NAVIGATION

Multidrone aerial surveys of penguin colonies in Antarctica

Kunal Shah[1]*, Grant Ballard[2], Annie Schmidt[2], Mac Schwager[3]

Speed is essential in wildlife surveys due to the dynamic movement of animals throughout their environment and potentially extreme changes in weather. In this work, we present a multirobot path-planning method for conducting aerial surveys over large areas designed to make the best use of limited flight time. Unlike current survey path-planning solutions based on geometric patterns or integer programs, we solve a series of satisfiability modulo theory instances of increasing complexity. Each instance yields a set of feasible paths at each iteration and recovers the set of shortest paths after sufficient time. We implemented our planning algorithm with a team of drones to conduct multiple photographic aerial wildlife surveys of Cape Crozier, one of the largest Adélie penguin colonies in the world containing more than 300,000 nesting pairs. Over 2 square kilometers was surveyed in about 3 hours. In contrast, previous human-piloted single-drone surveys of the same colony required over 2 days to complete. Our method reduces survey time by limiting redundant travel while also allowing for safe recall of the drones at any time during the survey. Our approach can be applied to other domains, such as wildfire surveys in high-risk weather conditions or disaster response.

INTRODUCTION

We consider the problem of planning paths for a team of aerial robots to conduct a photographic aerial survey of a given region on the ground. The robots must collect a set of images with their downward-facing cameras such that the entire area on the ground is covered. This requires that the images be as high resolution as possible, while ensuring that there is sufficient overlap for the images to be later stitched together to create a high-resolution mosaic of the area. Furthermore, because of weather variability, lighting variability, and dynamic conditions on the ground, the aerial survey should be conducted as quickly as possible, with all robots operating simultaneously, while respecting battery life and altitude limits, as well as respecting wildlife disturbance constraints and avoiding no-fly zones in the airspace. A video overview of the project can be seen in Movie 1.

Movie 1. Video overview of the project.

We present Path Optimization for Population Counting with Overhead Robotic Networks (POPCORN), which plans paths for a team of aerial robots to conduct aerial photographic surveys under these conditions. The core advantages of this algorithm are that it is faster than existing optimization approaches (*1*, *2*) and that it can be halted at any moment (after the first iteration) to return paths that satisfy the problem constraints (in this regard, the algorithm can be considered to be "anytime"). Furthermore, our method is based on the highly expressive language of satisfiability modulo theory (SMT), which allows us to encode requirements such as "start and end locations must coincide," leading to cyclical paths that result in a safe recall property that is absent from current methods like sweep or wavefront coverage planning (*3*, *4*). We demonstrate our method in extensive field surveys of Adélie penguin (*Pygoscelis adeliae*) colonies on Ross Island, Antarctica (77.5294°S, 167.2123°E). Our survey used autonomous, multirotor unpiloted aerial vehicles (UAVs), popularly known as "drones." Our multirobot survey system reduced the time for an aerial survey of a ~300,000-nest Adélie penguin colony spanning over 2 km^2 from 2 days to 3 hours.

We formulate the problem of taking aerial surveys as a constrained path-planning problem. We construct a lattice over the environment such that images taken at each point in the lattice will provide the required resolution and image overlap to produce the desired photomosaic. Our goal is then to find minimum length paths for the UAVs through the lattice ensuring that each point in the lattice is visited at least once, while respecting battery constraints. We also require that each robot enters and exits the lattice at the same point. Our core algorithmic contribution, POPCORN, solves a series of Boolean assignment problems within an SMT framework. Each problem can be solved notably faster than a mixed integer linear program (MILP) of the same size to produce paths that satisfy the constraints (such as a maximum path length bound, or the requirement that the paths be cyclical). At each iteration, the path length bound is decreased until the problem becomes infeasible, at which point the algorithm has found the minimal length paths for the survey. Such constraints are not possible to enforce when using a geometric method such as a sweep or spiral pattern. This algorithm

[1]Department of Mechanical Engineering, Stanford University, Stanford, CA, USA. [2]Point Blue Conservation Science, Petaluma, CA, USA. [3]Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, USA.
*Corresponding author. Email: k2shah@stanford.edu

has the advantage that it is "anytime," meaning that, at any iteration, it provides paths that satisfy the survey requirements; however, these paths can potentially be suboptimal. This is essential when the application requires quick actionable results, as may be the case in extreme outdoor environments. The POPCORN algorithm, together with a variety of supporting utilities, is an available open source as a Python software package named WADL.

