Loughborough University

This item was submitted to Loughborough's Research Repository by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Decomposition-based mission planning for fixed-wing UAVs surveying in wind

PLEASE CITE THE PUBLISHED VERSION

https://doi.org/10.1002/rob.21928

PUBLISHER

Wiley Periodicals, Inc.

VERSION

VoR (Version of Record)

PUBLISHER STATEMENT

This is an open access article under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

LICENCE

CC BY 4.0

REPOSITORY RECORD

Coombes, Matthew, Tom Fletcher, Wen-Hua Chen, and Cunjia Liu. 2019. “Decomposition-based Mission Planning for Fixed-wing Uavs Surveying in Wind”. Loughborough University. https://hdl.handle.net/2134/10318751.v1.

Received: 2 May 2019 | Revised: 14 November 2019 | Accepted: 1 December 2019

DOI: 10.1002/rob.21928

Check for updates

WILEY

REGULAR ARTICLE

Decomposition-based mission planning for fixed-wing UAVs surveying in wind

Matthew Coombes | Tom Fletcher | Wen-Hua Chen | Cunjia Liu

Department of Automotive and Aeronautical Engineering, Loughborough University, Loughborough, UK

Correspondence
Matthew Coombes, Department of Automotive and Aeronautical Engineering, Loughborough University, Loughborough, LE11 3TU UK.
Email: M.J.Coombes@lboro.ac.uk

Funding information
Science and Technology Facilities Council, Grant/Award Number: ST/N006852/1

Abstract

This paper presents a new method for planning fixed-wing aerial survey paths that ensures efficient image coverage of a large complex agricultural field in the presence of wind. By decomposing any complex polygonal field into multiple convex polygons, the traditional back-and-forth boustrophedon paths can be used to ensure coverage of these decomposed regions. To decompose a complex field in an efficient and fast manner, a top-down recursive greedy approach is used to traverse the search space to minimize the flight time of the survey. This optimization can be computed fast enough for use in the field. As wind can severely affect flight time, it is included in the flight time calculation in a systematic way using a verified cost function that offers greatly reduced survey times in the wind. Other improved cost functions have been developed to take into account real-world problems, for example, No-Fly Zones, in addition to flight time. A number of real surveys are performed to show the flight time in wind model is accurate, to make further comparisons to previous techniques and to show that the proposed method works in real-world conditions providing total image coverage. A number of missions are generated and flown for real complex agricultural fields. In addition to this, the wind field around a survey area is measured from a multirotor carrying an ultrasonic wind speed sensor. This shows that the assumption of steady uniform wind holds true for the small areas and time scales of an unmanned aerial vehicle aerial survey.

KEYWORDS

boustrophedon paths, coverage path planning, fixed-wing UAV, flight testing, polygon decomposition, recursive greedy, remote sensing, wind

1 | INTRODUCTION

Both commercial and hobbyist unmanned aerial vehicle (UAV) markets have grown significantly over the last few years. The worldwide market for agricultural drones was $494 million in 2016 and is anticipated to reach $3.69 billion by 2022, suggesting that agricultural drones will likely dominate all other drone sectors (WinterGreen Research Inc., 2016). They have reduced in price and have matured in both safety and reliability. UAV-based remote sensing has been shown to be effective across a huge range of applications with growing popularity, for example, forestry and agriculture (Grenzdörffer, Engel, & Teichert, 2008) and coastal and environmental remote sensing (Klemas, 2015). This is especially true in the field of precision agriculture (Anderson, 2014) due to their low cost, greater flexibility, and better spatial and temporal resolution compared with more traditional methods such as

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2019 The Authors. Journal of Field Robotics published by Wiley Periodicals, Inc.

manned aircraft or satellite remote sensing (Whitehead & Hugenholtz, 2014). Therefore, efforts to improve the safety, functionality, and efficiency of agricultural drones, and their algorithms, are essential.

Precision agriculture is a technique based on observing, measuring, and responding to inter- and intra-field variability in crops, with the aim of maximizing returns and/or minimizing costs. The types of data that are most often gathered for precision agriculture include weed & disease distribution, soil pH, crop yields, and various crop health indices. These data are traditionally gathered by hand, by agronomists, dividing a field into sections and sampling. This actionable agricultural data can be used to help increase yields, while lowering fertilizer, herbicide, and water use. However, farms can be vast (the average US farm is 175 ha; United States Department of Agriculture, 2016) so gathering any data by hand can be prohibitively time-consuming and expensive (Rew & Cousens, 2001). This necessitates a more effective way of agricultural surveying, such as using UAVs. An example would be site-specific weed management (SSWM) (Zhang & Kovacs, 2012); where weed maps are generated from remote sensing aerial images and herbicides are more efficiently applied to only the affected areas of the field. Other examples include disease mapping (Fornace, Drakeley, William, Espino, & Cox, 2014) and yield prediction (Bendig et al., 2014).

UAV remote sensing can be performed by either a rotary or fixed-wing UAV, equipped with an imaging system which can be fixed directly to the airframe, or mounted on a gimbal (Anderson, 2014). There is a wide range of imaging systems that can be used depending on the application, including visual, multispectral, or hyperspectral systems which sample over a wide range of electromagnetic frequencies. Rotary wing UAVs are ideal for low altitude, slow flights, providing high-resolution images but have very low endurance. Depending on the platform, payload, and batteries, flight times of around 15–20 min are typical. This means that only a small area can be imaged. Alternatively, fixed-wing UAVs tend to fly higher and faster, resulting in lower resolution imagery, but they have much higher endurance and can survey a much larger area in a single flight. This paper will deal only with fixed-wing remote sensing due to the focus of this paper on large aerial surveys.

The process of remote sensing from a small fixed-wing UAV begins by planning a path that enables large sets of images to be taken orthogonal to the ground to cover the entire region of interest (ROI), which is usually represented by a user-defined polygon (Colomina & Molina, 2014). For a simple convex field, the most common technique is a simple back and forth sweep motion known as a boustrophedon path. These images are then stitched together using structure-from-motion algorithms to generate a single high-resolution image, also known as an "orthophoto." A digital elevation model (DEM) and/or nonvisible spectral images of the ROI could also be produced using one of a number of commercially available photogrammetry software packages (Gini et al., 2013). These can then be used to extract actionable information on the surveyed field.

1.1 | Mission planning capability gap

Farms tend to be large areas, and due to historic land boundaries and local features such as roads, rivers, forests, and towns, are often complex irregular shapes (especially in Europe). A study of agricultural fields in Finland found that only 13% of farms were simple convex shapes (Oksanen & Visala, 2009). Until recently, large scale unmanned aerial surveys have not generally been performed because beyond visual line-of-sight flight for UAVs is legislatively difficult. However this is about to change; the Civil Aviation Authority (CAA) in the UK (Department for Transport (UK), 2016) and the Federal Aviation Administration (FAA) in the United States (Alta Devices, 2017) are pushing to make this more straightforward. This will enable farmers and operators, for the first time, to legally perform large-scale complex surveys over a number of fields in a single flight. The next limit to the size of the surveys that are possible is the limited endurance of UAVs. While UAV flights are limited by endurance, the coverage path planning (CPP) mission plan can make a huge difference to how long the flight can take, thus the total possible coverage area for given flight time. This can make the difference between performing the survey over single or multiple flights, which is to be avoided as it would add operational time, complexity, and cost.

Current commercial mission planning software packages are often simplistic and are not suited for large highly complex field shapes (Coombes, Chen, & Liu, 2018). They tend to take no account for the shape of the field, wind or any other operational factors. While there have been a number of CPP techniques developed in the literature to generate paths for a total coverage of an area, the vast majority have been primarily designed for ground robots (Latombe, 2012). Contrary to ground robots, aircraft are not constrained to operate within the bounds of the polygon and are subject to additional complexities such as no-fly zones (NFZs). Therefore, although similar, algorithms developed for ground robots are not directly applicable for UAVs.

There are three main categories of techniques developed in previous literature to perform sensor CPP tasks; cell decomposition (Choset, 2000), grid-based methods (Arney, 2007), and informative path planning (IPP) (Galceran & Carreras, 2013). IPP guides the vehicle around the ROI in real-time, aiming to maximize information gain while ensuring total coverage (Paull, Thibault, Nagaty, Seto, & Li, 2014). These methods are not suited to fixed-wing UAVs because they tend to have quite long erratic paths and require a gimbal to keep the imaging system orthogonal to the ground during the frequent turns (Acar, Choset, Rizzi, Atkar, & Hull, 2002). Alternatively, grid-based methods discretize the whole ROI polygon into a grid (Elfes, 1987) and plan a path the covers each unoccupied cell. This only approximates the region, also generating paths that would lead to a large number of turns, making grid-based methods unsuited for fixed-wing aircraft surveys for the same reasons as IPP. Grid-based methods also tend to require maneuvers that a nonholonomic vehicle, those with a constant forward motion, could not feasibly perform. Cell decomposition methods, on the other hand, decompose

the ROI into a number of smaller cells which are individually covered with a simple and efficient path. These methods lend themselves to aerial survey CPP, because the long, thin decomposed cells will require fewer turns and have the aircraft in straight level flight for longer periods of time where high quality stable images can be obtained (Li, Chen, Er, & Wang, 2011).

Much of the work in this area assumes a negligible effect from the wind. It is shown by Li et al. (2011) that minimizing the number of turns (NT) in the survey, that is, aligning the angle of a boustrophedon path with the long axis of each decomposed convex polygon, results in the most efficient survey path. This works by minimizing the time wasted in the turns between sweeps, where no images can be taken due to the aircraft roll angle. Huang (2001) tries to minimize flight time in a survey by using the similar heuristic cost function, the sum of long axis lengths (minimal sum of altitudes [MSA]) from each decomposed convex polygon. In addition to effectively trying to limit the number of turns, it also has the effect to limit the total number of decomposed cells which serves to decrease the extra wasted transit time between them. However, the wind strength and the flight path angle to the wind will have a massive impact on the flight time of a survey (Coombes, Chen, & Liu, 2017; Hovenburg, de Alcantara Andrade, Rodin, Johansen, & Storvold, 2018; Richards, 2018) due to high wind velocity relative to airspeed (typically 20%–50%). Flight times can be significantly improved by flying perpendicular to the wind direction (Coombes, Fletcher, Chen, & Liu, 2018). This means that in wind, finding the best path is a complex balancing act between flying perpendicular to the wind, minimizing the number turns and minimizing transit times between decomposed cells. Furthermore, CPP can be complicated further by being close to areas in which overflight is not allowed (e.g., built-up areas) or not advised for flight safety (e.g., power lines) known as NFZs.

In literature, research into complex large scale surveys by a fixed-wing UAV is currently relatively sparse. For complex-shaped fields; work is limited to extensions of ground robot decomposition techniques (Huang, 2001; Li et al., 2011; Xu, Viriyasuthee, & Rekleitis, 2011), and does not account for the additional complexities previously mentioned. In contrast, UAV path planning in the wind and the associated modeling is well understood (Ceccarelli, Enright, Frazzoli, Rasmussen, & Schumacher, 2007; Jennings, Ordonez, & Ceccarelli, 2008; Schopferer & Pfeifer, 2015), however, these techniques are not suited to CPP for aerial surveys. Only a single paper outside of our own work has been found that combines these two areas. Richards (2018) plots a full boustrophedon path in the same orientation across the whole concave field polygon, then calculates a time cost in steady uniform wind for each sweep and uses a modified traveling salesman solver to find the fastest order and direction to traverse all the sweeps without forcing the aircraft to perform maneuvers that it is incapable of. It is mentioned that this technique and others are dependent on an accurate flight time model in the real world. Therefore, comprehensive flight testing should be performed to verify such flight time models. While there are a few papers that do perform flight testing on fixed-wing UAV CPP (Avellar, Pereira, Pimenta, & Iscold, 2015; Paull et al., 2014; Xu et al., 2011) they did not attempt to predict flight time, rather just showing that complete coverage can be achieved. In addition, this paper assume a steady uniform wind, which is a standard assumption for many large scale planning problems. However, there are some examples for use with UAV gliders that use dynamic-wind maps to maximize soaring times, where actual flight tests were conducted successfully (Depenbusch, Bird, & Langelaan, 2018a, 2018b).

1.2 | Contributions

This paper aims to develop a CPP algorithm for complex and irregular field shapes that is fast enough to run when on-site conducting an aerial survey. The algorithm should take into account real-world considerations like wind, NFZ, and launch points and output a set of way-points that can be uploaded to most commercial fixed-wing UAV platforms. This enables operators to conduct a custom survey for user-defined aircraft, imaging systems, and survey requirements. The specific novel contributions from this paper are given as follows:

- Time-efficient fixed-wing UAV CPP for large irregular-shaped fields.
- Low computational time polygon decomposition algorithm.
- Real-world considerations: NFZ, launch points, wind.
- Consideration of the convex hull due to the fact that fixed-wing UAVs can fly outside of ROI.
- Experimental aerial surveys conducted, where the survey mission is planned using the proposed algorithm.

An example output is shown in Figure 1 where the UAV path is defined for two decomposed cells. Results will be provided in the form of simulations and extensive flight testing to compare its effectiveness to previous methods, and to show the validity of the flight time in wind model and the steady uniform wind assumption.

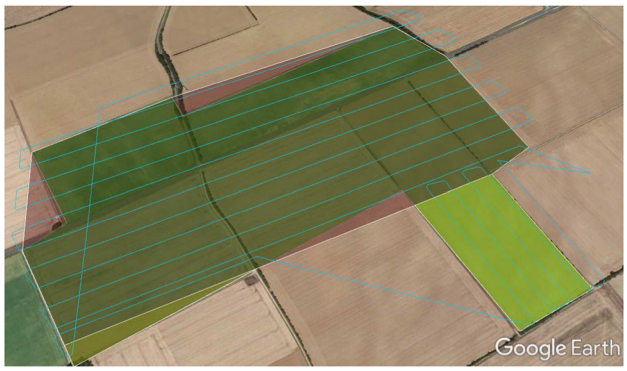


FIGURE 1 Example concave region of interest decomposed into two separate polygons, which are sequentially surveyed using boustrophedon paths [Color figure can be viewed at wileyonlinelibrary.com]

Note that this study is in parallel to the work in Coombes, Fletcher, et al. (2018). Whereas the focus of the previous paper was to produce a fully optimal path given large computational resources, this paper develops a suboptimal algorithm that trades a small loss of flight time efficiency for a large decrease in calculation time, enabling real-world use in the field.

1.3 | Paper structure

The structure of the remainder of the paper is as follows. Section 2 provides a succinct summary of the authors' previous work with CPP for fixed-wing UAVs, alongside the details of new work which has been used to develop the cost function in this paper. It also includes experimental evidence to show that the assumption of the uniform steady wind is valid over the area and time-scale of a typical survey. Section 3 explains the proposed recursive-greedy decomposition algorithm in detail. Section 4 presents and compares the paths generated from the proposed method to that of previous methods in simulation. A number of examples real-world fields are used to demonstrate the algorithms functionality, and a Monte-Carlo simulation using 30 randomly generated polygons is used to make a numerical comparison of the methods. Section 5 details flight testing performed, to assess time prediction accuracy and further comparison to previous techniques are made. Section 7 and 7.1 provide concluding remarks and suggest extensions to this study for the future. Appendix A shows image coverage results and orthophoto outputs of one of the flight tests, demonstrating that total image overage has been achieved in the real-world.

