Control and Optimization of UAV Trajectory for Aerial Coverage in Photogrammetry Applications

Dan POPESCU, Florin STOICAN, Loretta ICHIM
Politehnica University of Bucharest, Bucharest, 060042, Romania
dan_popescu_2002@yahoo.com, florin.stoican@gmail.com, iloretta@yahoo.com

***Abstract*—Photogrammetry is a well-studied and much-used analysis tool. Typical use cases include area surveillance, flood monitoring and related tasks. Usually, an Unmanned Aerial System (UAS) is used as support for image acquisition from an a priori delimited region in a semi-automated manner (via a mix of ground control and autonomous trajectory tracking). This in turn has led to various algorithms which handle path trajectory generation under realistic constraints but still many avenues remain open. In this paper, we consider typical costs and constraints (UAS dynamics, total-path length, line inter-distance, turn points, etc.) in order to obtain, via optimization procedures, an optimal trajectory. To this end we make use of polyhedral set operations, flat trajectory generation and other similar tools. Additional work includes the study of non-convex regions and estimation of the number of photographs taken via Ehrhart polynomial computations.**

***Index Terms*—digital photography, optimization, path planning, position control, unmanned aerial vehicles.**

I. INTRODUCTION

Among the methods for event detection, the interpretation of optical remote images is widely used and also gives the best results concerning price and accuracy. In order to detect events (like flood) by image analysis, three solutions usually appear in the literature: a) use of images from satellites [1-2], b) use of images from fixed cameras on the ground [3-4] and c) use of images from aircrafts or UAVs [5]. To monitor and evaluate the area of disasters, concatenated images, created by photomosaic generation, can be useful. Thus, the gaps or duplications of regions, in different analyzed images, are avoided. In this case, the UAV solution is a cheaper and more flexible one which can ensure superior image resolution even under adverse weather conditions. In this direction, the authors in [6] developed a solution for detection and evaluation of the dynamic evolution of the flood based on a collaborative team of UAVs.

More recently a multicopter-based photogrammetry procedure, was used to evaluate the effect of an earthquake on complex architectural landscapes [7]. Also, Feng et al. [8] used a UAV for urban flood monitoring thus showing that such platforms can provide accurate flood maps. In their proposed method, the authors show how the acquired images are ortho-rectified and combined into a single image.

All the above image acquisition strategies impose strict constraints on the photographs' capture during the UAV mission. I.e., photographs have to be captured: at a constant height (low/medium/high - the classification is relative, depending on context and application); such that there is a predefined overlap between neighboring photographs and there are no gaps in the area of interest (such that a photo-mosaic image covering the entire area is computed). While there are many specialized software applications which can merge photographs with partial overlap to generate a continuous mapping and detect features of interest, there are still open issues in the generation, control and optimization of the flight path to be followed by a UAV [9]. This apparently simple problem has a number of intricacies: turn maneuvers of the UAV should not "cut" into the shape of the area under observation, maximal distance between consecutive path lines has to be respected and, not in the least, the UAV operational costs (energy, time of travel) should be minimized [10-11].

Assuming that all low-level control loops are already designed such that a predefined trajectory is followed accurately and the payload is stabilized, we can reduce the path generation problem to an optimization problem where various constraints, parameters and costs are taken into account.

The constraints include inter-distance between consecutive path-lines, guarantees of shape coverage (i.e., through the addition of auxiliary turn points which assure that the UAV does not "cut" the shape under observation) and, if needed, course angle restrictions (if wind velocity is significant and the UAV's autopilot cannot accept a large deviation from the wind's vector or to limit brightness variations). The main parameter to be considered is the inter-distance between consecutive lines and directly depends on the photographs' overlap coefficient, usable area covered by each photograph and desired resolution (the last two directly depend on the UAV cruise altitude). Lastly, various different cost functions may be minimized such as fuel consumption, flight time, total path length, etc. Taking into account all these elements, we can formulate an optimization problem whose output is the optimal path which, for every set of given parameters, respects the constraint and minimizes the chosen cost function.

The first step is to define a convex shape over which the UAV has to fly. Moreover, the flight path needs to be chosen such that the photographs taken while flying cover the entire domain. The first issue of interest is the number of photographs which have to be taken. This can be assimilated to counting the number of integer points (each point represents a captured photograph) which lie inside a convex

The work has been funded by Program STAR, project 71/2013-MUROS and by Romanian National Authority for Scientific Research and Innovation, CNCS-UEFISCDI, project number PN-II-RU-TE-2014-4-2713.

Digital Object Identifier 10.4316/AECE.2016.03014

shape. These notions can be described formally using the Ehrhart polynomial and the associated theory [12].

Next, we discuss two alternative implementations of the constrained optimization problem. The first penalizes the length of the internal path (containing only the segments inside the shape under observation) and the second penalizes the entire path (including turn points). Lastly, we discuss the influence of wind [13] and the case of non-convex shapes (How should a non-convex area be covered such that the generated path is still optimal ?).

The main novelties introduced in the path planning procedure are the offline analysis (which estimates the number of photographs which will be captured) and the consideration of explicit constraints in the path planning procedure (both intrinsic - consecutive lines inter-distance and extrinsic - wind direction).

The problem that we are solving in this paper is the optimal coverage of the area to be monitored, from the point of view of energy consumption (a limiting factor, since the UAV is electrically powered trough a battery) and trajectory length. This problem is solved through new solutions concerning the path generation. Here, as a proxy for fuel consumption, we consider the total path-length. In order to create a path which respects the UAV dynamics we use flat trajectory constructions. This may prove to be useful when the area under coverage is short enough such that the UAV dynamics become significant.

As support for aerial surveillance we propose a fixed-wing type UAV system developed by the authors in the project MUROS [14] (Multisensory Robotic System for Aerial Monitoring of Critical Infrastructures), granted by the Romanian National Research Program STAR (Space Technology and Advanced Research) from ROSA (Romanian Space Agency). The system is completely autonomous, apart from the take-off stage where a human operator is needed. The area to be monitored is covered with the aid of a path, designed by a suitable optimization problem while the acquired images are analyzed in order to detect and assess the extent of disasters.