Our problem is an example of robotic coverage planning, similar to the algorithms that drive robotic vacuum cleaners to cover the floor of a room. Finding routes that physically cover an area has traditionally been solved by using an underlying geometric pattern, such as a sweep, spiral, or space-filling curve to achieve coverage (*3*–*6*). These techniques, although fast to compute, do not necessarily apply to cases where multiple robots are tasked with simultaneously covering an arbitrarily shaped area. Geometric patterns can overconstrain how the coverage area is partitioned, which can lead to inefficient overall usage of multiple drones (*7*). Furthermore, these geometric methods do not consider collision avoidance constraints or the extra travel needed when the takeoff and landing zones are physically far from the survey area. As a result, there can be a large gap between the start and end points of the survey section, causing the UAV to retraverse the same area and wasting battery life. In our case study, battery life conservation was paramount considering that lithium polymer batteries have reduced capacity in cold weather. This required us to consider the entire trajectory of the UAV from launch to landing to make the most effective use of our limited battery life. By solving for cyclical paths, our method can cover the same area with shorter paths. For example, regarding the penguin colony survey task described in this paper, our method provides a 17.3% average path length reduction when compared with sweep-style paths over the same-sized area.

In contrast to geometric methods, we encode our problem in predicate logic, which allows statements such as "if the robot is at a point in the lattice at time t, it must be at a neighboring lattice point at $t + 1$" and "every point in the lattices must be occupied at least once by at least one robot." These conditions can be converted to formal logical statements and passed to an SMT solver to find a feasible solution for the resulting decision problem. This framework allows us to find paths for multiple UAVs as well as encode other constraints, such as a maximum path length, that are possible within the limits imposed by the limited battery life. We can find shorter paths by iteratively solving the SMT problem and reducing the maximum allowable path length at each step, thus finding the shortest paths once the problem is infeasible. The freedom and flexibility afforded by SMT-based solutions allow our method to automatically produce paths for multiple UAVs to follow simultaneously, without the limitations of a particular geometric pattern.

Shorter aerial survey times allow for more surveys to be completed, capturing more data throughout the season. The Adélie penguin breeding season on Ross Island is about from late October to mid-February (*8*). The images collected during each multirobot survey were stitched together to form a large georeferenced image. These images will be used to estimate the number of nesting adults and chicks produced, as well as the nesting density in different parts of two colonies, Cape Crozier (77.4740°S, 169.1876°E) and Cape Royds (77.5554°S, 166.1580°E), on Ross Island, Antarctica. Ultimately, the data gathered will be used to test several hypotheses about the influence of fine-scale nesting habitats, nest density, and breeding success.

Related work

Our method extends the state of UAV field applications (*9*) by providing an autonomous multirobot solution to UAV survey planning suited for any size of survey area and giving the user freedom to choose the start/end points for each flight and to set a maximum path length. The most common solutions to UAV survey planning problems are sweep-style paths, also known as lawn mower or boustrophedon paths, which route a vehicle back and forth over a rectangular space (*10*–*12*). Other similar space-filling patterns include spiral patterns (*13*, *14*) and Hilbert curves (*15*). There are some multirobot works involving the aerial inspection of three-dimensional (3D) objects (*16*), but they require 3D geometry models and are not immediately suited for 2D photogrammetry applications. Other multirobot methods that are designed to find cyclical paths still can suffer from large amounts of backtracking (*7*) when using sweep-style patterns. Large backtracking also occurs in graph search approaches like the wavefront method (*3*, *17*). Energy-aware methods, also commonly based on geometric primitives (*15*, *18*), only take into account the survey regions, which implicitly assume that the UAV can take off or land near or within the survey area. Methods that focus on minimizing the number of turns have also been used as a proxy for minimum energy methods (*19*, *20*), which are better suited for fixed-wing or fast-moving aircrafts. In contrast, we focus on rotary aircrafts that do not consume that much energy when turning at slower speeds. Even though we do not explicitly solve for it, our paths in Figs. 3 and 8 do contain subpaths of sweep-like motions that are optimally short (*7*, *12*) for some rectangular areas. With many of these methods, the start and end points of these paths tend to be far apart spatially, which becomes a limitation and results in unnecessary backtracking if safe launch and landing locations are physically far from the survey area. This is a common occurrence in many survey tasks.