2 | COST FUNCTION—FLIGHT TIME IN WIND (FTIW)

The most efficient path to ensure complete image coverage of a single convex polygon by a fixed-wing UAV is a boustrophedon path (Li et al., 2011), which consists of back and forth sweeps across the polygonal ROI. To minimize the number of turns, the survey angle (ψ_s) is fixed based on the angle of the long axis of the polygon. The distance between sweeps (d_x) is set as shown in Equation (1). It is based on the horizontal angular field of view (FOV) of the sensor (α_x), the number of pixels the sensor has in the horizontal direction (n_x), the required ground sample distance (GSD) (D_S) and the lateral overlap ratio (or sidelap, w_s) is defined by the user based on the application. A higher w_s of around 0.6–0.8 is needed for DEM or a lower value of around 0.2–0.4 for orthophotos.

$$d_x = 2\tan\left(\frac{\alpha_x}{2}\right)\frac{(1-w_s)n_x D_S}{\alpha_x}. \tag{1}$$

The GSD represents the physical size of a pixel in an image and is measured in cm/pixel, different applications have different GSD requirements. So the height above ground of the survey (h) can be set based camera parameters and the desired GSD as shown in Equation (2).

$$h = \frac{D_S n_x}{\tan\left(\frac{\alpha_x}{2}\right)}. \tag{2}$$

Although it may be possible to overfly the complete convex hull of the ROI, this may not be efficient, or desirable. In this case, the ROI can be decomposed into a number of convex polygonal cells that can be surveyed individually and, when combined, provide total coverage of the ROI. Figure 2 shows an example of complex polygonal decomposition into three convex polygonal cells, with a boustrophedon path displayed for one of these.

This section describes the method used to calculate the flight time in wind cost (J_{FTIW}) for such a set of decomposed cells, which is to be used for the recursive optimization process in Section 3. The cost function is made up of two parts; the flight time (t) and an additional cost (J_Ω) representing a hard constraint to prevent overflying of NFZs. The total flight time for a set of cells made up of the time to fly; (a) the launch (ascent) path (t_a), (b) the sum of the times to fly each individual cell using boustrophedon paths ($\sum t_c^i$), (c) the total time required to traverse between the decomposed cells ($\sum t_x^j$), and finally (d) the landing (descent) path (t_d). See Equation (3), where n_c is the number of decomposed cells.

$$J_{\text{FTIW}}(P) = t + J_\Omega = \left(t_a + \sum_{i=1}^{n_c} t_c^i + \sum_{j=1}^{n_c-1} t_x^j + t_d\right) + J_\Omega. \tag{3}$$

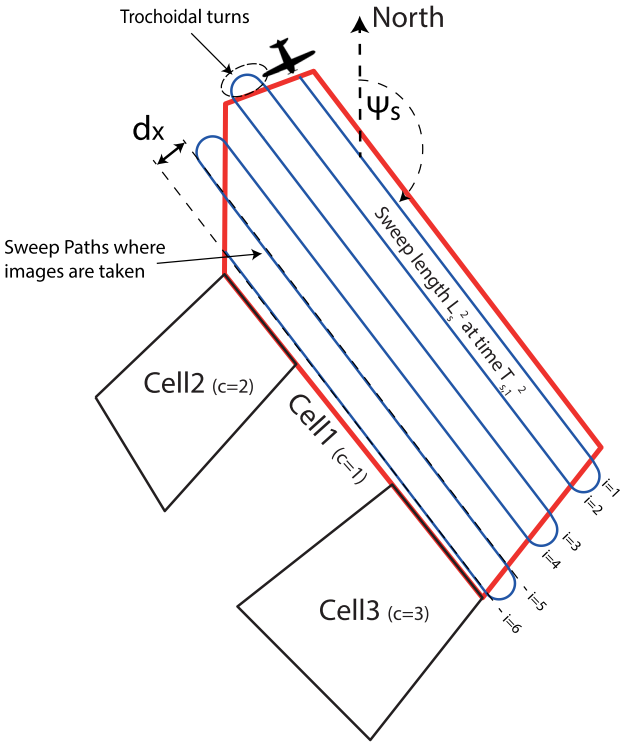


FIGURE 2 Example irregular field decomposition, with boustrophedon path planned along the long axis of cell 1 [Color figure can be viewed at wileyonlinelibrary.com]

2.1 | Single-cell FTIW

The complete single-cell path consists of two different states of flight; the straight sweep paths where the images are captured, and the trochoidal turn maneuvers used to transition between sweeps. The flight time for a single cell can be calculated as the sum of the time spent during each of these states, see Equation (4).

$$t_c = \sum_{i=1}^{n_s} t_{s,c}^i + \sum_{j=1}^{n_s-1} t_{t,c}^j. \tag{4}$$

2.1.1 | Sweep paths

The time to fly along a single sweep line (t_s^i) is calculated as the length of the sweep (L_s^i) divided by the aircraft's ground speed (v_g). If a steady uniform wind field across the survey is assumed, the aircraft's ground-speed can be calculated by wind triangle vector subtraction; this is standard book work but is illustrated in Richards (2018). Ground-speed is calculated for each sweep in Equation (5), where v_a represents the vehicle's airspeed, v_w represents the wind speed and the direction of the flight path (ψ_a) is either equal to the sweep direction for outward legs, or opposite to it for return legs, see Equation (6).

$$t_{s,c}^i = \frac{L_s^i}{v_a \cos\left(\arcsin\left(\frac{v_w}{v_a}\sin(\psi_a)\right)\right) + v_w \cos(\psi_a)}, \tag{5}$$

$$\psi_a = \begin{cases} \psi_s & \text{if } i \text{ odd}, \\ \psi_s + \pi & \text{if } i \text{ even}. \end{cases} \tag{6}$$

2.1.2 | Turns

The turns joining two sweeps are represented by Dubins paths modified to account for wind. This algorithm is laid out in detail in Coombes et al. (2017), and as such, will only have a brief overview here. A circular coordinated turn in a steady uniform moving body of air causes the path flown by aircraft relative to the ground to be trochoidal. Each turn is made up of three continuous connected paths; two trochoidal turns paths linked by a straight path tangential to both trochoids. The time taken for this full turn maneuver (t_t^j) consists of the time taken to fly each trochoidal turn (t_1^j, t_2^j), and the straight tangential path between them (t_3^j), shown in Figure 3 and Equation (7). For further detail about the derivations of these terms, please refer to Coombes et al. (2017) and Techy and Woolsey (2009).

$$t_{t,c}^j = t_1^j + t_2^j + t_3^j. \tag{7}$$

2.2 | Steady uniform wind assumption

The flight-time calculations for both the sweep and turn portions of flight assume a steady uniform wind over the relatively short time and area of the survey. If this were to be incorrect, this could significantly affect the time prediction accuracy. To show that the assumption does hold, the wind field (wind at multiple locations) was measured at a typical survey altitude of 100 m by a multirotor UAV. The measured wind field was assessed for spacial and temporal variations. Note that this step would not be necessary in a real-world survey, it has been included here to test the assumption of wind prediction.

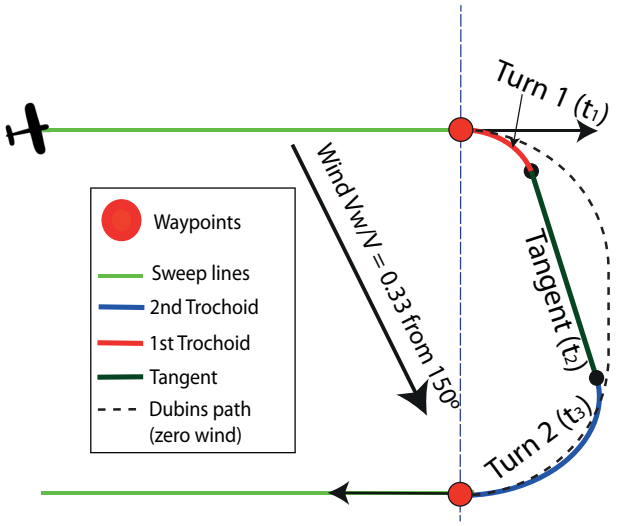


FIGURE 3 Trochoidal turn path formulation for a maneuver to transfer between two sweep paths. Using modified Dubins paths to account for the wind [Color figure can be viewed at wileyonlinelibrary.com]

To achieve this, a sweep pattern was flown over the survey area, stopping and hovering at 16 separate way-points and sampling wind for 15 s at each. This was repeated on four different days. The aircraft used was a DJI S1000 octorotor, with an FT205EV ultrasonic wind sensor mounted above the vehicle out of the rotor wash, as shown in Figure 4. This allows direct measurement of the wind at any altitude. It has a 15 min flight time using a 520 Wh battery, and a gross weight of 5 kg. Using a Raspberry Pi 2 running robot operating system (ROS), it interfaces with the autopilot and wind sensor to log wind speed and direction, position, heading, and altitude data.

The wind field is measured at the same survey site where the accuracy assessment flights in Section 4 are conducted, however, on four different days. The average measurements across the first wind field measurement flights were 7.3 m/s and 230.2°. Shown as (#1) in Table 1. This closely matched the wind forecast information (225° at 7 m/s) which was taken from a meteorological station located 5 miles north of the test site. The wind strength and direction remained quite consistent across the whole survey area and across the 14 min flight time, with standard deviations of 0.627 m/s and 10.72°. The wind field with the S1000 flight path is shown on Figure 5.

The other three (#2–#4) wind survey flights gave similarly consistent results as shown in Table 1. Weaker wind strength leads to slightly more variation in wind direction, however as the wind is weaker, this will have less effect on the aircraft. Using the same



FIGURE 4 S1000 multirotor unmanned aerial vehicle with FT205EV ultrasonic wind speed sensor mounted out of rotor wash used to perform the wind field measurements [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Wind field measurements result on four different days

#	Wind forecast (m/s)	Average Wd	Average Vw (m/s)	Std Wd	Std Vw (m/s)
1	225°/7	230.2°	7.3	10.72°	0.627
2	240°/7.5	244.12°	8.3	9.62°	1.13
3	270°/3	289.3°	3.74	14.5°	0.518
4	270°/6.5	259.4°	7.08	10.1°	1.2

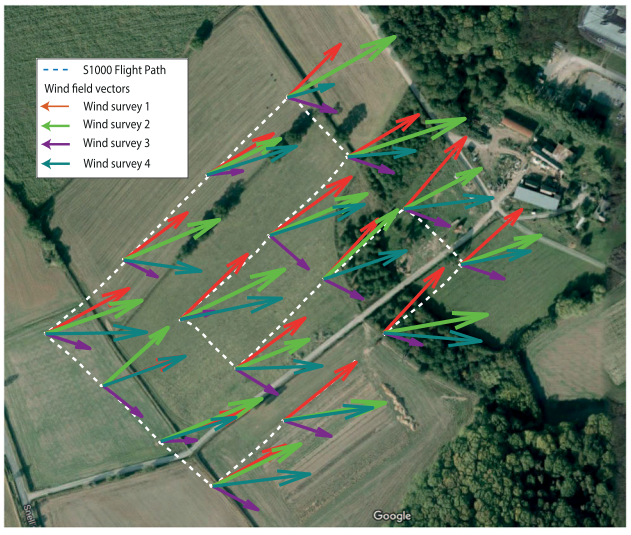


FIGURE 5 Visualization of the average wind vectors measured at each sampling point of the wind field, during four separate wind surveys [Color figure can be viewed at wileyonlinelibrary.com]

simulation techniques described in Section 4, it has been calculated that the variation in wind observed will result in less than 3% variation in the flight time for a typical mission.

2.3 | Traversal paths

Previous work by the authors (Coombes, Fletcher, et al., 2018) assumed that the sum of the flight times for the individual cells could be used as an estimate of the overall flight time. This was done to meet the dynamic programming requirement for optimal substructure. However, it has since been found that in some circumstances this algorithm tends to slightly over-decompose the ROI due to the lack of a penalty for doing so. As a result, one of the developments for this study has been to include the transfer paths from the launch site, between each decomposed cell, and back to the landing site into the cost function. These are generated as modified Dubins paths in the same way as the turn portion of the boustrophedon paths, as described in Section 2.1.2.

The traversal path is complicated by the fact that each cell (c^k) has four corners that the aircraft can start its survey from c_n^k, where $n \in [1, 2, 3, 4]$ (Note that the entry corner will also define the exit corner depending on whether there is an even or odd number of sweeps). The algorithm must, therefore, decide on how best to traverse a given set of decomposed cells. Shown in Figure 6 is an example ROI that has been decomposed into three convex polygonal cells. In this example, there are $3! \times 4^3$ different possible routes to take between the start and endpoint. To find the fastest one, a graph search is performed using Dijkstra's algorithm. Also shown in Figure 6

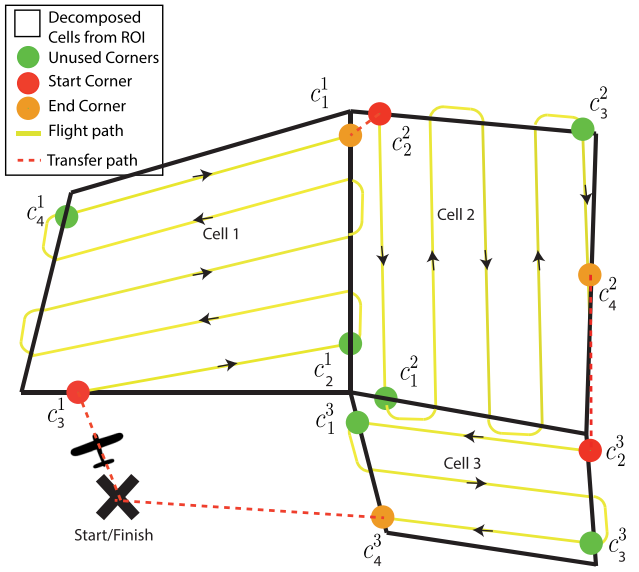


FIGURE 6 An example concave polygon decomposition with boustrophedon paths on each cell. Each boustrophedon path has four possible start and endpoints, this figure shows the fastest route to traverse each cell to and from the aircraft start location. Start → $c_3^1 \rightarrow c_1^1 \rightarrow c_2^2 \rightarrow c_4^2 \rightarrow c_2^3 \rightarrow c_4^3 \rightarrow$ end (Coombes, Fletcher, et al., 2018). ROI, region of interest [Color figure can be viewed at wileyonlinelibrary.com]

is the fastest path which has ensured that the transversal path is minimized.

2.4 | No-fly zones

Finally, while the overall aim is to minimize the flight time, there may be other operational factors that need to be considered. To demonstrate a method for their inclusion, NFZs have been included. These are areas that the survey is prohibited to overfly or perform turns over. This may be due to structures or personnel not under control by the operator: In this case, the aircraft is not to come within 50 m of these areas (Civil Aviation Authority, 2012).

If it is assumed that there are no NFZs within the operator-defined ROI, then there are three instances where the survey could violate this rule: (a) overflying an NFZ during cell traversal, (b) overflying NFZ while turning between sweeps, and (c) overflying an NFZ while corner cutting. Corner cutting is a technique used in the decomposition phase to find simpler faster decompositions, this will be detailed in Section 3. If any of these cases occur then a hard constraint is imposed by adding an infinite cost to the cost function.

To calculate if a NFZ will be violated, an area called the extended flight zone (EFZ) is created. The EFZ is a polygon that represents the total area that the aircraft will overfly for each decomposed cell, inclusive of the modified Dubins paths for turning and traversal. Each EFZ is represented by polygon $P_{E,k}$ with the same index k as its parent $P_{C,k}$. Each edge, E_k^j, of each $P_{E,k}$ is shifted outside $P_{C,k}$ by the turn radius of the aircraft where $R_t = \frac{V}{\psi}$ at the survey angle ψ_s. To approximate the area that the aircraft will turn in, we assume zero wind for the turn radius.

Shown in Figure 7 is an example of EFZ of a single convex survey cell from an already decomposed ROI. Here the NFZ is shown to be power pylons that come very close to the boundary of the ROI. Shown in the exploded view is how each edge is shifted outward from the original polygon. This creates an additional green area which is where the turns are performed. The areas in yellow are portions of the flight that the aircraft will overfly areas outside of the ROI due to corner cutting. In the case of this example, the pylon NFZ intersects with the EFZ so the cost J_Ω will be infinite and another solution will need to be used, as shown in Equation (7).

$$J_\Omega = \begin{cases} \infty, & \text{if area } (P_E \cap P_N) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

2.5 | Benchmark algorithms

Two alternative cost functions have been chosen from the literature for comparison. These are the NT (Equation (9); Li et al., 2011) and the MSA (Equation (10); Huang, 2001) algorithms previously mentioned in Section 1.1;

$$J_{\mathrm{NT}}(P) = n_s - 1, \tag{9}$$

$$J_{\mathrm{MSA}}(P) = \sum_{i=1}^{n_s} L_s^i, \tag{10}$$

where n_s is the total number of sweeps.