II. Method for Optimal Area Coverage

To automatically evaluate the extent of disasters in the monitored area, a single image without overlays or gaps is necessary. The camera mounted on the UAV takes photos which exhibit partial overlap and the photogrammetry algorithm has to process the data and output a continuous image which is then partitioned into smaller, non-overlapping images which cover the observation area and can later be used by the segmentation algorithm.

The photographs used for the segmentation procedure are captured along a path covering the area under surveillance. The resulting path planning procedure takes into account design constraints and various cost functions which lead to a constrained optimization problem. This is a typical approach in path/trajectory planning procedures [15] for either photogrammetric [10], [13] or other flight missions [9], [11]. In its simplest form, it means that an agent (i.e., the UAV) has to travel optimally across a predefined area such that a certain objective is fulfilled (e.g., observation of the area).

The first step is to provide the bounds of the region of interest (where the flooded area is to be detected). This is done by the operator and may need to be re-assessed during the area coverage stage if the extent of the flooded area is larger than initially believed. Further we propose a constrained optimization problem which accounts for both intrinsic (e.g., distance between consecutive lines of the path, orientation of the UAV with respect to the area) and extrinsic constraints (e.g., wind direction). Lastly, the influence of external parameters and the extension to non-convex areas are discussed. Additional constraints which describe UAV and payload limitations (UAV speed, camera snapshot speed, brightness variation, etc) have not been taken into account here but can certainly be added in a future work.

As a framework we consider polyhedral sets which are a good choice for practical implementations. They can approximate arbitrarily well any convex shape and apply naturally to constrained optimization problems. This comes from their dual (half-space/vertex) representation [16].

Let us consider a collection of vertices $\{v_i\}$ which define a polyhedral shape:

$$P = \left\{ x \in \Re^n : x = \sum_i \alpha_i v_i, \sum_i \alpha_i = 1, \alpha_i \in \Re_+ \right\} \quad (1)$$

to which corresponds the dual half-space representation

$$P = \left\{ x \in \Re^n : h_i^T x \le k_i, (h_i, k_i) \in \Re^n \times \Re \right\} \quad (2)$$

Note that x stands in both equations (1) and (2) as a variable which is inside of the polyhedron P. Under representation (1) this means that x can be expressed as a convex sum of the extreme vertices v_i of the polyhedron (via the weights α_i) and under representation (2) it means that x verifies all the constraints defining the polyhedron (given by the pairs (h_i, k_i)).

A. Interior points

An interesting question is how many snapshots have to be taken such that a pre-defined area is completely covered. We neglect partial overlapping and assume that some existing photogrammetry algorithm can be used to stitch together all the images in order to obtain a continuous mapping of the terrain. Assuming that each photo covers a square region of width 100 m the problem reduces to counting how many integer points (i.e., which have integer coordinates) are found inside the polyhedral set characterizing the area under observation. Formally this number is characterized by the Ehrhart polynomial (3):

$$L(P,t) = \left\{ x \in \mathrm{Z}^n : x \in tP \right\} \quad (3)$$

which gives the number of integer points which lie inside or on the boundary of polytope tP - the t-fold dilation of P. Factor t serves here as a proxy for the operating altitude of the UAV. If the altitude is increased, the area covered by photo increases as well. This is equivalent with multiplying P with a sub-unitary scaling factor. The computation of (3) is done through specialized tools which exploit the associated theory [12]. The actual computation of the Ehrhart polynomial coefficients is done with the Latte package [17-18].

B. Path generation

While the use of the Ehrhart polynomial gives a

qualitative understanding of the task ahead (i.e., many interior points means a long path), the actual cost which has to be minimized is the length of the path taken by the UAV (hereafter we simply assume that the photographs are taken along straight lines at predefined time intervals). The length of the path (for now we neglect the parts exterior to the polyhedral shape) consists of parallel line segments whose end points lie on the polytope boundary and which are not separated by more than *d* units (we consider that the photos taken along the path cover a square of dimension *d*×*d* units).

Let us assume that the path lines are parallel with the horizontal (always possible through a suitable rotation) and that the *i*-th segment is defined by the pair of end-points $(\underline{x}(y_i), y_i), (\bar{x}(y_i), y_i)$. Then we can write the optimization problem (5) which returns the set of vertical coordinates y_i:

$$\sum_i (\bar{x}(y_i) - \underline{x}(y_i)) = \min \tag{5a}$$

$$s.t.\ \underline{y} \le \ldots \le y_i \le y_{i+1} \le \ldots \le \bar{y} \tag{5b}$$

$$y_{i+1} - y_i \le d \tag{5c}$$

$$y_1 - \underline{y} \le \frac{d}{2}, \bar{y} - y_N \le \frac{d}{2} \tag{5d}$$

where:

- the sum from (5a) is the cost which sums the segment lengths. Note that because the end-points of the segment have the same height, we can simplify the distance to a linear term.

- (5b) forces the variables y_i to increase monotonically between the inferior vertical bound $\underline{y} = \min_{\xi \in P} [0 \quad 1]^T \xi$ and the superior vertical bound $\bar{y} = \max_{\xi \in P} [0 \quad 1]^T \xi$ of $\underline{P}$ (i.e., the minimum and the maximum of the projection of *P* on the vertical axis).

- (5c) and (5d) force that two consecutive segments are not further apart than *d* and that the first and respectively the last line from the sequence are not further away from the polytope than $\frac{d}{2}$.

- Mappings $\underline{x}, \bar{x} : [\underline{y}, \bar{y}] \to \Re$ denote the left-most and, respectively, right-most value of the x-coordinate of a point from inside *P* which has the y-coordinate fixed.

- *N* denotes the number of parallel lines composing the path.