Ultimately, the raw aerial image data gathered from these surveys are stitched together to form a large area-covering mosaic. This larger image is then used for analysis, e.g., used by ecologists to count fauna and flora (*21*) for population analysis or wildfire risk in forests and grasslands (*22*). Image stitching (*23*) is normally done via a feature detection algorithm (*24*), such as scale-invariant feature transform (*25*), and a bundle adjustment method (*26*, *27*) to align the spatial data. Elevation models can also be produced by recovering the 3D geometric information from the series of photos. These tools have been well studied and can be found in open-source (*28*) and commercial (*29*) products. Although the focus of this work is not the image stitching, our path planning must also consider the overlap requirements for the mosaic process. The size of the area imaged from one photograph taken from the air is a function of the camera's field of view, sensor aspect ratio, and UAV altitude. This image size, in conjunction with the UAV speed and photo interval area, is used to calculate the image overlap. The details are given in (*17*, *30*). Furthermore, different survey tasks, from photogrammetry to generating elevation models, may require different overlap requirements altogether. The final overlap requirement can be represented as a lattice with some spacing in meters. Even though our software package is independent of sensor technology and expects a desired lattice spacing rather than an overlap percent, we developed a simple tool to calculate the resulting overlap from the required image overlap from sensor and flight parameters. This tool is available in our open-source package along with our POPCORN implementation.

Once the lattice spacing and the start/end points for each UAV are identified, we generate a grid world as a graph $G(V, E)$, where

each vertex $v \in V$ represents a point in space over the survey area (*31*). A UAV can travel between two vertices if there is an edge $e \in E$ between them. This graph can be further refined by a user to specify keep-out zones, which are critical in our penguin survey application. The goal of the path-planning algorithm is to return a sequence of vertices for each robot such that the union of all the paths covers the entire area. Although decomposition methods that automatically divide survey areas can improve solution times by solving the problem in smaller steps (*31*, *32*), we manually divided the area (see Fig. 3) because of the fact that we also had to consider the ground observer and terrain constraints. We assume that we have access to the geofence as well as elevation data of the survey area. For situations in which these data are not available, there exists a rich body of literature for informative path planning (*33*–*36*) whereby the robots simultaneously explore their environment and plan paths to increase some information metric. These methods have been applied in plant health monitoring (*37*) and underwater data gathering (*38*).

Solving for a cyclic path through the coverage graph $G(V, E)$ in the single-robot case can be converted to the classic traveling salesman problem (TSP) (*39*–*41*). Because not all graphs admit a TSP solution, we allow for node revisitation in our formulation. Although TSP methods can be extended to allow for node revisitation, the resulting problem is much bigger than the original. A recent single-robot solution (*42*) combats this by combining a sweep planner to plan routes over subregions that are later stitched together via a TSP solver. Because the number of subregions is smaller than the original problem, the solution can still be found in a reasonable amount of time. A number of these formulations can be extended to multiple vehicles, known as the vehicle routing problem (VRP) (*43*, *44*), which are designed for routing vehicles through cities. Methods available for solving the TSP and VRP often resort to heuristic methods for large problems because of their exponential growth with problem size and focus on finding optimal solutions that are constrained to visit every vertex only once. Although our planning problem is inherently NP (nondeterministic polynomial time) complete, we make the distinction that our method focuses more on finding feasible paths that satisfy a maximum path length constraint and allows for vertex revisitation, in contrast to traditional TSP/VRP. Because the maximum path length is dictated by the UAV's battery constraints, we can find solutions tailored to the limitations of the chosen UAV rather than finding a vehicle that would need to meet some endurance requirement. Conceptually similar to TSP and wavefront methods, there are also approaches based on using the minimal spanning tree of a graph for coverage (*45*, *46*); however, they also suffer from large amounts of backtracking because some branches in the spanning tree correspond to long sections that are traversed twice.