3 | RECURSIVE GREEDY DECOMPOSITION

To use boustrophedon paths to survey an ROI described by an irregular concave polygon, it must first be converted into one or a number of convex polygons. Depending on the shape of the ROI, it may be more efficient to either split the area into multiple convex cells and survey these separately, or to simply calculate the convex hull of the concave polygon. This section describes the optimization algorithm used to make this decision so that one of the cost functions (Equation (3), (9), or (10)) laid out in the previous section can be minimized.

The proposed method uses trapezoidal decomposition to break a concave irregular polygon into two (or more) subregions, known as "cells." At the same time, the convex hull of the original polygon is calculated and the optimizer will select the option which produces the smallest cost. Because each of the two cells may or may not be convex, the algorithm works recursively until every cell included in the cost calculation is convex. Figure 8 demonstrates how the decomposition is performed. The algorithm is "greedy" in that if, at any time, the calculated flight time of the convex hull of any cell is found to be more efficient than the decomposition, then this option is

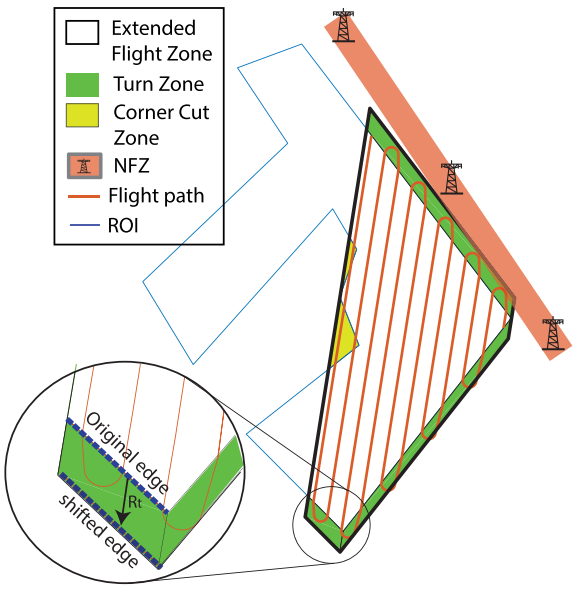


FIGURE 7 Example boustrophedon path over a decomposed cell, showing the formulation of EFZ from the turn zone and corner-cutting zone. As a part of the EFZ intersects with the NFZs this is an unacceptable decomposition. EFZ, extended flight zone; NFZ, no-fly zone; ROI, region of interest [Color figure can be viewed at wileyonlinelibrary.com]

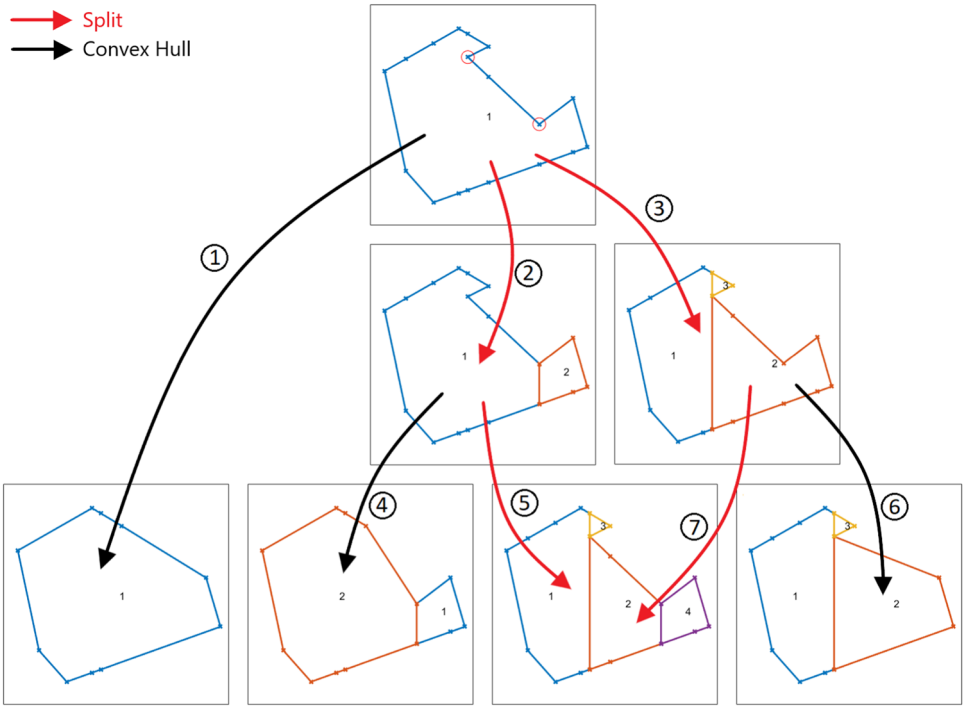


FIGURE 8 Example recursive-greedy decomposition search graph for a polygon with two concave vertices. The decomposition algorithm splits the cell using a vertical line that intersects with each concave vertex in turn. Note that steps ③ and ⑤ result in three decomposed cells (rather than two) due to the angle and shape of the original polygon [Color figure can be viewed at wileyonlinelibrary.com]

selected. Although this does not guarantee an optimal solution, it ensures that a solution is found quickly, even for highly irregular polygons with a large number of concave vertices. This is an important operational consideration because this algorithm is designed to be used in the field using portable computers with limited computational power.

3.1 | Recursive optimization algorithm

The algorithm works using a "top-down" architecture; calculating the cost of the convex hull and iteratively splitting concave regions, one concave vertex at a time until no concave cells remain. The pseudo-code for this function can be found in Algorithm 1. The steps of the algorithm can be summarized as follows;

1. First, the convex hull (P_h) of the ROI polygon (P_ψ) is determined and the relevant cost function (Equation (3), (9), or (10)) is used to calculate the associated cost (line 3). This cost is used to initialize the current best cost value ($J^{\lambda'}$, line 5) and the optimal set of decomposed polygons ($\mathbf{P}^{\lambda'}$, line 4).
2. Vertices that cause the polygon to be concave (aka "concave vertices," $\mathbf{V_c}$) are identified by their internal angle being greater than π (lines 9–14).
3. Trapezoidal decomposition is used to split the original polygon at the first of the remaining concave vertices (V_c^j). This will result in two or more polygons (aka "subregions," $\mathbf{P_s}$) being generated (line 18).
4. The cost of each subregion (P_s^k) is calculated directly using the relevant cost function if the subregion is convex (line 23). If the subregion is concave, the function recurses using the concave subregion as its input, which will output the optimal cost of that concave subregion (line 26).
5. The sum of the costs of the subregions (J_v) identified in Step 3 is calculated (lines 24 and 28). If this is less than the current best cost value ($J^{\lambda'}$), then the current best cost ($J^{\lambda'}$, line 32) and optimal decomposition set ($\mathbf{P}^{\lambda'}$, line 33) are updated.
6. Steps 3–5 are repeated until each of the concave vertices have been examined (line 21). The function outputs the final value of the current best cost ($J^{\lambda'}$) and its associated set of decomposed convex polygons ($\mathbf{P}^{\lambda'}$, line 37).

The order of the function can be approximated using Equation (11), where $n_{v,c}$ represents the number of concave vertices. This represents a significant improvement over the dynamic programming (DP) solution from our previous work (Coombes, Fletcher, et al., 2018) which was approximated as $O(c^{n_c}/2), (c > 1)$, where n_c represented the number of decomposed cells, a much larger number for a given ROI.

$$O(n_{v,c}) = \Sigma_{k=0..n_{v,c}}\left(\frac{n_{v,c}!}{k!}\right). \tag{11}$$

Algorithm 1 Recursive Splitting Algorithm

```
1: function recSplitFcn(P_ψ)
2:
3:     % Calculate Cost of Convex Hull
4:     [P_h] = convexHull(P_ψ)
5:     P^λ' = P_h
6:     J^λ' = J(P^λ')
7:
8:     % List Concave Vertices (V_c)
9:     V_c = ∅
10:    for V^i in P_ψ do
11:        if ∠(V^i) > π then
12:            V_c = V_c ∪ {V^i}
13:        end if
14:    end for
15:
16:    % Split ROI at each Concave Vertex
17:    for V_c^j in V_c do
18:        P_s = splitFcn(P_ψ, V_c^j)
19:        P_v = ∅
20:        J_v = 0
21:        for P_s^k in P_s do
22:            if isConvex(P_s^k) then
23:                P_v = P_v ∪ {P_s^k}
24:                J_v = J_v + J(P_s^k)
25:            else
26:                [P^λ'', J^λ''] = recSplitFcn(P_s^k)
27:                P_v = P_v ∪ P^λ''
28:                J_v = J_v + J^λ''
29:            end if
30:        end for
31:        if J_v < J^λ' then
32:            J^λ' = J_v
33:            P^λ' = P_v
34:        end if
35:    end for
36:
37: return (P^λ', J^λ')
38: end function
```

3.2 | Polygon rotation

As previously mentioned, the algorithm attempts polygon splits at each concave vertex. It achieves this by using a simplified version of the trapezoidal decomposition algorithm found in Seidel (1991). As explained in Coombes, Fletcher, et al. (2018), the long thin trapezoids often produced, lend themselves very well to efficient fixed-wing surveys. The simplification of the algorithm used here is, that instead of splitting the polygon at every vertex to make $n_v - 2$ (where n_v is number of vertices) trapezoids, it only makes the split at a single concave vertex at each step (line 18 of Algorithm 1). As trapezoidal decomposition splits along vertical sweep lines, this often leads to decomposed cells that have similar ψ_s angles. If the vertical sweep line is aligned with the wind, this could lead to longer flight times, as the UAV will be flying directly into and away from the wind. However, if the ROI is rotated, the dominant ψ_s across multiple cells can be changed. Due to the efficiency of the recursive greedy search (RGS) algorithm, this is used to enlarge the search space by running the recursive function for a range of initial polygon rotations (ψ_r), the number of which is dictated by a user-selected positive integer number n_r. This can be run using parallel computing techniques, even on a portable computer.

For each polygon rotation angle (ψ_r), there will be an optimal set of polygons ($\mathbf{P}^{\lambda'}$) which will have an associated cost ($J^{\lambda'}$) selected by the recursive greedy function. The rotation angle which produces the lowest overall cost (J^{λ}), is output as a set of polygons ($\mathbf{P}_{\psi}^{\lambda}$) after

re-rotation back to the original angle. Note that only angles between $0 \leq \psi_r < \pi$ are required because $\pi \leq \psi_r < 2\pi$ will produce identical results. Also note that the wind direction must also be rotated for use in the cost function to keep it in the same relative angle.

Algorithm 2 explains the polygon rotation angle optimization using a brute force search. A set of angles (ψ_r) is defined (line 1) and the optimal cost (J^λ) is initialized as infinity (line 2). Using a loop (line 3), the initial polygon (P_i) is rotated by the angle to be checked (ψ_r) (line 4) and the decomposition algorithm is called (line 5) to calculate the optimal cost ($J^{\lambda'}$) and set of decomposed convex cells ($\mathbf{P}^{\lambda'}$) for the given angle of rotation. The decomposed cells are rotated back to the original frame of reference (line 6). Finally, if the optimal cost at the current angle is lower than the current overall optimal cost (J^λ), then J^λ and the corresponding set of optimized cells ($\mathbf{P}^\lambda_\psi$) are updated.

Algorithm 2 Polygon Rotation

1: $\mathbf{R} = \{\psi_r, 0 \leq \psi_r < \pi, n(\mathbf{R}) = n_r\}$
2: $J_\lambda = \infty$
3: **for** ψ_r in $\mathbf{R}$ **do**
4: $\quad P_\psi = \text{cellRotation}(P_i, \psi_r)$
5: $\quad [\mathbf{P}^{\lambda'}_\psi, J^{\lambda'}] = \text{recSplitFcn}(P_\psi)$
6: $\quad \mathbf{P}^{\lambda'} = \text{cellRotation}(\mathbf{P}^{\lambda'}_\psi, -\psi_r)$
7: $\quad$ **if** $J^{\lambda'} < J^\lambda$ **then**
8: $\quad\quad \mathbf{P}^\lambda = \mathbf{P}^{\lambda'}$
9: $\quad\quad J^\lambda = J^{\lambda'}$
10: $\quad$ **end if**
11: **end for**
12: **return** $(\mathbf{P}^\lambda, J^\lambda)$

4 | SIMULATION RESULTS AND DISCUSSION

This section presents simulated mission plans for a number of complex real-world example fields and randomly generated field-style polygon shapes to demonstrate the improvements that the proposed algorithm offers over previous cost functions. Three complex, concave, agricultural fields will be used as examples, which are each shown in Figure 9. The launch/landing locations are from feasible accessible locations and are also marked in Figure 9. All of these field polygons have a large recess which is a notoriously hard feature to plan an efficient mission around. This is because, when decomposed, they tend to generate a high number of convex polygons resulting in a large search space for path optimization (Coombes, Fletcher, et al., 2018). The example fields have a relatively high complexity (11–16 vertices) and a wide range of areas (35–129 ha) for this type of problem.

The aircraft simulated is a senseFly eBee; a commercially available and widely used survey UAV. The eBee is a relatively low-velocity fixed-wing aircraft. The imaging system is a MicaSense RedEdge multispectral camera. Only an orthophoto is required, therefore, the requirements for image overlap/sidelap have been set to 30%. The RedEdge has a narrow FOV of 47.2°, so to get the required GSD of 4.3 cm/pixel, the survey is conducted at an altitude of 70 m. From Equation (1), this makes the distance between the sweeps d_x to be 46.2 m. To show how the proposed method handles the presence of wind, a steady uniform southerly wind ($\psi_w = 0°$) of 5 m/s is also applied to the simulation. The aircraft and survey parameters are summarized below in Table 2.

The polygon decomposition and mission planner algorithms laid out in Section 3 are applied to each of the example polygons. To compare methods, three missions per polygon will be planned, one for each of the cost function options; FTIW (Equation (3)—proposed algorithm), MSA, and NT (Equation (9) and (10), both previously discussed in Sections 1 and 2.5). As previously discussed, the FTIW cost function has two main advantages over the cost functions found in the literature. First, the FTIW cost takes into account the effect of wind on the flight time. This is expected to cause the suggested sweep direction to be aligned more perpendicular to the wind, which has already been found to be beneficial (Coombes, Fletcher, et al., 2018) for real-word survey efficiency. Second, the proposed algorithm may choose to survey the convex hull of a concave polygon rather than decomposing further ("corner cutting"). This is expected to result in fewer decomposed cells and hence a reduction in the transition time between them and will be selected by the optimizer only when it is beneficial overall.

The flight-times for each field, and for each cost function are shown in Table 3. It can be seen that the FTIW algorithm outperforms both cost functions from the literature by between 5% and 25% in all three examples. This is to be expected because these examples have been selected to showcase the benefits of the proposed algorithm. A fairer numerical comparison is presented later in the paper using Monte-Carlo simulation of randomly generated polygons.