Note that *N* has to be computed a priori. Since we wish to cover the entire polytope, the value is obtained from (6):

$$(N-1) \cdot d + 2 \cdot \frac{d}{2} \ge \bar{y} - \underline{y} \tag{6}$$

The constraint in (6) states that for *N* lines, the *N-1* inter-line distances have to be less or equal to *d* and that the two end lines (the first and last) have to be less than *d*/2 from the end-points of the polytope. This comes from noting that along each line, the photos taken cover *d*/2 on the top part and *d*/2 on the bottom part.

Mappings $\underline{x}, \bar{x}$ partition the boundary of the polytope into two subsets: the one visible from the left and the one visible from the right. Hence, these mappings are piecewise affine (as each of them is a collection of facets of the polytope) and need to be reformulated into an optimization problem solvable with existing software. Specifically, they can be modeled via mixed integer programming techniques which lead to an MILP (mixed integer linear problem) which has to be solved with specialized solvers (e.g., Gurobi, CPLEX etc.).

1) Remark

The above discussions hold for the general case of higher dimensions. Mappings $\underline{x}, \bar{x}$ can be seen as lower and upper level projections along a subspace with domain defined over the projection of the polytope on the perpendicular subspace.

The external path has, in addition to the previously discussed components, external points which serve as guidance return points for the UAV. That is, for *i* odd, after following the line $\underline{x}(y_i), \bar{x}(y_i)$, the UAV will go towards $(\bar{x}(y_i) + d_e, y_i)$. Similarly, for *i* even, after following line $\underline{x}(y_i), \bar{x}(y_i)$, the UAV will go towards $(\underline{x}(y_i) + d_e, y_i)$. These external steps permit to change direction and approach in a timely manner the next line $(\underline{x}(y_{i+1}), \bar{x}(y_{i+1}))$. Using the notation of (5a)-(5d) we can then write optimization problem (7):

$$\begin{aligned} &\sum_i (\bar{x}(y_i) - \underline{x}(y_i)) + \\ &\sum_{i,i\ odd} dist[(\bar{x}(y_i) - (\bar{x}(y_{i+1}) + d_e, y_{i+1})] + \\ &\sum_{i,i\ even} dist[\underline{x}(y_i), y_i) - (\underline{x}(y_{i+1}) - \\ &d_e, y_{i+1})] = \min \end{aligned} \tag{7a}$$

$$s.t.\ \underline{y} \le \ldots \le y_i \le y_{i+1} \le \ldots \le \bar{y} \tag{7b}$$

$$y_{i+1} - y_i \le d \tag{7c}$$

$$y_1 - \underline{y} \le \frac{d}{2}, \bar{y} - y_N \le \frac{d}{2} \tag{7d}$$

The modification of the optimization problem with respect to (5a)-(5d) appears in the design of the cost. Since we also take into account the external parts, we need to add them into the cost (the 2nd and 3rd terms respectively). Note that at this point the problem becomes an MIQP (mixed integer quadratic problem) since the cost has a quadratic component (i.e., we can no longer reduce the distance to a linear term as the end-points no longer have the same height).

2) Remark

We note that the current approach minimizes the path length but other criteria are of course possible. Typical choices are the flight time and energy minimization. These depend on the internal model of the UAV (which is nonlinear and parametrized after a multitude of factors). For the current choice of the cost (path length) we simplified the model by assuming that the UAV will stay close to the reference path through suitably chosen auto-pilot control loops.

3) Remark

We have assumed that the area under observation is a convex shape. It might very well be the case that the area is non-convex. A possible solution is the construction of the

convex hull of the area (and then the snapshots taken outside the area are discarded at the processing stage). Otherwise, we may decompose the region into a union of convex shapes and apply the optimization procedures from above to each of them separately. In order to reduce the total length of the path we may consider as an additional factor, the rotation angle for the lines followed by the UAV. Strictly speaking, this would introduce an additional level in the optimization problem where at the upper-level the rotation angle θ would be the optimization variable. To avoid this, we consider an iterative procedure where we take successive values of θ and solve problem (5a)-(5d) (or (7a)-(7d)), as shown in Algorithm 1. Note that the mappings $\underline{x}, \overline{x}$ will have to be recomputed for each different θ. The output of the algorithm will be the pair (l^*, θ^*), which denotes the minimum path length and the angle for which it is obtained, respectively.

```
Algorithm: Search after rotation angle
Input: P, d, δθ, θ_w, Δθ
Output: θ*, l*
1.  θ = Δθ - θ_w;
2. while θ < Δθ + θ_w do
3.     θ = θ + δθ;
4.     solve problem (5a)-(5d) (or (7a)-
       (7d)) for
       P→R(θ)P and obtain the path length
       l(θ);
5.     if l* > l(θ) then
6.           l* = l(θ), θ* = θ;
7.     end
8. end
```

$R(\theta)$ stands for the rotation matrix, which in $\Re^2$ is:

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

$\delta\theta$ denotes the angle increment with which is the current angle value θ is modified at each step in the algorithm.

4) Remark

Searching over the entire interval [0°, 360°] might not be suitable. For example, in the presence of wind we may not accept a deviation of the line path greater than a $\Delta\theta$ from the wind direction θ_w (the maximum amount which the auto-pilot can handle). Thus, the algorithm would have to consider a feasible search interval $(\theta_w - \Delta\theta, \theta_w + \Delta\theta)$.

C. Considerations on UAV dynamics

Up to this point the problem discussed was exclusively geometric in nature with the underlying dynamics of the UAV completely ignored. If the straight line lengths and the turn radius are relatively short then the dynamics become relevant and have to be taken into account.

To this end, we consider the dynamics of a 2D 3-DOF model of an airplane (8) in which the autopilot forces coordinated turns (zero side-slip) at a fixed altitude [19]:

$$\begin{aligned} \dot{x}(t) &= V_a(t)\cos\Psi(t) \\ \dot{y}(t) &= V_a(t)\sin\Psi(t) \\ \Psi(t) &= \frac{g\tan\Phi(t)}{V_a(t)} \end{aligned} \tag{8}$$

The state variables are represented by the position $(x(t),y(t))$ and the heading (yaw) angle $\Psi(t) \in [0, 2\pi]$ rad. The input signals are the airspeed velocity $V_a(t)$ and the bank (roll) angle $\Phi(t)$, respectively.