We formulate the multirobot coverage problem as an SMT, which is a generalization of Boolean satisfiability (SAT) because it allows us to encode all the specific survey-related constraints. Others (*47*) have provided an in-depth description of SMT versus SAT. SMT allows us to use predicate logic (universal and existential qualifiers) in addition to prepositional logic used in SAT solvers. Although the coverage problem can also be encoded as a MILP (*1*, *2*), we found that, for areas of similar size, MILP methods, even using state-of-the-art solvers, took 4 to 5 days to solve on a powerful workstation. In contrast, our method produced useable solutions for the same survey area in a few hours on a laptop. Although SMT is commonly used for mathematical theorem proving, it has recently been seen in a variety of robotic applications.

Specifically, SAT and SMT have been of interest in the motion planning community (*48*, *49*) because they can model high-level (e.g., task) and low-level (e.g., dynamics) constraints within an expressive framework. In contrast to continuous time path-planning frameworks, we formulate our problem as a discrete path-planning problem on a graph (*1*, *50*), which allows us to easily embed the coverage constraint into the path-planning problem. Similar to the method presented in (*51*, *52*), this work uses one-hot style encoding to translate the multirobot path-planning problem into an SMT. The works (*51*, *52*) formulate an SMT problem to find paths for the robots to move to fixed designated goal positions under collision avoidance constraints. In contrast, we solve for paths that move the robots to completely cover the graph under path length constraints. Both our method and those in (*51*, *52*) use an off-the-shelf SMT solver.

These SMT formulations have been successful in finding feasible paths for navigating obstacles (*53*), expressing reach-avoid problems (*54*). There have also been some implementations with manipulator arms (*55*) and multirobot indoor navigation (*56*). Integrated task and motion planning was explored in (*57*) where the authors used SMT to plan low-level paths through a complex kitchen environment to complete high-level tasks. Last, SAT/SMT-based solvers for a series of multirobot patrolling and periodic routing problems were explored in (*58*, *59*), where the authors used SMT in conjunction with a TSP solver to generate paths for highly coupled multirobot motion planning problems. The flexibly afforded by SMT allows the particular details of the multirobot survey problem to be encoded within a single framework. Although SMT is an NP-complete problem, modern SMT solvers that unitize multiple processors `minismt` (*60*) and `z3` (*61*) have accelerated the solution times for large problems (*62*, *63*). Inspired by the SAT/SMT path-planning methods, we present a multiagent coverage path-planning algorithm designed to compensate for the short battery life of small UAVs by minimizing backtracking when surveying large areas.

We develop a system for multiple UAVs to provide fast, repeatable aerial surveys of large-scale environments in extreme conditions. In this work, we demonstrate the utility of our system to advance population ecology in aerial surveys of Adélie penguin colonies in Antarctica. Current methods for aerial survey planning using commercial tools, such as DroneDeploy (*64*) and DJI mission planner (*65*), allow the user to define a geofence that produces a single, sweep-style path to survey the specified area. POPCORN accepts a geofence along with the desired grid spacing and maximum flight duration or length of the UAV flight. POPCORN will then automatically create a lattice based on the geofence and spacing to produce multiple paths under the maximum flight length. These paths will have notably less backtracking than the corresponding sweep-style path. The interface for POPCORN also allows the user to specify the speed and altitude of the UAV such that resulting paths can be exported to any mission control software such as QGroundControl (*66*) or Universal Ground Control Station (UGCS) (*67*).

Traditionally, colony survey data are gathered by expensive and disruptive helicopter surveys (*68*–*70*). Detailed noise levels of helicopter and UAV surveys are given in (*71*), showing that there is a considerable decrease in wildlife disturbance when using small UAVs over full-scale helicopters. Fixed-wing UAVs (airplanes) have also been used in wildlife applications (*72*), but takeoff and landing considerations, as well as size requirements, make fixed-wing aircrafts infeasible for our remote survey operations. Fixed-wing UAVs may also cause disturbance to wildlife because they can be perceived as

predatory birds (*73*). In contrast, rotary-wing UAV surveys allow for more control over the survey process, as well as higher-resolution images, because UAVs can fly at much lower altitudes without disturbing penguins. In addition, when compared with helicopter surveys, using multiple UAVs in an autonomous system allows for a higher frequency of faster surveys, providing the ability to capitalize on brief windows of favorable weather. There has also been some work on conducting population surveys with ground robots roving through the colony (*74*), but this approach is not suitable for the scale of the colonies that we surveyed, nor practical in the uneven, rocky, and steeply inclined terrain.