FIGURE 9 Example complex concave field survey polygons used to explore the effectiveness of the proposed algorithm in simulation (a) 11 vertex arable field polygon 34.89 ha, with adjacent NFZ in red; (b) 15 vertex arable field polygon 47.86 ha; (c) 16 vertex arable field polygon 129 ha. NFZ, no-fly zone [Color figure can be viewed at wileyonlinelibrary.com]

Examining the results in more detail, Figure 10 shows the optimized mission plan for each cost function for the 11-vertex field from Figure 9a. It can be seen that the MSA and NT paths use six and eight decomposed cells, respectively, but the FTIW output uses just four. This is because the FTIW algorithm is able to take advantage of the ability to cut corners rather than relying on decomposition. As a result, the aircraft will spend less time transitioning between cells where no surveying is taking place and is the primary reason for the 4.8% and 7.18% improvements over the NT and MSA cost functions, respectively. While the proposed algorithm will take advantage of prior knowledge of the wind, in this case, the proposed method cannot find an efficient low-cell-count decomposition perpendicular to the wind. This is simply due to most of the polygons' edges being aligned close to 45° to the wind direction.

TABLE 2 Simulated senseFly eBee and survey parameters

Parameter	Value
Wind	0°/3 m/s
V	7 m/s
$\dot{\psi}$	0.7 rad/s
w_s, w_o	0.3, 0.3
h	70 m
GSD	4.3 cm/pixel
D_x	46.2 m

Abbreviation: GSD, ground sample distance.

TABLE 3 Calculated time comparisons between FTIW, NT, and MSA cost functions for three example fields

		Simulated flight times			FTIW % improvement vs.	
#	Polygon	FTIW (s)	NT (s)	MSA (s)	NT (%)	MSA (%)
1	11 Vertices	1,552	1,631	1,672	4.84	7.18
1	11 Vertices with NFZ	1,603	–	–	–	–
2	15 Vertices	1,121	1,182	1,436	5.16	21.9
2	15 Vertices 90° Wind	1,077	1,234	1,424	12.7	24.37
3	16 Vertices	2,953	3,277	3,342	9.89	11.64

Abbreviations: FTIW, flight time in wind; MSA, minimal sum of altitude; NT, number of turns.

The second example from Figure 9b is more complex with 15 vertices, five of which are concave. For this field, mission plans have been generated for both southerly and westerly winds ($\psi_w = 0, 90$). Their solutions are shown in Figure 11. For the southerly wind, the FTIW solution shows lower flight times than both NT and MSA algorithms, although the advantage over the MSA (21.9%) is much more pronounced than that over the NT algorithm (5.2%). This is because many of the polygon long axes are already perpendicular to the wind for the NT solution. The MSA algorithm, on the other hand, not only generates more cells but also flies largely parallel to the wind direction.

For comparison, Figure 11 also shows the flight paths for the three algorithms for a westerly wind, in which case the advantage of accounting for the wind becomes very obvious. It can be seen that the flight paths for the NT and MSA cost functions are identical because the do not take into account the wind in the planning, but the flight times have changed. Note that the flight time for the NT algorithm has increased due to the fact that the dominant sweep direction has gone from perpendicular to parallel to the wind direction. In contrast, the flight time for the MSA algorithm has reduced for the opposite reason. The FTIW algorithm, however, has accounted for the wind and changed the dominant sweep direction, and in this case, the convex hull of the entire ROI is found to be faster than decomposition resulting in a substantial 24.4% improvement over MSA and a 12% improvement over NT.

The final example, shown in Figure 9c, has similar complexity to the previous example (16 vertices and five concaves) but is a much larger field with an area of 129 ha. Shown in Figure 12 are the decomposition solutions and flight paths for each cost function. Using the FTIW algorithm gives the fastest flight time at 2,953 s, approximately 10% faster than the other algorithms. Contrary to the other two examples, the advantage of the FTIW algorithm comes

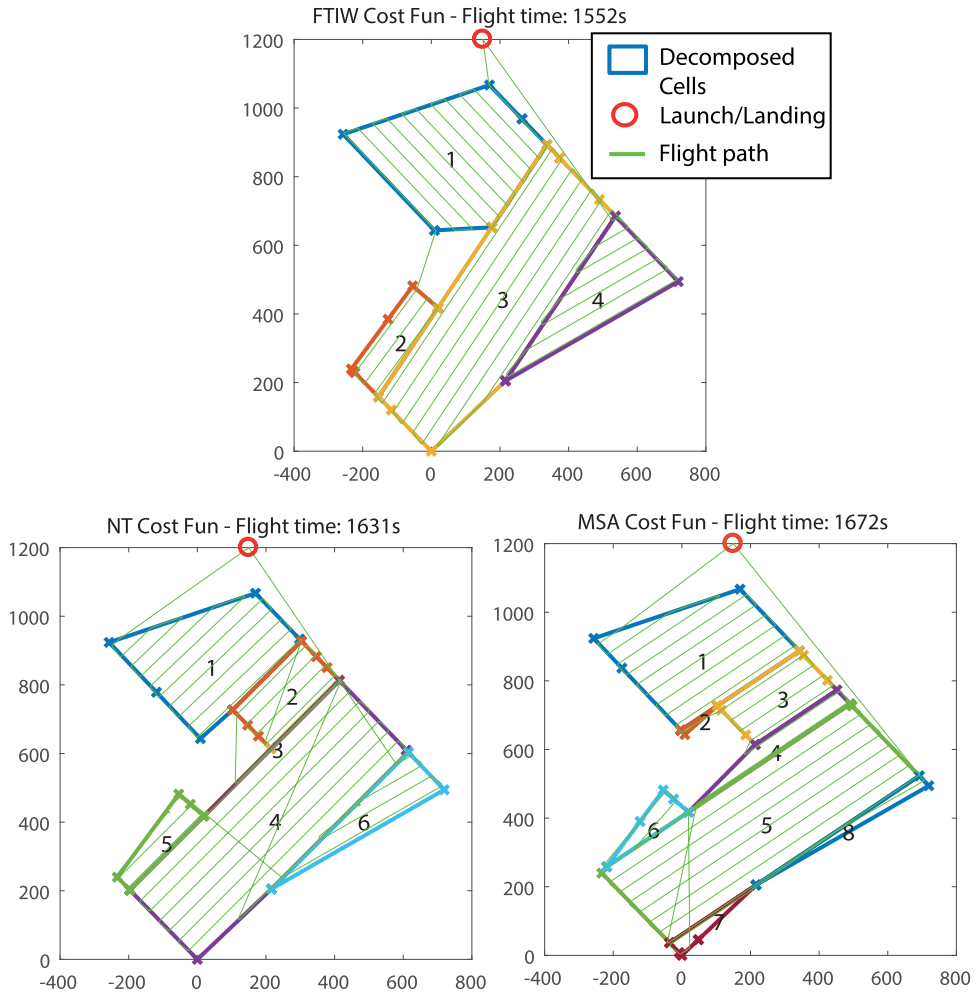


FIGURE 10 Field polygon from Figure 9a decomposed and flight paths generated using FTIW, MSA, and NT methods. Predicted flight times are shown. FTIW, flight time in wind; MSA, minimal sum of altitude; NT, number of turns [Color figure can be viewed at wileyonlinelibrary.com]

primarily from flying perpendicular to the wind. In this case, cutting corners is less beneficial because it requires unnecessarily surveying much more land area. The magnitudes of the times are also significant in this case because the maximum flight time of the senseFly eBee, as configured, is approximately 50 min (3,000 s). Therefore, the 10% flight time reduction from around 3,300 s to just below 3,000 s could make the difference as to whether the survey could be performed in a single flight.

4.1 | NFZ example

Section 2.4 showed how violating NFZs can be avoided by adding a hard constraint to any decomposition which results in overflying an NFZ. The 11-vertex field from Figure 9 is used to illustrate this, where the red area in Figure 9a is the NFZ containing uncontrolled buildings. Previously, the NFZ was not taken into account, and the NFZ was violated by a number of turns at the north-east end of cell 3. Figure 13 shows a decomposition where the NFZ is no longer violated. This was achieved by flying the entire north-east edge at an angle (ψ_r) of 135°. As seen in Table 3, this solution is around 50 s slower than the previous solution but is still faster than both algorithms from the literature that do not account for the NFZ.

4.2 | Computational complexity

One of the major advantages of this suboptimal method is that it runs much faster than the optimal method from Coombes, Fletcher, et al. (2018), which uses a full initial trapezoidal decomposition in its "bottom-up" dynamic programming (DP) approach. As the number of initial trapezoidal decomposed cells rises, the computational complexity for DP grows exponentially; making it practically infeasible to find a mission plan for a field polygon with more than 18 initial cells. However, as the proposed method searches the solution space very differently to the DP method, it is hard to compare their computational time complexity in a rigorous way. As DP tries all combinations of each initial decomposed cell and attempts to merge them, the computation time increases with the initial number of cells, which is based on a number of polygon vertices $N - 2$. However, the proposed method recursively slices cells apart at concave vertices, therefore, its time complexity increases with a greater number of concave vertices.

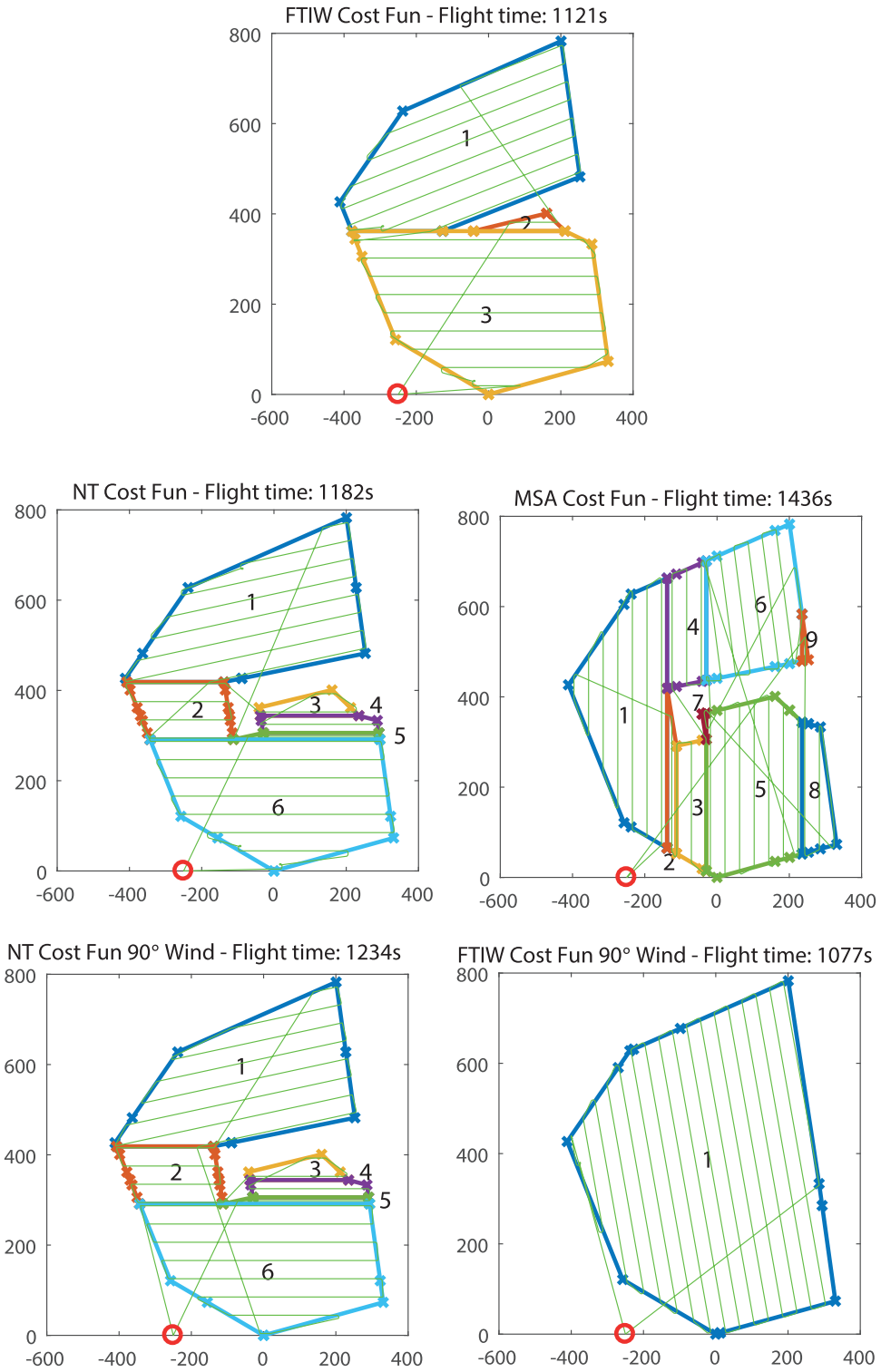


FIGURE 11 Field polygon from Figure 9b decomposed and flight paths generated using FTIW, MSA, and NT methods. This is done for both southerly and westerly winds. Predicted flight times are shown. FTIW, flight time in wind; MSA, minimal sum of altitude; NT, number of turns [Color figure can be viewed at wileyonlinelibrary.com]

To show how computational complexity increases with polygon complexity, one of 20 random polygons has its polygon vertex number reduced to just four vertices (a single trapezoid), then its optimization runtime is recorded for both RGS and DP methods. Then vertices are added back one at a time and the runtime recording is repeated until the mission for the entire polygon has been made. This is then repeated for each of the 20 polygons and an average runtime is calculated for each vertex count. The results of this computation time comparison is shown in Figure 14. In the simpler runs with a vertex count <17, the DP method is faster due to the less computationally intensive cost

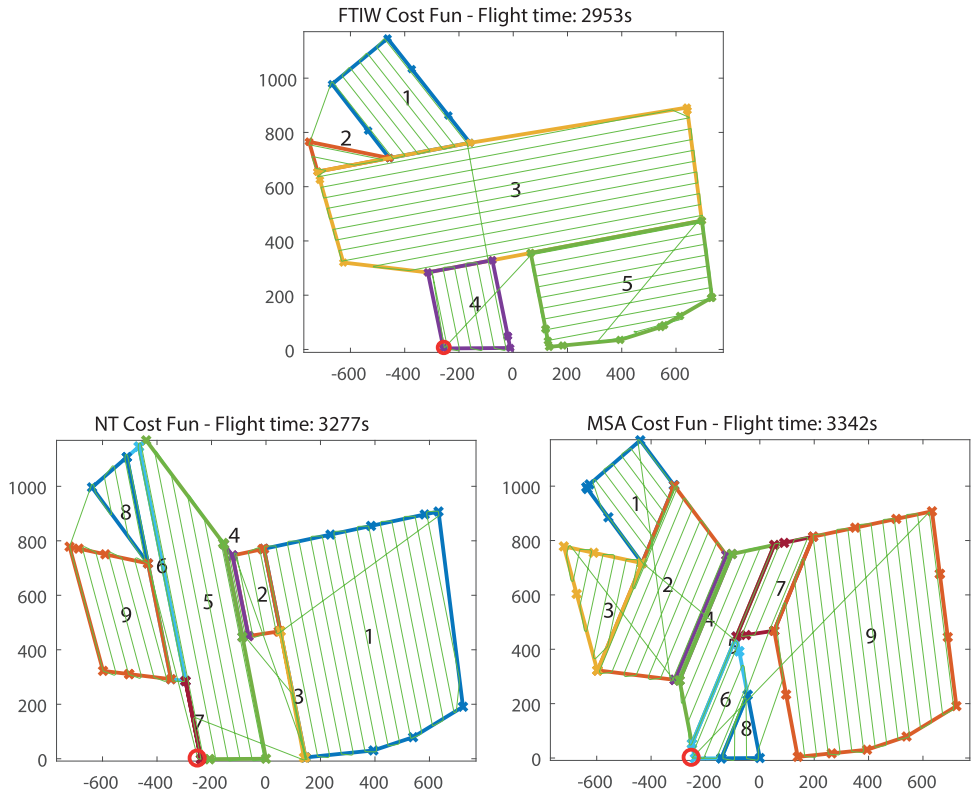


FIGURE 12 Field polygon from Figure 9c decomposed and flight paths generated using FTIW, MSA, and NT methods. Predicted flight times shown. FTIW, flight time in wind; MSA, minimal sum of altitude; NT, number of turns [Color figure can be viewed at wileyonlinelibrary.com]

function. However, as the vertex count crosses 17 cells, the proposed technique becomes much faster; quickly becoming orders of magnitude faster than the DP method. For example, at 20 vertices, the DP method took 1,720 times longer than RGS and it was not feasible to run the DP algorithm on polygons with a vertex count greater than this due to its computational time.