Also, the airspeed and the bank angle are regarded as the autopilot pseudo-controls. In other words, we assume that control loops exist at the lower level and that they can handle any set point sent from the higher control level. A refining of the model (not followed here) is to assume that these low-level controllers correspond, in closed-loop form, to first and second order systems.

We denote $z(t) = [z_1(t) z_2(t)]^T = [x(t) y(t)]^T$ as the flat outputs of dynamics and compute the remaining variables in terms of these outputs and their derivatives (9):

$$\begin{aligned} \Psi(t) &= \arctan\left(\frac{\dot{z}_2(t)}{\dot{z}_1(t)}\right), \\ V_a(t) &= \sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)}, \\ \Phi(t) &= \arctan\left(\frac{1}{g}\frac{\ddot{z}_2(t)\dot{z}_1(t) - \dot{z}_2(t)\ddot{z}_1(t)}{\sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)}}\right) \end{aligned} \tag{9}$$

Variables $z_1(t)$ and $z_2(t)$ are called "flat outputs" and describe both the state and the input independently (that is, the dynamic link between state and input is hidden in the description of the flat outputs) [20-22].

It suffices then to find the flat outputs which validate given constraints (in our case the UAV *x,y* components of the states have to pass through the way-points determined a priori) and to introduce the results back into (9). Note that in general it is difficult to obtain a flat output $z(t)$ which respects state and input constraints. Henceforth, the output is parametrized after some basis functions (10):

$$z(t) = \sum_{i=1}^{N} \alpha_i \Lambda^i(t), \quad \alpha_i \in R \tag{10}$$

Assuming that the basis function is known (polynomial, Bezier, B-spline functions, etc) it only remains to find the coefficients α_i which respect the constraints of interest.

Due to their properties, in what follows we will consider B-spline basis functions to characterize (10). They are given by a recurrent construction which takes a knot-vector (11)

$$\mathbb{T} = \{\tau_0, \tau_1, \ldots, \tau_m\}, \tag{11}$$

of monotonously increasing time instants such that the $\underline{i}$-th function of degree d is given by (12):

$$\begin{aligned} B_{i,1}(t) &= \begin{cases} 1, & \text{for } \tau_i \le t < \tau_{i+1} \\ 0, & \text{otherwise} \end{cases}, \\ B_{i,d}(t) &= \frac{t-\tau_i}{\tau_{i+d-1}-\tau_i} B_{i,d-1}(t) + \frac{\tau_{i+d}-t}{\tau_{i+d}-\tau_{i+1}} B_{i+1,d-1}(t) \end{aligned} \tag{12}$$

For $d > 1$ and $i = 0,1,\ldots,n = m-d$.

Considering a collection of control points (13):

$$P = [p_0, \ldots, p_n] \tag{13}$$

we define a B-spline curve (14) as a combination of these control points and the B-spline basis functions $B_d(t) = [B_{0,d}(t), \ldots, B_{n,d}(t)]^T$:

$$z(t)=\sum_{i=0}^{n}B_{i,d}(t)p_i=PB_d(t) \quad (14)$$

Further details are to be found in, e.g., [26-27].

Taking a collection of way-points and the time stamps (15) associated to them:

$$\mathbb{W}=\{w_k\}\text{ and }\mathbb{T}_{\mathbb{W}}=\{t_k\}, \quad (15)$$

for any $k=0,\ldots,N$ we can construct a flat trajectory (16) which passes "near" each way-point w_k at the time instant t_k, i.e., find a flat output $z(t)$ such that

$$x(t_k)=\Theta(z(t_k),\ldots z^{(r)}(t_k))\in\{w_k\}\oplus S_k, \quad (16)$$

for any $k = 0,\ldots,N$ and where S_k is an a priori defined *sensing region* around the k-th way-point.

Note that the constraints usually appear in the input or state space (as is the case in (16)). Hence, these have to be translated into the flat output space. The mapping from one to another is not necessarily straightforward. Here, the B-splines parametrization is of help due to their geometrical properties. In particular, we note that derivatives of (10) translate into B-spline parametrizations of lower-order which can be then rewritten as combinations of higher order B-splines (via the matrices M_r, L_r used in the next equations). In our case, this means that the way-point (16) becomes (17):

$$\tilde{\Theta}(\mathrm{B}_d(t_k),\mathrm{P})\in\{w_k\}\oplus S_k,\forall k=0\ldots N, \quad (17)$$

with $\tilde{\Theta}(\mathrm{B}_d(t),\mathrm{P})=\Theta(\mathrm{PB}_d(t),\ldots,\mathrm{P}\,M_rL_rB_d(t))$ a shorthand notation which highlights the fact that the various mappings described in (8) and (9) can be always expressed in terms of the control points (13). Hence, even if the mapping is nonlinear, it is still possible to handle it through a nonlinear solver.

The same B-spline properties can be used to write a cost $\tilde{\Xi}(\mathrm{B}_d(t),\mathrm{P})$ which minimizes path length [28], or indeed any other cost which penalizes states and inputs.

With these elements it is then straightforward to write the (possibly nonlinear) optimization problem (18) which provides as result a collection of control points:

$$P=\arg\min_{\mathbf{P}}\int_{t_0}^{t_N}\|\tilde{\Xi}(\mathrm{B}_d(t),\mathrm{P})\|\,dt \quad \text{s.t. constraints (17) are verified} \quad (18)$$

which uniquely characterize the flat output (10) and hence, the dynamics (8-9).

Some remarks are presented below:

1) Remark

The sensing region S_k can be manipulated to account for the constraints of the photogrammetry scheme as it allows relaxing selectively the constraints (16). For example, for the internal way-points, through which we wish to pass, the region can be taken as the empty set (i.e., $S_k=0$) whereas for the external way-points (i.e., those outside of the active region), where we are content to pass "near" but not necessarily "through" way-points, the region can be taken as some constant shape defined a priori (i.e., $S_k=S^*$).