Antarctic species have evolved to live in a cold climate, but as the environment changes, understanding how their populations react becomes critical for long-term ecology and conservation. Adélie penguins are important indicators of ecosystem function and change (*75*) in the Southern Ocean. In the face of rapid changes in sea ice and other factors in their pelagic environment, understanding their response to such changes is critical in assessing their ability to adapt to the changing climate and other potential perturbations.

Although nest density of Adélie penguins is known to vary with terrain (*76*, *77*), the effect of nest density itself on nest success is challenging to examine, in part because of difficulties estimating breeding success for large areas and having enough variability of terrain represented to tease apart the independent effects of density and nesting habitat. Understanding the mechanisms of density dependence and the relative influence of nest density versus nesting habitat on breeding success will be critical for assessing this species' ability to persist because warming increasingly leads to habitat change.

Once UAV data collection and the image analysis have been performed, we will estimate breeding success (chicks produced per breeding pair) as well as nest density for two entire Adélie penguin colonies (Cape Royds and Cape Crozier). Combining these data with the high-resolution digital elevation models generated from the UAV imagery will allow us to test the influence of fine-scale variability in nesting habitat and nest density on breeding success at all locations in the colonies. This analysis will be presented in a future publication focused on the ecology findings of the research, whereas this paper is focused on the multi-UAV robotic system developed to conduct the aerial surveys. The Cape Crozier colony is large enough to display substantial variability in nest habitat, density, and success. Thus, having data on the entire colony, as well as data for Cape Royds, will allow us to test hypotheses about spatial variability in habitats that otherwise might require data from numerous colonies.

As the climate warms, previously uninhabited areas in and near penguin colonies are being exposed by retreating glaciers and snowfields (*78*). Whether this area is colonized may depend on the quality of the exposed area relative to the existing habitat. Habitat quality may be quantified, in part, in relation to its ability to mitigate the effects of climate change, because extreme weather may affect some parts of the colony more than others, turning some nesting locations into localized population sinks (*79*–*81*). In addition, fine-scale elevation change, perhaps as little as a few centimeters, can affect the risk of a nest flooding and subsequently failing. Image data gathered from the surveys will be used to better understand the effects of climate change on penguin habitat quality.

RESULTS

Mission outline

We fielded a team of four UAVs executing paths planned with POPCORN to conduct a total of 10 aerial surveys of two Adélie penguin colonies on Ross Island, Antarctica between November 2019 and February 2020. A video overview of the project can be seen in Movie 1. The chief focus of our surveys was the Cape Crozier colony (Fig. 2), the largest colony with more than 300,000 nests spreading over an area of 2 km^2. The Cape Crozier penguin rookery is situated from roughly sea level to 100 m above median sea level (AMSL) on the coast of Ross Island. The Crozier rookery is divided into two sections, the larger West Crozier and the smaller East Crozier, as seen in Fig. 2. Between the rookery and a small five-person field camp, there are rocky ridges (100 to 400 m AMSL), permanent snow, ice fields, and nesting south polar skuas (*Stercorarius maccormick*) (Fig. 4). These geographic constraints, as well as penguins moving about the colony, frequent high winds, and rocky snowy terrain, made it difficult to establish safe takeoff and landing zones within or around the edge of the colony. Furthermore, keeping batteries warm enough for flight (15°C) in the face of subzero conditions (temperatures over the austral summer season are −15° to 0°C) for hours at a time was infeasible in the field. These restrictions required the UAVs to take off at the field camp, enter the colony (2 km from the camp) at key safe points, lower altitude (50 m) above the colony, decelerate to a nondisruptive speed (4 m/s) to take survey photos, and lastly exit the colony near or at the same key safe point to accelerate and fly back to the field camp as seen in Fig. 1.

A separate colony on the other side of Ross Island, Cape Royds, with ~3000 nests (see Fig. 2) was also surveyed. Although these areas are much smaller than the main Crozier colony, similar geographic and logistical constraints were present. We conducted eight surveys of Cape Crozier and two of Cape Royds.

