, pp. 1–12, 2017.

[18] C. Arismendi, D. Alvarez, S. Garrido, and L. Moreno, “Nonholonomic motion planning using the fast marching square method,” *International J. Advanced Robotic Systems*, vol. 12, no. 56, pp. 1–14, 2015.

[19] S. Garrido, M. Malfaz, and D. Blanco, “Application of the fast marching method for outdoor motion planning in robotics,” *J. Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 106–114, 2013.

[20] J. Gomez, A. Vale, S. Garrido, and L. Moreno, “Performance analysis of fast marching-based motion planning for autonomous mobile robots in ITER scenarios,” *J. Robotics and Autonomous Systems*, vol. 63, pp. 36–49, 2015.

[21] J. Gomez, A. Lumbier, S. Garrido, and L. Moreno, “Planning robot formations with fast marching square including uncertainty conditions,” *J. Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 137–152, 2013.

[22] D. Alvarez, J. Gomez, S. Garrido, and L. Moreno, “3D robot formations planning with fast marching square,” in *Proc. IEEE International Conference on Autonomous Robot Systems and Competitions*, Espinho, Portugal, 2014, pp. 59–64.

[23] V. Gonzalez, C. A. Monje, L. Moreno, and C. Balaguer, “Fast marching square method for UAVs mission planning with consideration of Dubins model constraints,” in *Proc. 20th IFAC Symposium on Automatic Control in Aerospace (ACA 2016)*, vol. 49, no. 17, Sherbrooke, Quebec, Canada, 2016, p. 164–169.

[24] R. Storn and K. Price, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces,” *J. Global Optimization*, vol. 11, no. 14, p. 341–359, 1997.

[25] J. V. Gomez, D. Alvarez, S. Garrido, and L. Moreno, “Kinesthetic teaching via fast marching square,” in *Proc. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, 2012, pp. 1305 – 1010.

[26] J. Sethian, “A fast marching level set method for monotonically advancing fronts,” in *Proc. National Academy of Sciences*, vol. 93, no. 4, USA, 1996, p. 1591–1595.

[27] A. Valero-Gomez, J. V. Gomez, S. Garrido, and L. Moreno, “Fast marching methods in path planning,” *IEEE Robotics and Automation Magazine,* vol. 20, no. 4, pp. 111–120, 2013.

[28] S. Garrido, L. Moreno, M. Abderrahim, and D. Blanco, “FM2: a real-time sensor-based feedback controller for mobile robots,” *International J. Robotics and Automation*, vol. 24, no. 1, pp. 3169–3192, 2009.

[29] J.V. Gomez, D. Alvarez, S. Garrido, L. Moreno, Fast Methods for Eikonal Equations: An Experimental Survey. IEEE Access, pp. 39005 – 39029, 2019.