2) Remark

Last but not least, the basis functions have to be properly chosen. First, their number should be taken such that the problem actually has a solution (too few coefficients α_i for a given set of constraints and the problem may become infeasible). Next, the degree of the basis functions has to be sufficiently large to assure continuity for the states and inputs. For example, if we wish to have smooth inputs in the roll angle component $\Phi(t)$ it means that we need to have $z(t)$ continuous up to its 4th derivative (since a function is smooth if its second derivative is continuous and in (9) terms $\ddot{z}_1(t)$ and $\ddot{z}_2(t)$ appear which means that $\ddot{z}(t)$ itself has to be twice continuously derivable).

III. Results

In the previous section we introduced the notion of a lattice of integer points superposed over a polytopic set. These points represent the points which we wish to photograph and the dilation factor t parametrizes the effective width of the photo (it may change as a function of UAV altitude, desired degree of superposition between neighboring photos, etc).

Considering a polytope P defined by the extreme points {(-2,4),(2,3),(3,-1),(-1,-2),(-3,0)}×100 m (Fig. 1), the associated Ehrhart polynomial (4) is obtained by computing (via the aforementioned Latte toolbox):

$$L(P,t)=25t^2+3t+1 \quad (19)$$

With this analytic description we can provide the number of points which are covered by the given shape for various values of parameter t. For example, at $t=1/100$ we have 29 points inside P. Note that larger values of t lead to more interior points (scaling P with t is equivalent with decreasing with $1/t$ the inter-distance between points in the lattice). Conversely, smaller values of t lead to less points.

The integer points inside and outside of the polyhedral set are drawn in Fig. 1 as solid red and empty markers respectively. In the magnified detail we illustrate the area covered by each of these points.

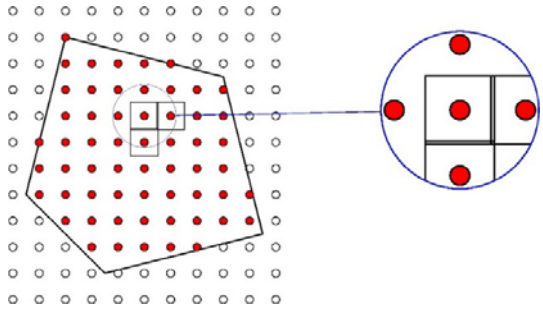

Figure 1. Illustration of snapshot counting

Using the same example as before, we illustrate the results of the proposed method. First, in Fig. 2 we show the left-most and right-most mappings $\underline{x},\overline{x}$.

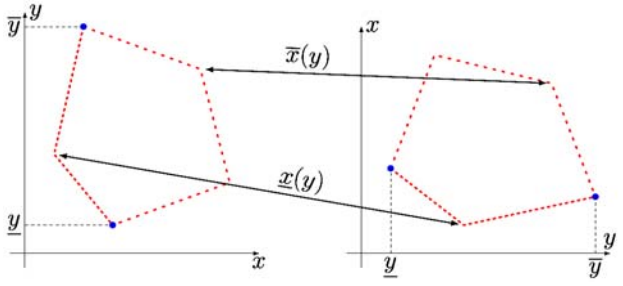

Figure 2. Illustration of mappings $\underline{x}(y),\overline{x}(y)$

Each of these mappings is piecewise affine and for our example they are:

$$\underline{x}(y)=\begin{cases}-\dfrac{4y}{5}-3 & y\in(-2.5,0)\\ -\dfrac{y}{2}-3 & y\in(0,4.5)\end{cases},$$

$$\overline{x}(y)=\begin{cases}\dfrac{8y}{11}+\dfrac{9}{11} & y\in(-2.5,-1)\\ -\dfrac{y}{4}+\dfrac{11}{4} & y\in(-1,3)\\ -\dfrac{8y}{3}+10 & y\in(3,4.5)\end{cases} \quad (20)$$

These, together with the rest of the constraints and the auxiliary binary variables used to model the PWA (piecewise affine) functions are used to solve (5a)-(5d) and (7a)-(7d) respectively. The resulting solution (the points $(\underline{x}(y_i),\overline{x}(y_i),y_i)$ is depicted in Fig. 3 which illustrate the inner and external path, respectively.

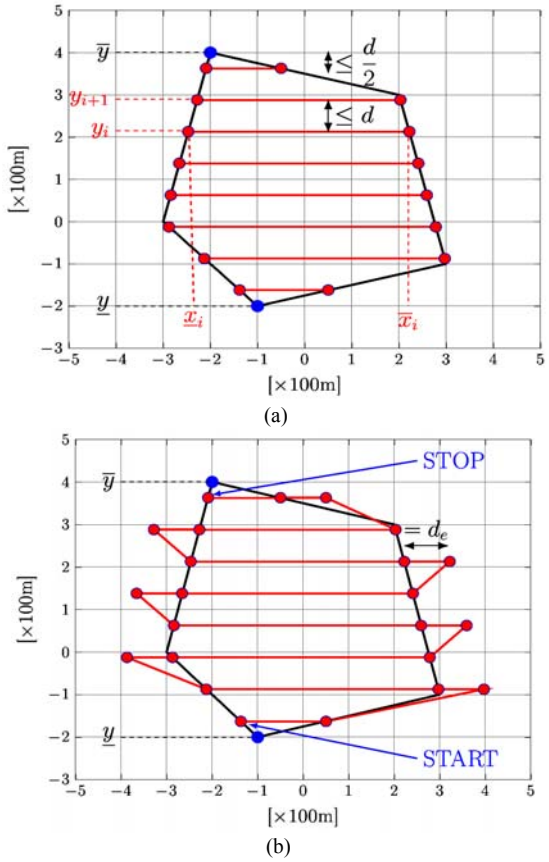

Figure 3. Illustration for optimal path

The result of solving (5a)-(5d) with parameter d = 75 m is depicted in Fig. 3(a) and has length 3371.88 m. The result of solving (7a)-(7d) with parameters d = 75 m, $d_e = 100$ m is depicted in Fig. 3(b) and has a total of length 4529.96 m.