Survey path planning

The two colonies of interest (Crozier and Royds) with their locations on Ross Island are shown in Fig. 2. An individual image with 72.2 m by 55.4 m ground area from the main Crozier area with hundreds of nesting penguins and an expanded view of individual nesting

Fig. 1. Drones at the field camp. (**Left**) UAV safely landed at the Cape Crozier field camp on Ross Island, Antarctica. The dark-colored ridge that the UAVs had to fly over before beginning their surveys can be seen ascending to the right behind the field camp. (**Right**) UAV fleet taking off just outside of the field camp. The UAVs are circled in red.

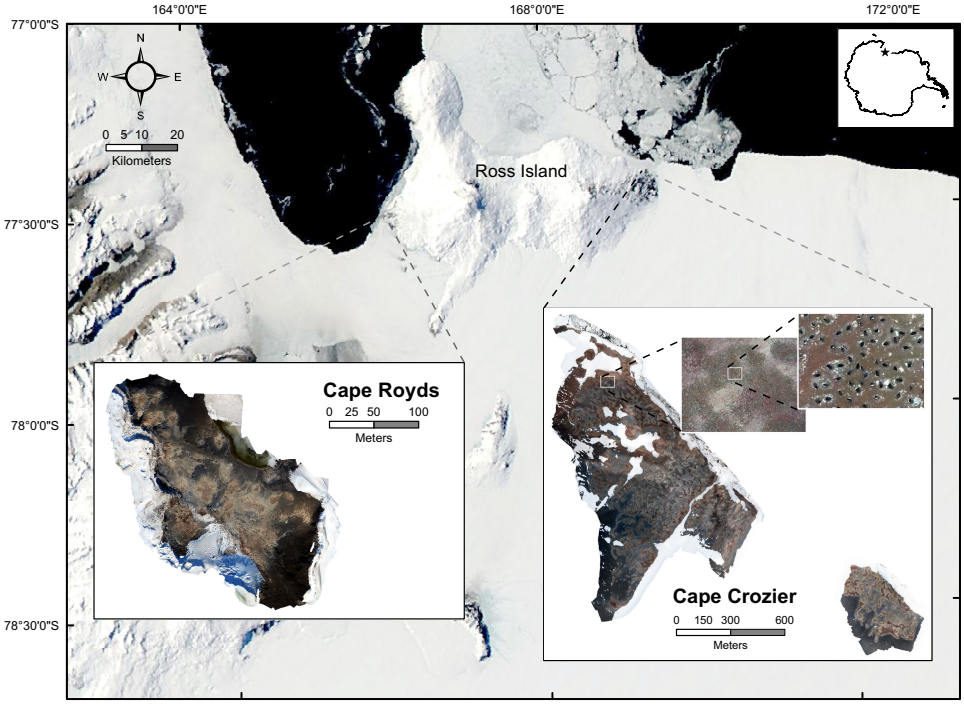

Fig. 2. Survey locations. Cape Royds and Cape Crozier with location (star) on the Antarctic continent (top right). Examples of the full mosaics produced during our survey of Cape Royds (bottom left) and Cape Crozier (bottom right) are shown, along with an individual image from the Crozier survey with hundreds of nesting penguins expanded to show individual nests.

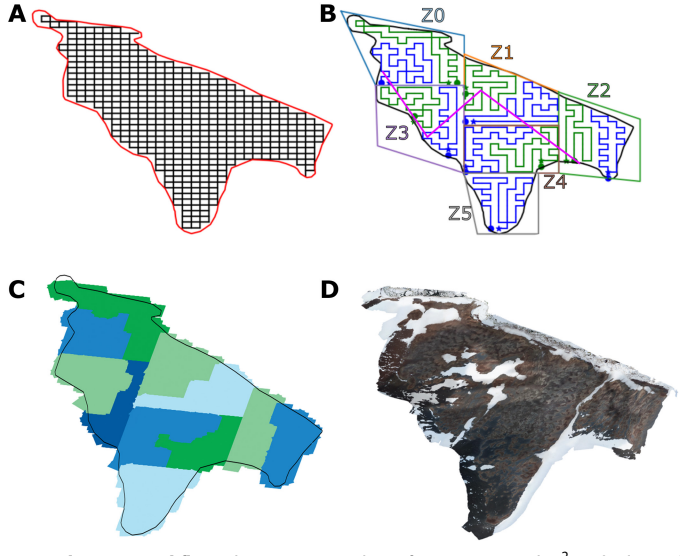

Fig. 3. Planning workflow. The coverage grid (**A**) of West Crozier (2 km^2) with planned UAV paths in blue and green with a human observer path in magenta and six zone locations (Z0 to Z5) (**B**). The resulting photo coverage with individual path coverage in blue and green hues corresponding with (B) (**C**) with the final stitched mosaic (**D**).

penguins can also been seen. More than 2000 images were taken during each survey and stitched together to produce large mosaics that cover the whole colony, like those depicted in Fig. 2. Ecologists will use these mosaics to identify adult and juvenile penguins during the egg incubation and chick brooding periods in an effort to generate population estimates of the colonies. Here, we focus on the engineering advancements rather than the ecology.