4.3 | Monte-Carlo simulation

To present a fair numerical comparison between the proposed method and those from the literature, 30 polygons have been randomly generated between 11 and 16 vertices. The improvement offered by using FTIW over NT and MSA is assessed. In this case, the

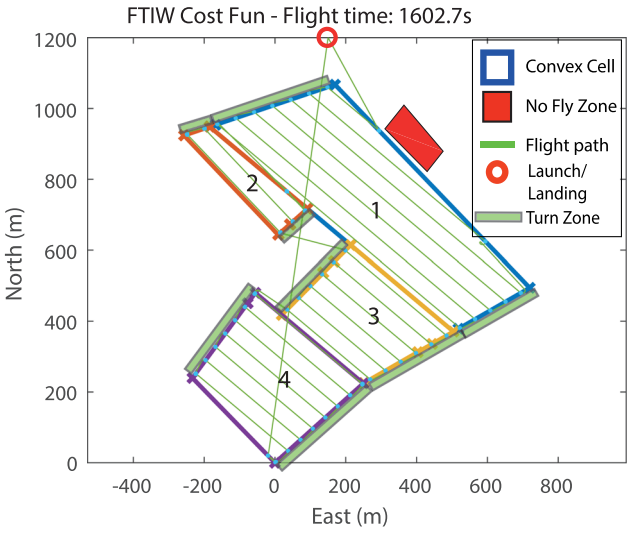


FIGURE 13 Field polygon from Figure 9a decomposed and flight paths generated using the FTIW and NFZ cost functions to plan a fast route in wind while avoiding NFZs. FTIW, flight time in wind; NFZ, no-fly zone [Color figure can be viewed at wileyonlinelibrary.com]

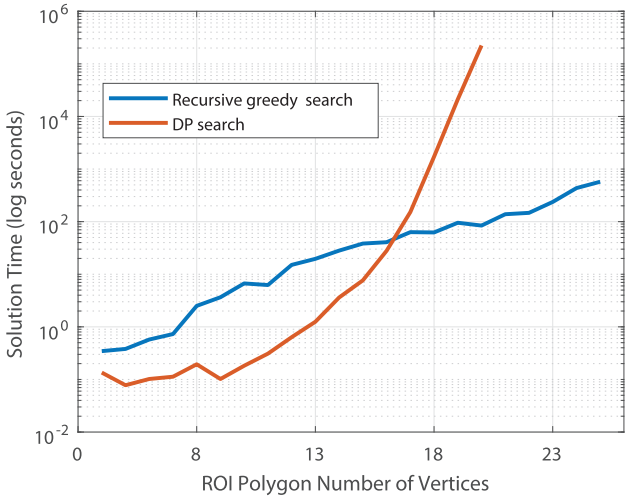


FIGURE 14 Average computation time for both DP and RGS methods against a number of polygon vertices in the field to plan a mission for. DP, dynamic programming; RGS, recursive greedy search; ROI, region of interest [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 4 Simulated X8 and survey parameters

Parameter	Value
Wind	270°/5 m/s
I	15.5 m/s
$\dot{\psi}$	0.7 rad/s
w_s, w_o	0.3, 0.3
h	120 m
GSD	8.2 cm/pixel
D_x	73 m

Abbreviation: GSD, ground sample distance.

platform used is the popular Skywalker X8, a flying-wing aircraft that flies at a higher airspeed of 16 m/s (leading to higher turn radii). This has been chosen because it matches the aircraft which is used for flight testing in Section 5. The GSD requirements have been relaxed for this flight to 8.2 cm/pixel; meaning that the survey can be conducted at a higher flight altitude of 120 m for logistical reasons. The full aircraft and survey parameters are listed in Table 4.

Using identical sets of 30 random polygons, the simulation is run with five different wind strengths that range from 0 to 13 m/s in an easterly direction ($\psi_w = 270°$). This is a large variation, which represents between 0% and 81% of the aircraft's airspeed. The results of these Monte-Carlo simulations are shown in Table 5, and a few example polygons and the generated paths are shown in Figure 15. It can be seen that under all wind conditions, using the proposed method gives a significant flight time advantage. When the wind speed is increased, the advantage of the FTIW cost function becomes even greater.

The first run has zero wind speed, which means that the advantage provided by corner cutting is isolated. It can be seen that this has a significant affect on the results, providing a 9.3% reduction in flight time compared to the NT algorithm and an 8.6% reduction compared to the MSA cost function.

As the wind speed increases, the flight time advantage increases further. At low wind speeds (5 m/s, 30% of airspeed), the FTIW advantage increases marginally up to 9.6% compared to the NT cost function and reduces marginally down to 8.4% compared to the MSA algorithm. However, for high wind speeds (13 m/s, 81% of airspeed), the advantage of the prior wind knowledge is much more significant, increasing to 19% and 15.4% compared to NT and MSA, respectively.

A weak correlation has also been found between the complexity of the polygon and the FTIW advantage as shown in Figure 16. With an increase in the number of vertices and especially concave vertices, the polygon becomes more complex. Because the alternative methods do not allow for corner cutting, their pathways tend to be based on a higher number of decomposed cells, increasing the path complexity and the transit time between cells. Therefore, it is concluded that with an increase in either polygon complexity or an increase in wind speed, the advantages of using the proposed method is maximized.

TABLE 5 Monte-Carlo simulation results—30 random polygons runs—FTIW comparison to MSA and NT

		FTIW improvement over NT		FTIW improvement over MSA	
#	Description (m/s)	Av time (s)	Av percentage (%)	Av time (s)	Av percentage (%)
1	$V_w = 0$	192	9.3	172	8.6
2	$V_w = 3$	201	9.5	177	8.5
3	$V_w = 5$	210	9.6	184	8.4
4	$V_w = 10$	399	13.6	346	11.2
5	$V_w = 13$	846	19	749.6	15.4

Abbreviations: FTIW, flight time in wind; MSA, minimal sum of altitude; NT, number of turns.

4.4 | Monte-Carlo RGS comparison to DP

Finally, the results of the proposed RGS method and the previous DP method from Coombes, Fletcher, et al. (2018) are compared. It should be noted that the previous method uses a simpler FTIW cost function that does not account for transit paths to keep the computational time acceptable. To make a fair comparison, the cost function of the RGS method has been reverted to the previous FTIW cost function to match.

With an easterly wind and the same survey parameters as the previous Monte-Carlo simulations above, 30 polygons have had missions planned using both the DP and RGS optimization methods. It has been found that the suboptimal RGS method presented here results in an average increase in the flight time of just 2.3% compared to the DP solution. This is acceptable given the massive increase in computational efficiency attained, enabling calculations to be performed in the field.

Three example polygons with both the DP and RGS solutions are shown in Figure 17. In the uppermost Figure 17a is an example where the DP method creates a slower solution. The reason for this is that the RGS finds a decomposition where the transit distance between both cells is shorter than for the DP method. In Figure 17b both methods have the exact same solution. In Figure 17c the DP method has found a solution 2% faster than the proposed method, this is as there is only a single concave vertex; making the search space for the RGS small.

5 | SURVEY FLIGHT TESTING

To assess the proposed algorithm under real-world conditions, a number of actual fields have been surveyed using the RGS CPP method presented here to generate missions. These missions will be assessed for predicted time accuracy from the FTIW model, compared to alternative mission planning algorithms and to demonstrate the effectiveness of the NFZ avoidance constraint. For each flight, the fixed-wing survey UAV autonomously flies the

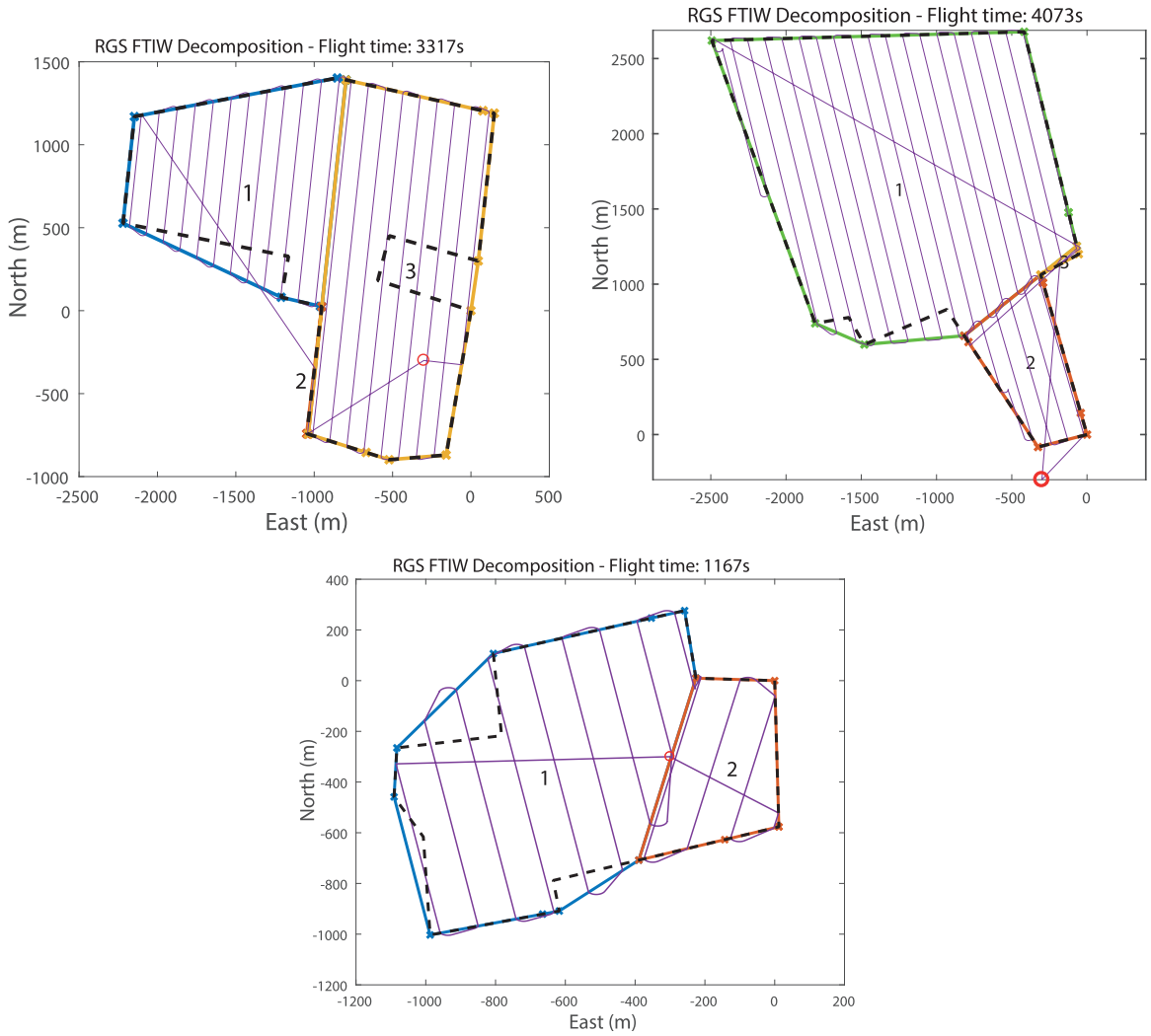


FIGURE 15 Three example random polygons from the 30 Monte-Carlo simulations with flight paths generated for RGS with the FTIW cost function. FTIW, flight time in wind; RGS, recursive greedy search [Color figure can be viewed at wileyonlinelibrary.com]

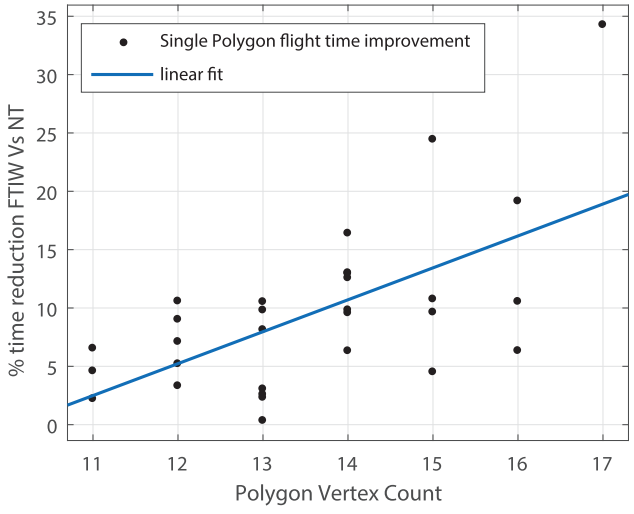


FIGURE 16 Correlation between the number of polygon vertices and flight time reduction using FTIW cost function with corner cutting, against using NT cost function without corner cutting. FTIW, flight time in wind; NT, number of turns [Color figure can be viewed at wileyonlinelibrary.com]

automatically generated CPP way-points using the L1 adaptive navigation controller available on the Pixhawk autopilot used on the test aircraft (Park, Deyst, & How, 2004). The desire was to test the CPP algorithm in a highly realistic scenario using the exact type of UAV, autopilot, sensors, and navigation algorithm that would be used for agricultural surveys.

5.1 | Survey flight platform and data collection

The Skywalker X8 is the fixed-wing UAV used for surveying. It has a 40 min endurance at 16 m/s, giving it approximately 38 km of range. It is fitted with an autopilot to automatically control all aspects of the aircraft during flight. The aircraft and its systems can be seen in Figure 18 and whose major flight parameters are listed in Table 6.