Further, we search for the optimal angle which minimizes the total path length. We use Algorithm 1 with starting parameters $\Delta\theta = 180^o, \theta_w = 180^o, \delta\theta = 5^o$. In Fig. 4(a) we illustrate the internal path (as obtained from (5a)-(5d)) and the total path (as obtained from (7a)-(7d)) as functions of rotation angle θ. The average computation time for both problems was 0.9713 sec and respectively 0.9999 sec. While these times may vary with hardware, solvers and problem instance, they indicate that solving these optimization problems is fast and does not pose great difficulties.

The minimum, under (7a), is reached at value θ^* = -105° corresponding to a total path l^* = 4020.18 m (Fig. 4(b)).

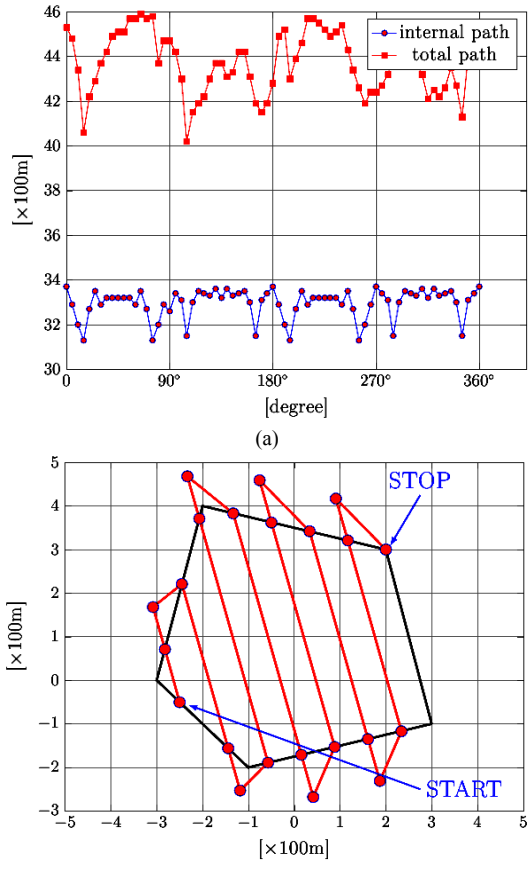

Figure 4. Illustration of optimal path computations

We also analyze the situation in which the area is non-convex. There are two ways in which we may consider the problem: i) we compute the convex hull of the area and apply the optimization problem from before (and ignore the snapshots which are outside); ii) we decompose the area in a union of convex shapes and compute the paths for each shape separately. Note that in the latter case we have to consider the lengths between successive starting and stopping points.

As an illustrative example, a non-convex area composed of three convex shapes is considered (Fig. 5) and the two approaches mentioned above are applied. In Fig. 5(a) we first compute the convex hull (dashed line) and obtain a total path length of 9517.91m whereas in Fig. 5(b) we compute total path lengths for each of the shapes (1369.12 m, 2774.27 m and 2304.19 m), consider the lengths between the stop and start path for the pair of first/second and second/third shapes (645.30 m and 655.02 m) and obtain a total path length of 7747.9 m. Note that the second approach gives (at least for this example) a shorter total path than the first approach. An improvement, not treated here would be to decide the direction of covering the path such that the links between the shapes are minimized (observe that the distances between $STOP_1$ – $START_2$ and $STOP_2$ – $START_3$ would have been shorter if we would have switched $START_2$ and $STOP_2$ between themselves.

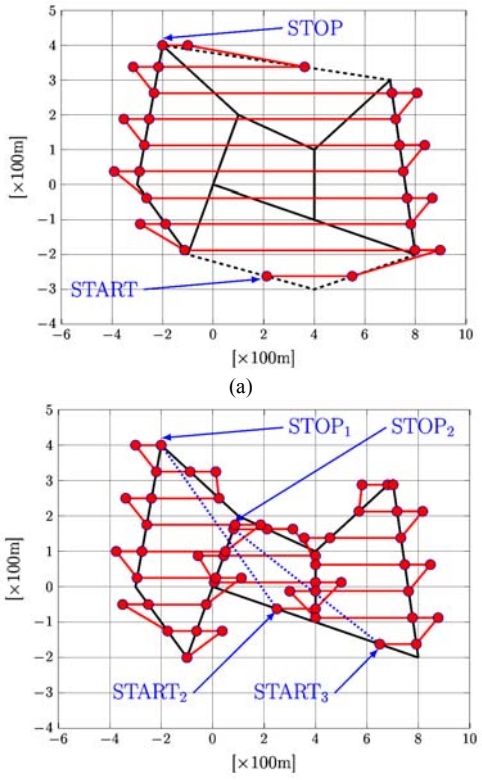

Figure 5. Illustration of optimal path computations for non-convex shapes

As illustrated by the two figures, in this particular case, dividing the region into smaller ones pays of, since we obtain a smaller total length. However, this may not hold for any non-convex shape.

In Fig. 6 we show a couple of line-paths depicted over a real terrain feature (with different rotation angles).

For UAV dynamics, we consider a B-spline parameterization which has numerical and algorithmic advantages with regards to polynomial or Bezier parameterizations [22-23]. We consider an optimization problem where we minimize the total-path length of a B-spline curve which passes thorough N a priori given way-points and has N+1 control points while also respecting UAV dynamics (8). The resulting flat outputs (which are also the *x, y* components of the state) are depicted in Fig. 7a) as the solid red line. The dashed blue line represents the control point polygon which bounds the curve.

Figure 6. Line path generation over a real terrain feature

The total length of the path is 33.30 × 100 m. An improvement, making use of results described in [23] is to relax the constraints such that the trajectory has only to pass through a neighbourhood of them. This cannot be applied to the internal way-points, only to the external ones $(\underline{x_i}(y_i)-d_e, y_i)$ and $(\overline{x_i}(y_i)+d_e, y_i)$.

As it can be seen, the external points are no longer reached rather the trajectory will pass through a neighbourhood of them (Fig. 7b). The total path-length is 31.17 × 100 m, a slight improvement with respect to the exact case.

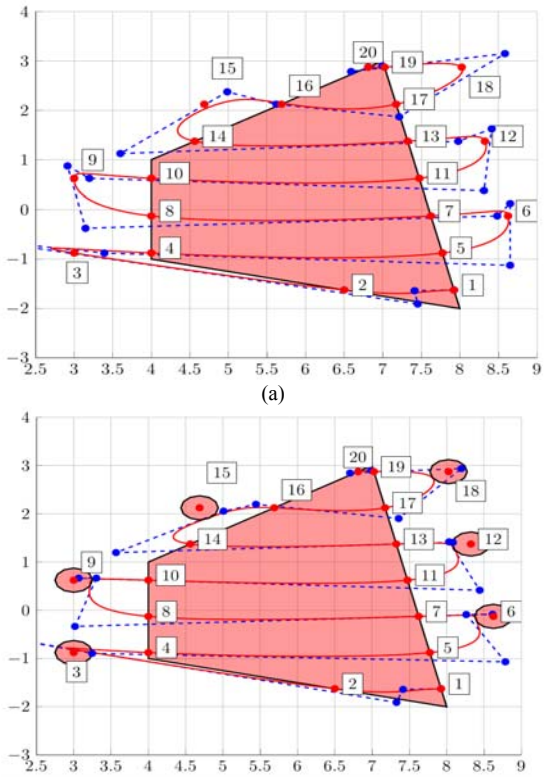

Figure 7. Real trajectory under UAV dynamics: (a) flat trajectory with exact constraints, (b) flat trajectory with relaxed constraints

In Fig. 8 the remaining state $\Psi(t)$ - the course angle and the inputs $V_a(t)$ and $\Phi(t)$ are depicted (by introducing $z_1(t)$ and $z_2(t)$ into relations (9)).

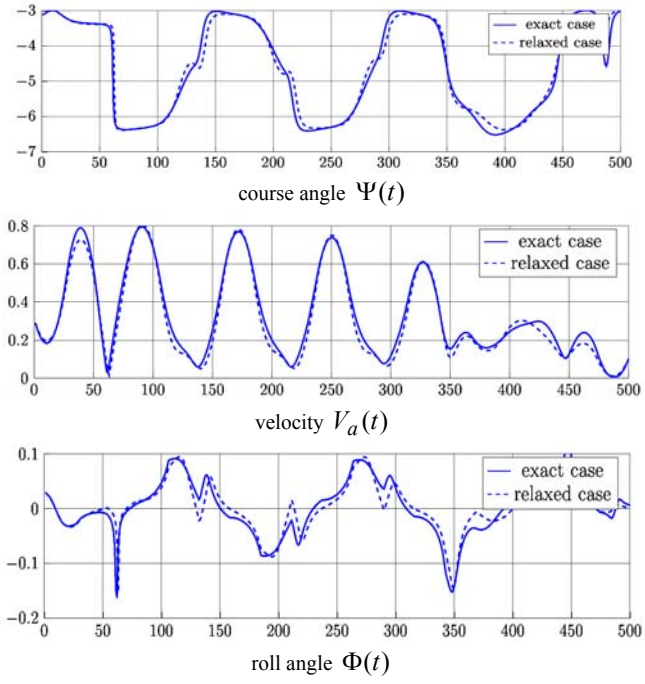

Figure 8. State and input values for UAV dynamic

The dashed lines represent the state and inputs corresponding to this relaxed case (whereas the solid lines correspond to the exact case). It can be seen that the extremes are less pronounced (accordingly, less stress for the UAV and less fuel consumption).

To solve the problems and illustrate the results, we have used the Yalmip [24] and MPT3 [25] toolboxes in the Matlab environment on a computer with 2.2 Ghz CPU and 8 Gb of RAM memory. While the computation times are relative to the implementation and particularities of the hardware, we point that the computation times are reasonable and allow "on-the-fly" computation of the optimal path.

Future directions may consider fuel consumption as a cost to be minimized, bounds on the UAV velocity and the integration of the flat trajectory computation into the way-point computation. In particular, keeping the velocity constant is of great importance as many photogrammetry schemes do not estimate/use GPS information to retrieve the UAV position and hence assume a constant velocity and take photos at constant time intervals.

IV. Conclusion

The paper studied a typical photogrammetry problem through the prism of control and optimization theory. The novelty lies in the analysis and computation of an optimal path covering the area of interest. That is, for a given polyhedral region which has to be covered by parallel lines (along which photographs are taken) we have given both an estimation of the required number of photographs and provided a minimum-length path covering the area. For the latter case we formulated a constrained optimization problem where various constraints and parameters were considered in order to obtain a minimum-length path. We took into account the maximum distance between consecutive lines and turn conditions (such that the UAV is guaranteed to follow the interior lines). We have also discussed the path generation problem in the presence of wind and for regions with non-convex shapes. Lastly, we took into account simplified UAV dynamics in order to compute a feasible trajectory which passes through the points defined earlier. We further relaxed the problem by considering neighborhoods centered on the external turn points.