The electronic set-up consists of a Pixhawk autopilot with PX4 firmware, Pixhawk airspeed sensor, GPS, Raspberry Pi 2, Arduino Nano, AeroProbe-Micro-ADC, and a MicaSense RedEdge multispectral camera. Pixhawk is a high-performance autopilot-on-module suitable

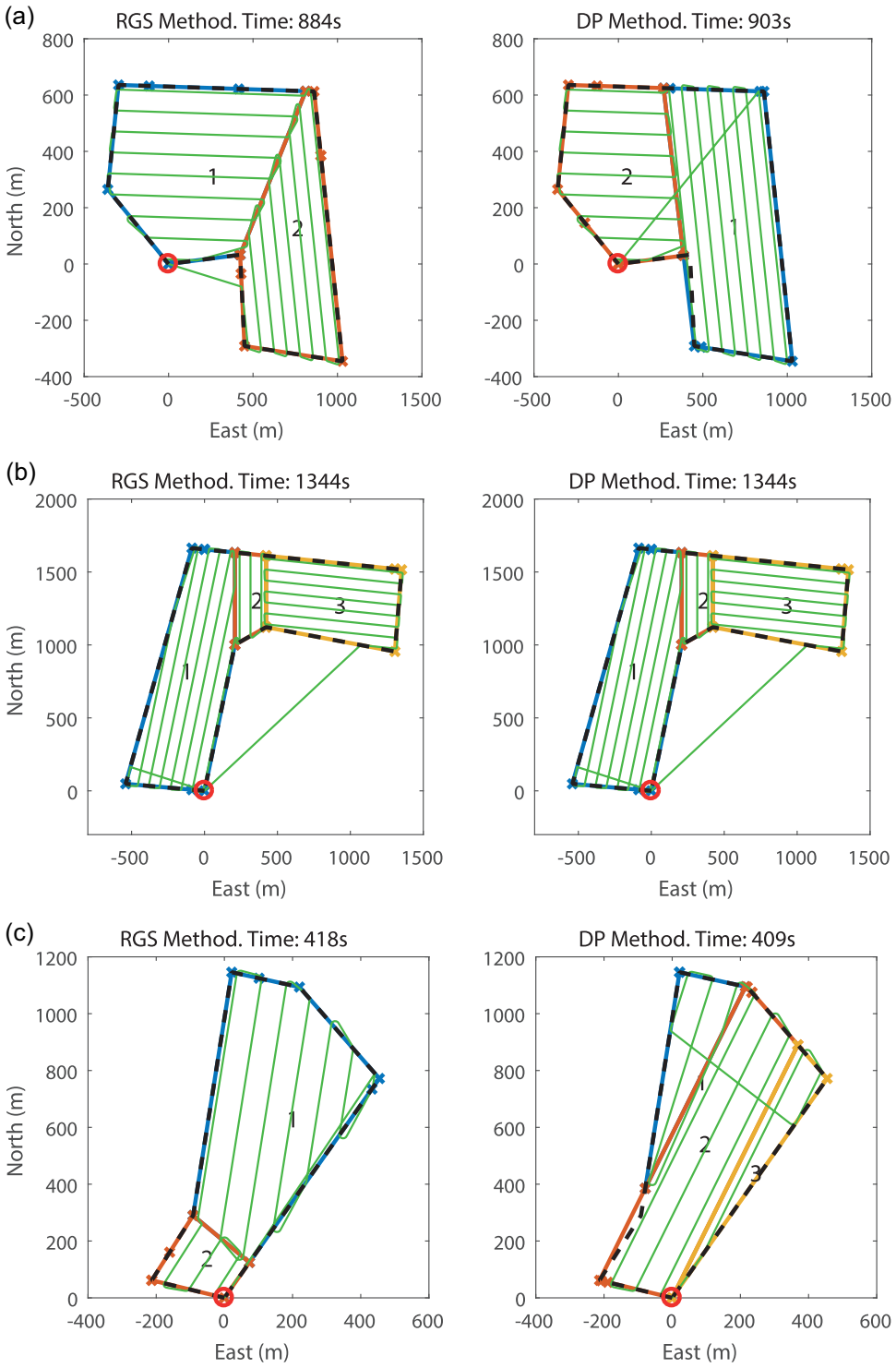


FIGURE 17 Three example random polygons from the 30 polygon Monte-Carlo simulation, with paths and times generated both by the RGS and the DP method (a) seven vertex random field polygon—example of DP method being slower; (b) seven vertex random field polygon—example of DP method with the same solution; (c) six vertex random field polygon—example of DP method being faster. DP, dynamic programming; RGS, recursive greedy search [Color figure can be viewed at wileyonlinelibrary.com]

for fixed-wing and multirotor aircraft, helicopters, cars, and boats. By intercepting the autopilot telemetry on a Raspberry Pi 2 running ROS and a ROS package called MAVROS, all the autopilot telemetry data can be logged in real-time. Further information regarding the vehicle set-up is laid out in detail in Koubaa (2018).

To perform accurate measurements of the wind, a more accurate airspeed sensor than the low-cost sensor that is used by Pixhawk is needed. The sensor chosen was the AeroProbe-Micro-ADC, which is a high accuracy multihole Pitot static probe and data logger.



FIGURE 18 The Skywalker X8 fixed-wing aerial survey flight test platform [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 6 Parameters for the X8 flying wing

Parameter	Value
Wing area	0.734 m^2
Aspect ratio	7.48
Mass	2.75 kg
V_{stall}	10 m/s
V_{cruise}	16 m/s

Using another ROS package, ROSSerial, enables the addition of other sensors by interfacing with an Arduino Nano which has a range of I/O available that the Raspberry Pi lacks, such as UART, analogue inputs, SPI and so forth. This enables the parsing of the other serial link from AeroProbe which contains the airspeed, Angle of attack, and angle of sideslip data. This data is published to the ROS network as user-defined topics. All the relevant topics related to air data, IMU, GNSS, and AeroProbe are recorded using ROSBags. These can then easily be postprocessed by MATLAB's robotics system toolbox (RST) for analysis.

TABLE 7 Survey and aircraft parameters used for accuracy assessment flights

Parameter	Value
Wind	180°/1 m/s
V	15.5 m/s
$\dot{\psi}$	0.7 rad/s
w_s, w_o	0.3, 0.3
h	100 m
GSD	6.9 cm/pixel
D_x	62 m

Abbreviation: GSD, ground sample distance.

6 | FLIGHT TEST RESULTS

In total, four survey sites were chosen to be surveyed; each with different sizes and complexities. The experiments have been broken down into three sections: flight time accuracy, comparison between this method and those from the literature and demonstrating the avoidance of NFZs. In addition to this, total image coverage with the RedEdge multispectral camera is demonstrated for one of the survey sites, this is presented in Appendix A.

It should be noted that due to current regulations, UAVs are unable to fly more than 500 m away from the safety pilot, meaning that the size of fields used for testing are smaller than ideal. This algorithm lends itself to larger survey sizes, where a full survey time would be closer to the endurance of the UAV.

6.1 | Accuracy assessment

Figure 19 shows the field polygon to survey, the decomposition (projected above the polygon for clarity), and the way-points generated

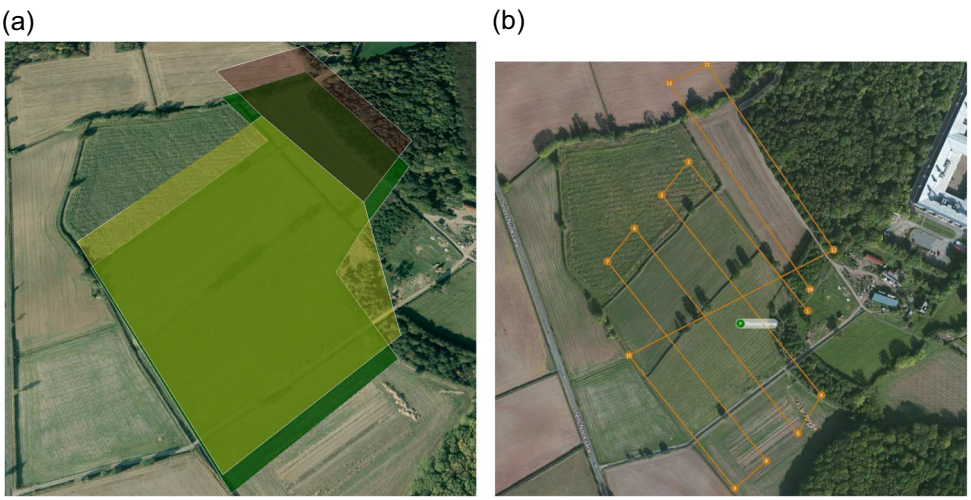


FIGURE 19 Google maps images of decomposition and way-points used for accuracy assessment surveys (a) field polygon and its RGS FTIW decomposition and (b) CPP way-points generated, to be uploaded to autopilot. CPP, coverage path planning; FTIW, flight time in wind; RGS, recursive greedy search [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 8 FTIW model flight time prediction accuracy

Day#	Measured wind (m/s)	FTIW time (s)	Actual times (s)	Time error (%)	Model dist (m)	Actual (m)	Dist error (%)
1	00°/0.7	177	184 182	3.9 2.8	2,745	2,871	4.6
2	340°/5	205	196 199	−4.4 −2.9	2,863	2,872	0.3
3	310°/6	216	209 212	−3.2 −1.9	2,872	2,912	1.7

Abbreviation: FTIW, flight time in wind.

by the algorithm. It is good practice to extend the way-points further along the sweep line so it has extra distance outside of the polygon to turn and level off before it re-enters the polygon to ensure all photos are orthogonal to the ground. In this case, a 20 m overshoot distance is applied to all way-points, which is why through the way-points are seen outside of the ROI polygon. For NFZ calculations, this extra 20 m will be added to the distance R_t. These sets of way-points were generated for the wind conditions for the first day of testing which is a light northerly wind at 2 kt as shown in Table 7.

As previously stated, flying perpendicular to the wind is more efficient, however, in this case, as the wind was very low, the solution results in sweep directions of both decomposed cells flying nearly directly in to and out of the wind. This is due to the shape of the field lending itself to less wasteful turns in that direction, which outweighs the small detrimental effect of the wind at this low strength. To show the time prediction accuracy, this flight was repeated on an additional 2 days to get a range of wind conditions, and for each day, the survey is conducted twice. Under other wind conditions, this might not be the

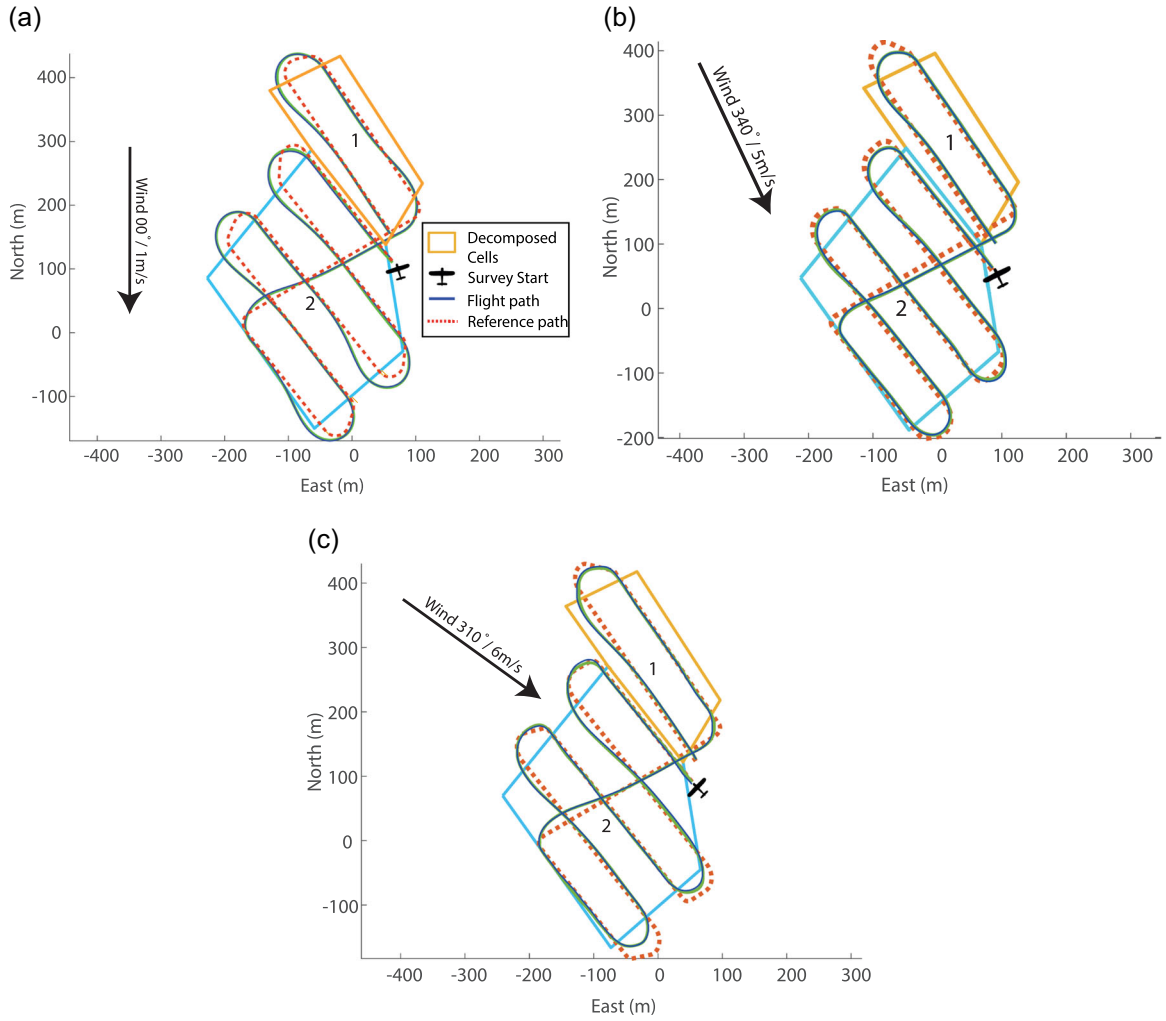


FIGURE 20 Accuracy assessment surveys, on 3 separate days, with two surveys per day. Comparison between calculated and actual flight paths (a) Day 1: Wind 00°/1 m/s; (b) Day 2: Wind 340°/5 m/s; and (c) Day 3: Wind 310°/6 m/s [Color figure can be viewed at wileyonlinelibrary.com]

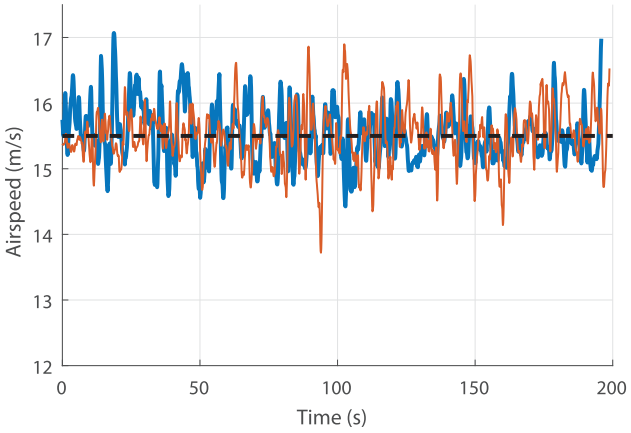


FIGURE 21 Airspeed compared to reference for both surveys on Day #2 [Color figure can be viewed at wileyonlinelibrary.com]

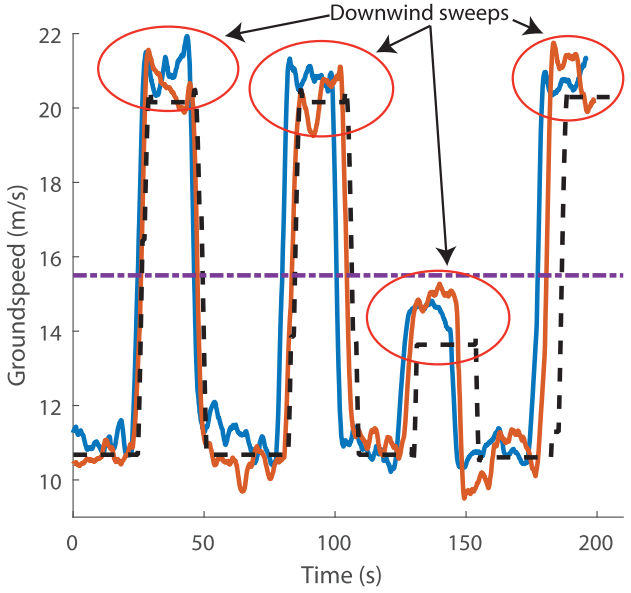


FIGURE 22 Ground speed comparison between predicted and actual for 340°/5 m/s flight on #2 [Color figure can be viewed at wileyonlinelibrary.com]

optimum decomposition, however, the mission plan was kept the same for comparison. Table 8 shows the predicted time and flight distance compared to the actual time and distance flown by the X8. The flight paths flown are presented alongside the ideal path in Figure 20.

The time error percentage for all six surveys is low; with values ranging between −4.4% and +3.9%. There are three sources of error: path following errors, airspeed errors, and wind errors. As can be seen from Figure 20, the path following was not perfect; there was some overshoot when turning downwind, and slight undershoot when turning upwind. This led to path distance errors of between 0.3% and 4.6% which goes some way to explain the small-time error, but for Days #2 and #3, the actual mission was faster than predicted, so a longer flight path does not explain this. Shown in Figure 21 is the measured airspeed during the two surveys on Day #3. Although it does a reasonable job of airspeed tracking, there is quite a lot of noise due to gusting and the aggressive turn maneuvers performed. Good ground speed prediction is essential for good time accuracy, and any errors in airspeed and wind will combine. Shown in Figure 22 is the vehicle's ground speed compared to that predicted. It also does a good job of tracking, however, there is a slight over-speed in the downwind turns, which has to lead to the actual survey being faster than the predicted.

While only six flights were flown that were used solely for FTIW model validation, over the next two subsections, 13 more flight tests have been performed across a wide range of wind conditions and field shapes with errors in the flight time prediction ranging from −6.8% to +11.0%. The average error across all surveys was 4.13%.