References

[1] R. K. Pandey, J.-F. Cretaux, M. Berge-Nguyen, V. M. Tiwari, V. Drolon, F. Papa, S. Calmant, "Water level estimation by remote sensing for the 2008 flooding of the Kosi river," Int. J. Remote Sens., vol. 35, no. 2, pp. 424–440, 2014. doi: 10.1080/01431161.2013.870678

[2] H. Khurshid, M. F. Khan, "Segmentation and Classification Using Logistic Regression in Remote Sensing Imagery," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 8, no. 1, pp. 224–232, 2015. doi: 10.1109/JSTARS.2014.2362769

[3] R. Koschitzki, E. Schwalbe, H. Maas, "An autonomous image based approach for detecting glacial lake outburst floods," ISPRS-Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. 1, pp. 337–342, 2014. doi:10.5194/isprsarchives-XL-5-337-2014

[4] S.-W. Lo, J.-H. Wu, F.-P. Lin, C.-H. Hsu, "Cyber surveillance for flood disasters," Sensors, vol. 15, no. 2, pp. 2369–2387, 2015. doi:10.3390/s150202369

[5] J.-N. Lee, K.-C. Kwak, "A trends analysis of image processing in unmanned aerial vehicle," Int. J. Comput. Inf. Sci. Eng., vol. 8, no. 2, pp. 261–264, 2014.

[6] M. Abdelkader, M. Shaqura, C. G. Claudel, W. Gueaieb, "A UAV based system for real time flash flood monitoring in desert environments using Lagrangian microsensors," in International Conference on Unmanned Aircraft Systems (ICUAS), 2013, pp. 25–34. doi: 10.1109/ICUAS.2013.6564670

[7] C. Achille, A. Adami, S. Chiarini, S. Cremonesi, F. Fassi, L. Fregonese, L. Taffurelli, "UAV-based photogrammetry and integrated technologies for architectural applications-methodological strategies for the after-quake survey of vertical structures in Mantua (Italy)," Sensors, vol. 15, no. 7, pp. 15520–15539, 2015. doi:10.3390/s150715520

[8] Q. Feng, J. Liu, J. Gong, "Urban flood mapping based on Unmanned Aerial Vehicle remote sensing and random forest classifier-A case of Yuyao, China," Water, vol. 7, no. 4, pp. 1437–1455, 2015. doi: 10.3390/w7041437

[9] S. Siebert, J. Teizer, "Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system," Autom. Constr., vol. 41, pp. 1–14, 2014. doi:10.1016/j.autcon.2014.01.004

[10] H. Eisenbeiss, M. Sauerbier, "Investigation of UAV systems and flight modes for photogrammetric applications," Photogramm. Rec., vol. 26, no. 136, pp. 400–421, 2011. doi: 10.1111/j.1477-9730.2011.00657.x

[11] K. J. Obermeyer, "Path planning for a UAV performing reconnaissance of static ground targets in terrain," in AIAA Guidance, Navigation, and Control Conference, pp. 10–13, 2009. http://dx.doi.org/10.2514/6.2009-5888

[12] R. Diaz, S. Robins, "The Ehrhart polynomial of a lattice polytope," Ann. Math., vol. 145, no. 3, pp. 503–518, 1997. doi: 10.2307/2951842

[13] B. Ruzgiene, T. Berteska, S. Gecyte, E. Jakubauskienė, V. C. Aksamitauskas, "The surface modelling based on UAV Photogrammetry and qualitative estimation," Measurement, 2015. doi:10.1016/j.measurement.2015.04.018

[14] D. Popescu, L. Ichim, T. Caramihale, "Flood areas detection based on UAV surveillance system, 19th International Conference on System Theory, Control and Computing (ICSTCC), pp. 753–758, 2015. doi: 10.1109/ICSTCC.2015.7321384

[15] S. M. Adams, C. J. Friedland, "A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management," in 9th International Workshop on Remote Sensing for Disaster Response, 2011.

[16] T. Motzkin, H. Raiffa, G. Thompson, R. Thrall, "The double description method," Contrib. Theory Games, vol. 2, pp. 51, 1959.

[17] V. Baldoni, N. Berline, M. Koeppe, M. Vergne, "Intermediate sums on polyhedra: computation and real ehrhart theory," Mathematika, vol. 59, no. 01, pp. 1–22, 2013. doi: 10.112/S0025579312000101

[18] V. Baldoni, N. Berline, J. De Loera, B. Dutra, M. Koppe, S. Moreinis, G. Pinto, M. Vergne, J. Wu, A user's guide for LattE integrale v1. 7.2. 2014.

[19] I. Prodan, S. Olaru, R. Bencatel, J. B. De Sousa, C. Stoica, S.-I. Niculescu, "Receding horizon flight control for trajectory tracking of autonomous aerial vehicles," Control Eng. Pract., vol. 21, no. 10, pp. 1334–1349, 2013. doi:10.1016/j.conengprac.2013.05.010

[20] M. Fliess, J. Levine, P. Martin, P. Rouchon, On Differentially Flat Nonlinear Systems, Nonlinear Control Systems Design. Pergamon Press, 1992.

[21] J. Levine, Analysis and Control of Nonlinear Systems: A Flatness-based Approach. Springer Science & Business Media, 2009.

[22] F. Suryawan, "Constrained Trajectory Generation and Fault Tolerant Control Based on Differential Flatness and B-splines," Newcastle University, 2010.

[23] J. Lofberg, "YALMIP : A Toolbox for Modeling and Optimization in MATLAB," in Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.

[24] M. Herceg, M. Kvasnica, C. N. Jones, M. Morari, "Multi-Parametric Toolbox 3.0," in Proc. of the European Control Conference, Zurich, Switzerland, 2013, pp. 502–510.

[25] F. Stoican, I. Prodan, D. Popescu, "Flat trajectory generation for way-points relaxations and obstacle avoidance," 23th Mediterranean Conference on Control and Automation (MED), pp. 695–700, 2015. doi: 10.1109/MED.2015.7158827

[26] W. Gordon, R. Riesenfeld, "B-spline curves and surfaces," Computer Aided Geometric Design, pp. 95–126, 1974.

[27] N. Patrikalakis, T. Maekawa, Shape Interrogation for Computer Aided Design and Manufacturing. Springer Science & Business, 2010.

[28] F. Stoican, D. Popescu, "Trajectory generation with way-point constraints for UAV systems."Advances in Robot Design and Intelligent Control, pp. 379–386, 2016. doi: 10.1007/978-3-319-21290-6_38.