6.2 | NFZ flight testing

As described in Section 2.4, operators often need to avoid overflying NFZs. Within the proposed technique there is a means to do this. The two survey locations selected to evaluate the effectiveness of NFZ avoidance are shown in Figure 23. The first area (Figure 23a) is a small group of fields with a total area of 11.5 ha. This area is adjacent to a farmhouse which will be the simulated NFZ. However, the single field in Figure 23b is much larger at 18.4 ha and has two adjacent farmhouses which will both be the simulated NFZs. Due to the large size of the second field, if the survey was conducted with the previous parameters, the survey would take longer than the aircraft's battery life, therefore, the survey is shortened by flying with a side lap (w_s) of 20%, leading to an increased sweep spacing (D_x) of 85 m, and at a slightly faster airspeed of 16.5 m/s. For each field, two missions will be generated with the proposed algorithm; one inclusive

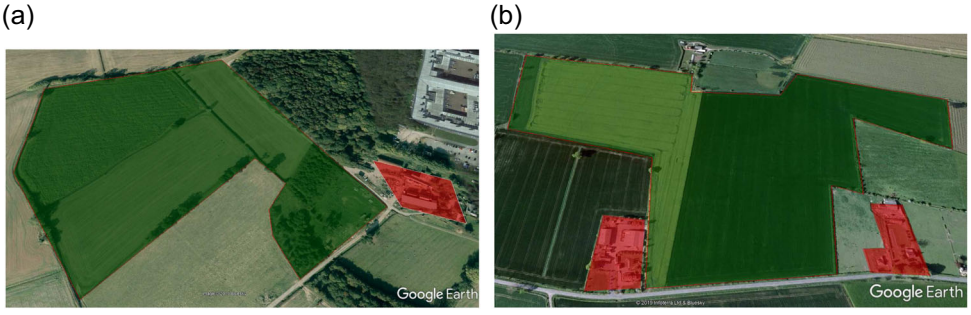


FIGURE 23 Two flight-test fields with NFZs adjacent to the survey area. ROI and NFZ polygons are overlaid in green and red, respectively. NFZ, no-fly zone; ROI, region of interest [Color figure can be viewed at wileyonlinelibrary.com]

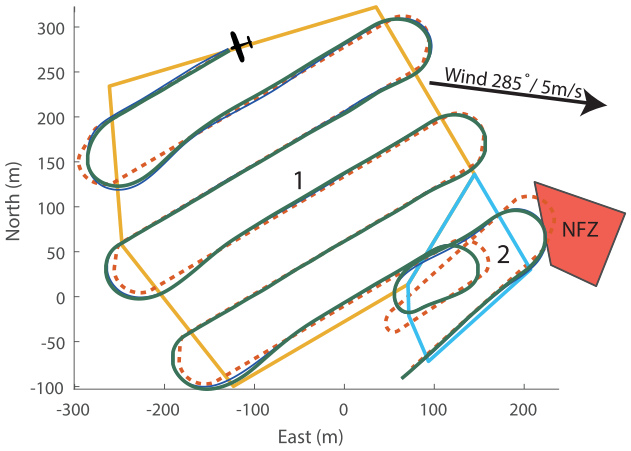


FIGURE 24 Actual survey flight path compared to calculated for FTIW decomposition of a small field, where NFZ is violated. FTIW, flight time in wind; NFZ, no-fly zone [Color figure can be viewed at wileyonlinelibrary.com]

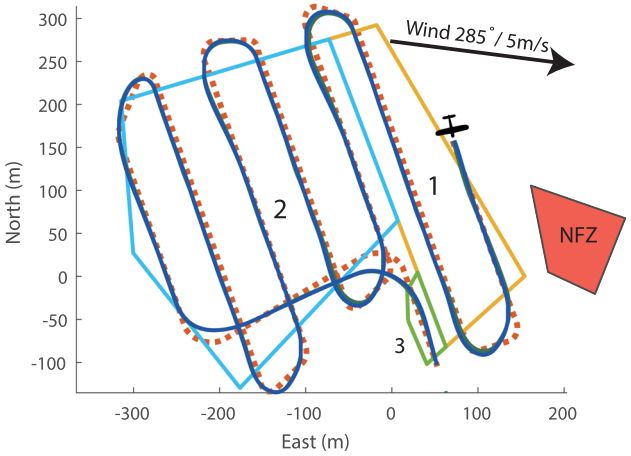


FIGURE 25 Actual survey flight path compared to calculated for FTIW and NFZ cost function decomposition of small field. FTIW, flight time in wind; NFZ, no-fly zone [Color figure can be viewed at wileyonlinelibrary.com]

of the NFZ constraint from Equation (8) and one neglecting it. Note that during this survey, full permission to overfly the buildings within the NFZs was obtained.

For the small field, the calculated path is plotted against the actual survey flight paths for both generated missions in Figure 24 and Figure 25. This survey is flown with the parameters from 7, however, with wind from 285° at 5 m/s. The predicted and actual flight times and distances are displayed in Table 9. It can be seen that neglecting the NFZ constraint results in the aircraft overflying the NFZ, however, when it is included, a decomposition is found where the NFZ is avoided with an associated time penalty of 8 s. During the flight tests, however, the measured survey times very similar suggesting that this penalty is within the small error bounds (±2%) of the time prediction.

Two more flight tests were conducted over the larger field from Figure 23b; these are shown in Figure 26 exclusive of the NFZ constraint and in Figure 27 inclusive of the constraint. As before, both are repeated to give four surveys overall. The conditions on that day were challenging with wind from 330° at 10 m/s and gusting, representing 60% of the planned airspeed. The NFZ exclusive decomposition was able to remove four of the concave vertices to make a simple two-cell decomposition that allowed for a low number of turns and transitions. Despite the high wind speed, the shape of the field outweighed the benefit of flying perpendicular to the wind. In contrast, the path inclusive of NFZ avoidance is more complex with three cells and longer path transitioning between cells. This lead to a significant increase (70 s, 21%) in the predicted flight time. However, this has allowed the survey direction of cell 2 to be aligned in such a way as to avoid the NFZ. Due to the challenging wind conditions, the error in flight time was much larger than for the smaller field, with the NFZ exclusive tests taking around 5% longer than predicted and the NFZ inclusive flights finishing 11% and 9.9% earlier than predicted. Detailed analysis of the flight paths has concluded that due to the NFZ path being more complex and having more turns downwind, the autopilot's path planner tended to make turns earlier, before the way-point was reached, to avoid huge overshoot due to the wind. This leads to lower path distances than predicted, resulting in lower flight times than predicted.

6.3 | Comparison to MSA and NT

To demonstrate that the proposed algorithm generates faster mission paths than the previous methods, two fields will be surveyed using the proposed algorithm with the FTIW cost function. In addition, the larger field (Figure 23b) from the previous section will be surveyed with a MSA cost function CPP solution, and a smaller 15.5 ha *Z*-shaped field with only three concave vertices will be surveyed with using the NT cost function.

TABLE 9 Time and path distance comparison for both survey fields between FTIW and NFZ cost function

Cost Fun	Wind (m/s)	Model time (s)	Actual times (s)	Time error (%)	Model dist (m)	Actual (m)	Dist error (%)
Small field (Figure 23a)							
NFZ	285°/5	217	212, 218	2.03, 0.55	3,131	3,027	3.32
FTIW	285°/5	208	212, 210	−1.5, −0.9	3,117	3,079	1.24
Large field (Figure 23b)							
NFZ	330°/10	396	352, 360	11, 9.9	4,505	4,360	3.17
FTIW	330°/10	327	343, 343	−5, −5	4,452	4,594	1.7

Abbreviations: FTIW, flight time in wind; NFZ, no-fly zone.

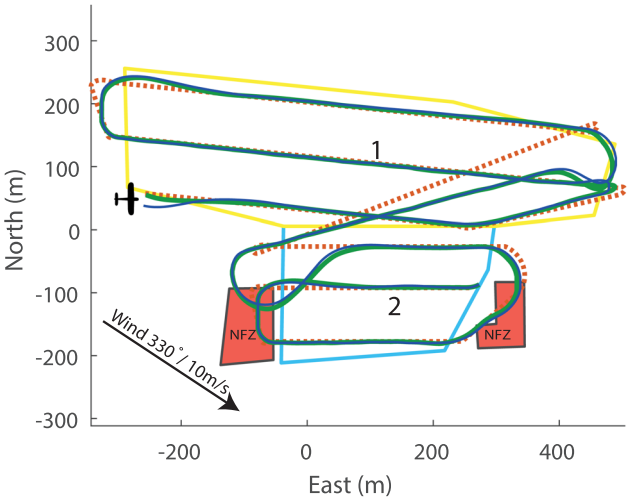


FIGURE 26 Actual survey flight path compared to calculated for FTIW decomposition of a large field, where NFZ is violated. FTIW, flight time in wind; NFZ, no-fly zone [Color figure can be viewed at wileyonlinelibrary.com]

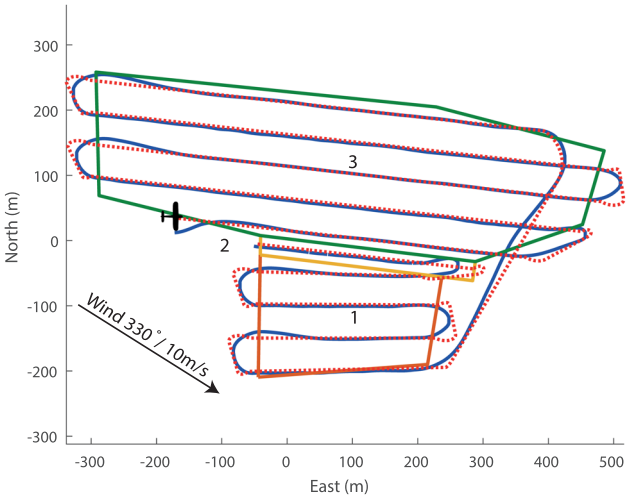


FIGURE 28 Calculated and actual flown path from FTIW cost function decomposition survey of large complex field shape with greater image overlap, to compare to alternative methods. FTIW, flight time in wind [Color figure can be viewed at wileyonlinelibrary.com]

The decompositions, predicted and actual survey paths are shown for the large field FTIW mission in Figure 28, and for the MSA mission in Figure 29. Survey time comparisons are shown in Table 10. The FTIW decomposition survey took 476.8 s compared to 779 s for the MSA survey, representing a 38.8% improvement. As was seen during simulations, the MSA decomposition results in a much larger six cell decomposition compared to three cells for FTIW, this led to an increased planned flight path of 7,631 m compared to just 6,235 m for the FTIW cost function. Due to the shape of the field and the relative wind direction, there was little advantage to flying perpendicular to the wind, because that would increase the path length significantly. If the very strong wind was from 270°, a decomposition similar to that from Figure 27 would have been more beneficial.

On the day of the Z-shaped field surveys, the wind was weaker from 310° at 6 m/s. The shape of this field lends itself well to finding an advantage from flying perpendicular to the wind. Shown in Figure 30, the FTIW decomposition has found a solution that aligns the sweeps perpendicular to the wind across both cells. In comparison, the NT solution in Figure 31 has the aircraft flying directly into and out of the wind. From Table 10 it can be seen that the FTIW predicted time of 311 s is 15% faster than NT at 357.8 s. During the actual flights, the disparity increases resulting in 300 s for FTIW, compared to 375 s for NT which represents 20.1% difference. This larger difference observed in the real flights is due to the prediction for FTIW path being too high. This was a result of two automatically generated cell transit way-points between cells 1 and 2 being very close together, meaning that the trochoid calculation could not find a turn solution without turning more than 360°.

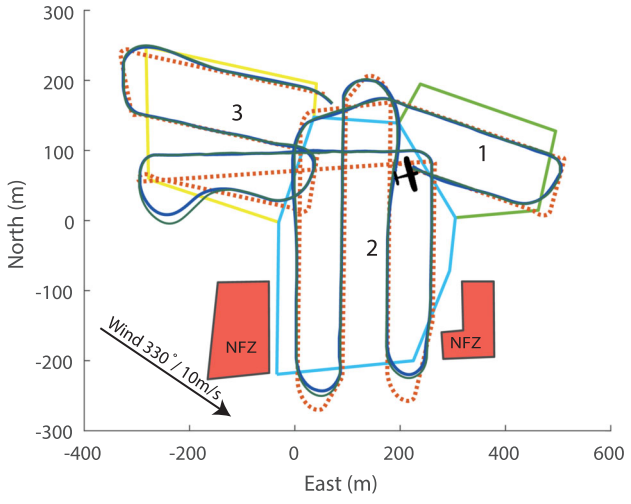


FIGURE 27 Actual survey flight path compared to calculated for FTIW and NFZ cost function decomposition of a large field. FTIW, flight time in wind; NFZ, no-fly zone [Color figure can be viewed at wileyonlinelibrary.com]

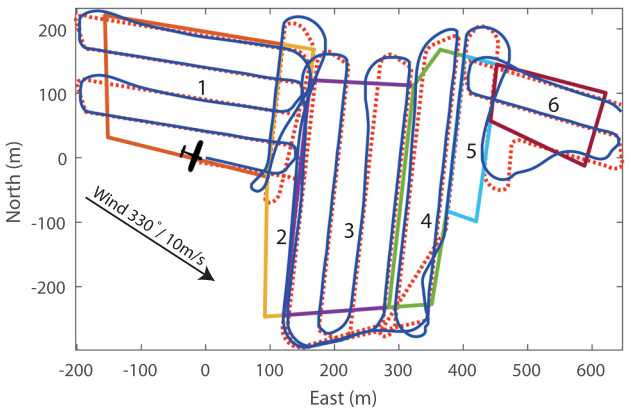


FIGURE 29 Calculated and actual flown path from MSA cost function decomposition survey of large complex field shape with greater image overlap, to compare to FTIW cost function. FTIW, flight time in wind; MSA, minimal sum of altitude [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 10 Time and path distance comparison for both survey fields between FTIW and MSA and NT cost functions

Cost Fun	Wind (m/s)	Model time (s)	Actual times (s)	Time error (%)	Model dist (m)	Actual (m)	Dist error (%)
Large field (Figure 23b)							
FTIW	330°/10	502	477	4.98	6,236	6,211	0.39
MSA	330°/10	730	779	−6.81	7,632	7,500	1.17
FTIW time % improvement over MSA				38.8			
Z-shaped field							
FTIW	310°/6	311	303	2.57	4,166	4,049	2.8
			296	4.86		4,046	2.86
NT	310°/6	358	375	−4.8	4,989	5,163	−3.48
FTIW time % improvement over NT				20.1			

Abbreviations: FTIW, flight time in wind; MSA, minimal sum of altitude; NT, number of turns.

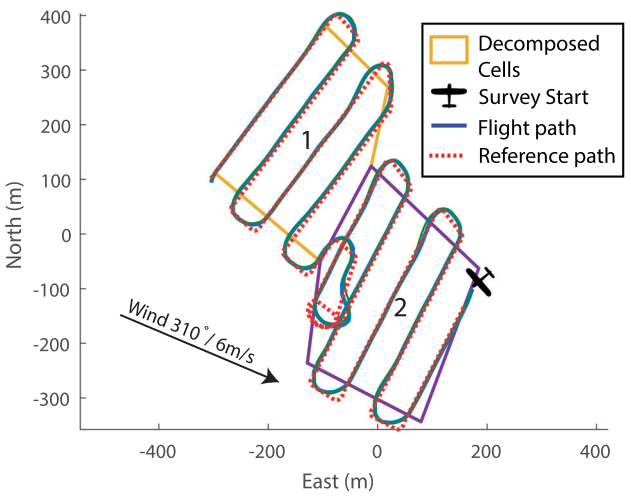


FIGURE 30 Calculated and actual flown path from FTIW decomposition survey of Z-shaped survey field, to compare to MSA cost function decomposition. FTIW, flight time in wind; MSA, minimal sum of altitude [Color figure can be viewed at wileyonlinelibrary.com]

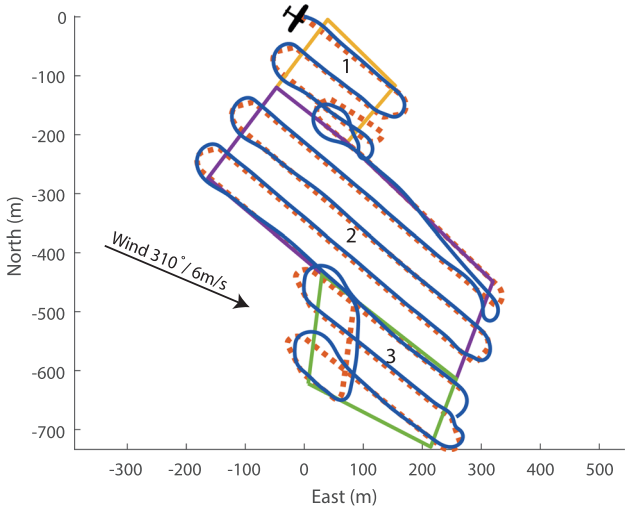


FIGURE 31 Calculated and actual flown path from MSA decomposition survey of Z-shaped survey field, to compare to FTIW cost function decomposition. FTIW, flight time in wind; MSA, minimal sum of altitude [Color figure can be viewed at wileyonlinelibrary.com]

However in the actual flight, the autopilot instead flew very wide; making the actual path length smaller. This double turn is shown in Figure 30.

7 | CONCLUSION

A CPP method has been developed to plan fixed-wing aerial survey missions for complex irregular fields in wind. The field polygon is decomposed into a small number of convex polygons using a fast, but suboptimal RGS) technique, and are each covered using a boustrophedon path. The algorithm finds a minimum flight time decomposition based on the FTIW cost function. It was also shown that by modifying the cost function, real world considerations can be accounted for, for example, ensuring the survey does not fly into user-defined NFZs.

The suboptimality of this method has compared to the optimal DP technique and produced survey paths on average only 2.3% slower. However the DP method is shown it be significantly slower computationally. For example, for a field polygon with 20 vertices, the DP method took, on average, 1,720 times longer to solve than the new proposed RGS.

A number of simulations were performed to demonstrate the benefits of this method compared to those already published in the literature. It was shown that there can be a significant reduction in flight time by accounting for wind, as the sweep direction relative to a strong wind is critical for planning an efficient mission. For example, it was shown with a strong wind speed of 70% of the aircraft's airspeed, a 19% time advantage could be obtained. By letting the aircraft cut corners, and allowing overflying of areas outside of the field polygon, the final decomposition can be simplified. This results in fewer decomposed cells, reducing wasted time transitioning between cells. These features were also demonstrated to give a flight time advantage over previous methods during real world survey flight tests. In both simulations and experimental work, it has been shown that the advantage of the proposed algorithm is greatest for more complex ROIs, larger ROIs and for stronger wind conditions.

Further flight testing was performed to show that the FTIW model was accurate enough to make good flight time predictions to be used in the real world. It was shown that over 19 surveys, the average absolute error was just 4.13%, which is largely accounted for

by the autopilots airspeed tracking during turning maneuvers. Finally, the assumption of a steady uniform wind over typical survey areas and time-scales was experimentally testing using field measurements from a multirotor aircraft. Variation in wind was small, with standard deviations of 0.627 m/s in strength and 10.72° in direction. This was further shown to be true as ground speed measured during the survey flights matched closely with the prediction.

7.1 | Future work

There are a number of potential improvements to this study that have not yet been implemented. In a few surveys, it was found that the paths generated to transition between cells can require more aggressive maneuvers than the aircraft can perform. As there are only four possible starting points and the start point of the next cell may be just meters away from the end of the current cell survey path, the aircraft would have to perform an unnecessary 360° maneuver to line up for the next start point. However these artefacts could be eliminated if the cells did not have to be sequentially and fully surveyed before moving on to the next cell. If each generated sweep path generated from the RGS technique is used in a method like that developed by Richards (2018), a path can be generated that has the aircraft traverse all sweep paths in any order without forcing any impossible maneuvers.

Second, it has been observed that overshoot of way-points will not only cause time errors but more important, it can cause image locations to be missed, meaning perhaps a complete orthophoto is not obtained. In high winds, this can be mitigated by attempting to make turns into wind as this will keep the ground speed low so the autopilot will follow the path more accurately. This consideration could be included in the FTIW cost function, or it could be included in the method discussed above by not only removing impossible maneuver but also minimizing the number of downwind turns performed.

ACKNOWLEDGMENTS

The authors would also like to thank FTTechnologies for the use of and support of their lightweight FT205EV ultrasonic wind sensor. This study was supported by the Science and Technology Facilities Council (STFC) under Newton Fund with grant number ST/N006852/1.

ORCID

Matthew Coombes http://orcid.org/0000-0002-4421-9464

REFERENCES

Acar, E. U., Choset, H., Rizzi, A. A., Atkar, P. N., & Hull, D. (2002). Morse decompositions for coverage tasks. *The International Journal of Robotics Research*, *21*(4), 331–344.

Alta Devices (2017). Beyond visual line of sight (BVLOS).

Anderson, C. (2014). Agricultural drones. *MIT Technology Review*, *117*(3), 58–60. https://www.technologyreview.com/s/526491/agricultural-drones/

Arney, T. (2007). An efficient solution to autonomous path planning by approximate cell decomposition. *2007 Third International Conference on Information and Automation for Sustainability* (pp. 88–93).

Avellar, G. S. C., Pereira, G. A. S., Pimenta, L. C. A., & Iscold, P. (2015). Multi-UAV routing for area coverage and remote sensing with minimum time. *Sensors*, *15*(11), 27783–27803.

Bendig, J., Bolten, A., Bennertz, S., Broscheit, J., Eichfuss, S., & Bareth, G. (2014). Estimating biomass of barley using crop surface models (CSMS) derived from UAV-based rgb imaging. *Remote Sensing*, *6*(11), 10395–10412.

Ceccarelli, N., Enright, J. J., Frazzoli, E., Rasmussen, S. J., & Schumacher, C. J. (2007). Micro UAV path planning for reconnaissance in wind. American Control Conference, 2007. ACC'07, IEEE (pp. 5310–5315).

Choset, H. (2000). Coverage of known spaces: The boustrophedon cellular decomposition. *Autonomous Robots*, *9*(3), 247–253.

Civil Aviation Authority. (2012). *Cap 722 unmanned aircraft system operations in UK airspace* (Technical Report).

Colomina, I., & Molina, P. (2014). Unmanned aerial systems for photogrammetry and remote sensing: A review. *{ISPRS} Journal of Photogrammetry and Remote Sensing*, *92*, 79–97.

Coombes, M., Chen, W., & Liu, C. (2018). Fixed wing uav survey coverage path planning in wind for improving existing ground control station software. *2018 37th Chinese Control Conference (CCC)* (pp. 9820–9825).

Coombes, M., Chen, W. H., & Liu, C. (2017). Boustrophedon coverage path planning for uav aerial surveys in wind. *2017 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 1563–1571).

Coombes, M., Fletcher, T., Chen, W.-H., & Liu, C. (2018). Optimal polygon decomposition for uav survey coverage path planning in wind. *Sensors*, *18*(7). 2132.

Department for Transport (UK) (2016).*Unlocking the UK's high tech economy: Consultation on the safe use of drones in the UK* (Technical Report).

Depenbusch, N. T., Bird, J. J., & Langelaan, J. W. (2018a). The autosoar autonomous soaring aircraft, part 1: Autonomy algorithms. *Journal of Field Robotics*, *35*(6), 868–889.

Depenbusch, N. T., Bird, J. J., & Langelaan, J. W. (2018b). The autosoar autonomous soaring aircraft part 2: Hardware implementation and flight results. *Journal of Field Robotics*, *35*(4), 435–458.

Elfes, A. (1987). Sonar-based real-world mapping and navigation. *IEEE Journal on Robotics and Automation*, *3*(3), 249–265.

Fornace, K. M., Drakeley, C. J., William, T., Espino, F., & Cox, J. (2014). Mapping infectious disease landscapes: Unmanned aerial vehicles and epidemiology. *Trends in Parasitology*, *30*(11), 514–519.

Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, *61*, 1258–1276.

Gini, R., Pagliari, D., Passoni, D., Pinto, L., Sona, G., & Dosso, P. (2013). UAV photogrammetry: Block triangulation comparisons. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, XL-1/W2 (pp. 157–162).

Grenzdörffer, G., Engel, A., & Teichert, B. (2008). The photogrammetric potential of low-cost UAVs in forestry and agriculture. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *31*(B3), 1207–1214.

Hovenburg, A. R., de Alcantara Andrade, F. A., Rodin, C. D., Johansen, T. A., & Storvold, R. (2018). Inclusion of horizontal wind maps in path planning optimization of UAS. *2018 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 513–520).

Huang, W. H. (2001). Optimal line-sweep-based decompositions for coverage algorithms. *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)* (Vol. 1, pp. 27–32).

Jennings, A. L., Ordonez, R., & Ceccarelli, N. (2008). Dynamic programming applied to UAV way point path planning in wind. *2008 IEEE*

International Conference on Computer-Aided Control Systems (pp. 215–220).

Klemas, V. V. (2015). Coastal and environmental remote sensing from unmanned aerial vehicles: An overview. *Journal of Coastal Research*, *31*, 1260–1267.

Koubaa, A. (Ed.). (2018). *Robot Operating System (ROS)* (Vol. 778). Springer.

Latombe, J.-C. (2012). *Robot Motion Planning* (Vol. 124). Springer Science & Business Media.

Li, Y., Chen, H., Er, M. J., & Wang, X. (2011). Coverage path planning for {UAVs} based on enhanced exact cellular decomposition method. *Mechatronics*, *21*(5), 876–885.

Oksanen, T., & Visala, A. (2009). Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, *26*(8), 651–668.

Park, S., Deyst, J., & How, J. P. (2004). A new nonlinear guidance logic for trajectory tracking. *AIAA guidance, navigation, and control conference and exhibit* (pp. 16–19).

Paull, L., Thibault, C., Nagaty, A., Seto, M., & Li, H. (2014). Sensor-driven area coverage for an autonomous fixed-wing unmanned aerial vehicle. *IEEE Transactions on Cybernetics*, *44*(9), 1605–1618.

Rew, L. J., & Cousens, R. D. (2001). Spatial distribution of weeds in arable crops: Are current sampling and analytical methods appropriate? *Weed Research*, *41*(1), 1–18.

Richards, A. (2018). Flight optimization for an agricultural unmanned air vehicle. *European Control Conference (ECC)*.

Schopferer, S., & Pfeifer, T. (2015). Performance-aware flight path planning for unmanned aircraft in uniform wind fields. *2015 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 1138–1147).

Seidel, R. (1991). A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Computational Geometry*, *1*(1), 51–64.

Techy, L., & Woolsey, C. A. (2009). Minimum-time path planning for unmanned aerial vehicles in steady uniform winds. *Journal of guidance, control, and dynamics*, *32*(6), 1736–1746.

United States Department of Agriculture. (2016). *Farms and land in farms* (Technical Report).

Whitehead, K., & Hugenholtz, C. H. (2014). Remote sensing of the environment with small unmanned aircraft systems (UASs), part 1: A review of progress and challenges. *Journal of Unmanned Vehicle Systems*, *02*(03), 69–85.

WinterGreen Research Inc. (2016). *Agricultural drones market shares, strategies, and forecasts, worldwide, 2016 to 2022* (Technical Report).

Xu, A., Viriyasuthee, C., & Rekleitis, I. (2011). Optimal complete terrain coverage using an unmanned aerial vehicle. *2011 IEEE International Conference on Robotics and Automation* (pp. 2513–2519).

Zhang, C., & Kovacs, J. M. (2012). The application of small unmanned aerial systems for precision agriculture: A review. *Precision Agriculture*, *13*(6), 693–712.

How to cite this article: Coombes M, Fletcher T, Chen W-H, Liu C. Decomposition-based mission planning for fixed-wing UAVs surveying in wind. *J Field Robotics*. 2020;37:440–465. https://doi.org/10.1002/rob.21928

APPENDIX A

A.1 | Survey image coverage

To show that total image coverage was achieved during the large survey from Figure 27, the MicaSense RedEdge multispectral camera was used to collect GPS tagged aerial images in the red, green, blue, near-inferred, and red edge parts of the spectrum. These images are imported into AgiSoft Photoscan™ to be stitched together into an orthophoto so that total coverage can be demonstrated.

A selection of outputs from AgiSoft is shown in Figure A1. Figure A1a shows the final orthophoto of the field with the seam-lines showing where each image is stitched together. It is clear to see which were the downwind sweeps by the frequency of the images from the seam-lines. This is also clear from Figure A1b where the image capture locations and orientations are displayed over a 3D model of the field. This shows why having an imaging system with a low cycle time (minimum time between images) is important so it can maintain longitudinal overlap at high ground-speeds.

It is clear that despite the high wind, total image coverage has been achieved. As the RedEdge was set to take a photo every second, all parts of the field have >9 images covering the area despite the high image rotations, as seen in Figure A1c.

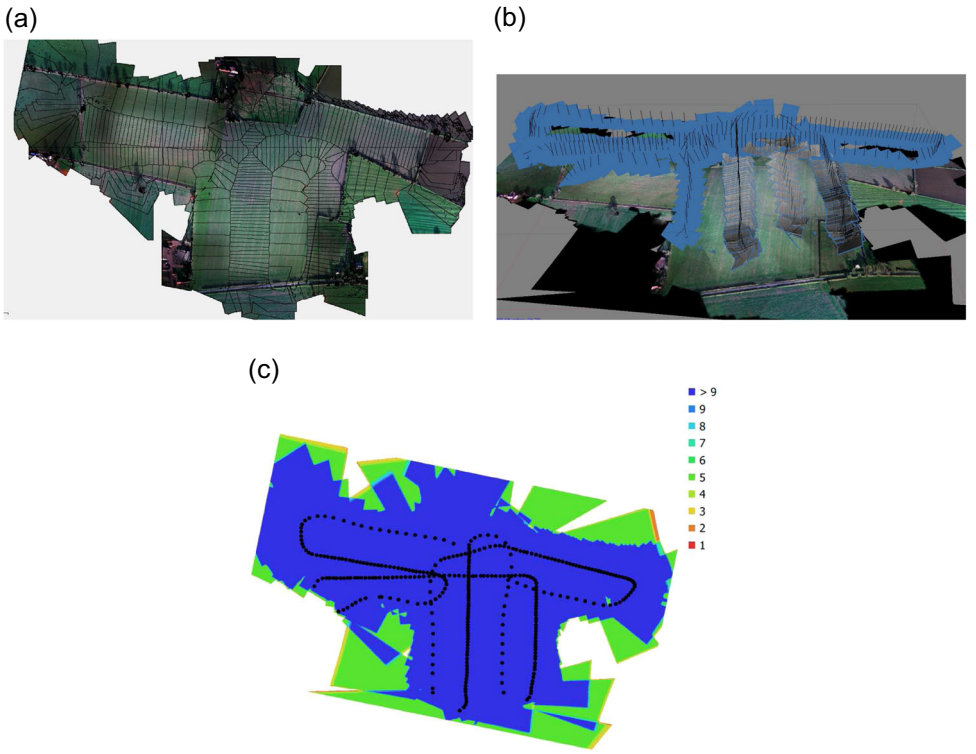


FIGURE A1 Output from AgiSoft Photoscan from processed aerial images, taken during the survey flight from Figure 27 (a) Orthophoto generated showing the seam lines between stitched images. (b) RGB textured 3D model of field, showing the location the image was taken. (c) Image overlap coverage of field. RGB, red, green, blue