Figure 3 shows the West Cape Crozier area with a regular grid with 40-m spacing between vertices. The figure also shows planned paths over the area along with the photo coverage and final stitched image. The figure illustrates the workflow of our system, which takes in the border of a designated geographic area and outputs paths for a given number of UAVs to survey the area. First, the area of interest is specified and overlaid with the coverage grid; Fig. 3A. Next, we select the start and end locations and apply POPCORN to generate the paths offline seen in Fig. 3B. Then, we launch the UAVs that execute these routes to collect the survey images. Simulated photo coverage of each UAV seen as the colored patches is shown in Fig. 3C. Last, we stitch the images together to produce the resulting mosaic image Fig. 3D.

Because of the elevation variation across the colony, we divided West Crozier into six zones (Z0 to Z5), seen as colored polygons in Fig. 3B. This was done to ensure that an observer was able to maintain visual lines of sight to all UAVs from a single point for the entire route within the zone. The observer path is shown in Fig. 3B as a magenta line. An observer needed to be present for compliance with the United States Antarctic Program (USAP) flight mandate for all UAV activity, as well as to monitor for any wildlife disturbance. The interior zones were also placed in such a way that approach paths were not above any nesting penguins. As a result, we could use the majority of the battery life for the survey task without having to risk losing a UAV by flying over sea ice or open water at the north side of the colony boundary. The Cape Crozier colony required 13 flights (11 for West Crozier and 2 for East) per survey. By implementing our multi-UAV survey method, we were able to achieve a 3-hour (2 hours of flight time) survey time for Cape Crozier, in contrast to the 2-day single manual piloted survey done in the previous year (*82*). Cape Royds, on the west side of Ross Island, is home to about 3000 breeding pairs and only required a single 7-min flight per survey.

The paths in Fig. 3 were imported in UGCS (*67*), a flight ground control software that handles the interface between our route planner and the UAV's onboard flight computer. While the paths were generated offline, their cyclical property allowed the user to place the starting node anywhere along the path without recalculating the whole path to account for unforeseen issues that may arise in the field. For example, we encountered a group of skuas along one of the entry paths from the launching area and were able to adjust the start point accordingly in the field without having to replan the entire route.

Figure 4 shows the total path for each route, including the field camp staging area in the lower left corner. The human safety pilots were placed at the yellow dot to maintain line of sight to the UAV so that manual control could be established in case of an emergency. The safety pilots were purely ancillary in our case, watching as the

UAVs executed their path autonomously. Orange no-fly zones mark the area of skua territory and were avoided to minimize disturbance to nesting birds. After the pilots initiated launch, the UAV flight controller executed the routes and autonomously landed back at the field camp to be prepped for the next flight.

Flight time efficiency

A major focus of this research was to reduce the total flight time of the survey; thus, we consider path length as a constraint rather than a consequence of the planner. We found experimentally that, in Antarctic weather conditions (−10° to 0°C and 5 to 10 knots), the battery drained at 0.08 ± 0.01%/s. Factoring the large transit distances (1 to 2 km), the UAVs needed to traverse between the colony and field camp at the speed limit within the colony (4 m/s), resulting in about 10 min of usable survey time per UAV. Our planner was designed to minimize the amount of backtracking while still meeting the coverage requirements. To capture the amount of redundant travel due to our planner, we define the path efficiency of a set of M plans as

$$\eta_{\text{path}} = \frac{\sum_{i=1}^{M} V_{\text{unique}}^{(i)}}{\sum_{i=1}^{M} V_{\text{total}}^{(i)}}.$$

where $V_{\text{unique}}^{(i)} = V_{\text{total}}^{(i)} - V_{\text{repeat}}^{(i)}$ is the number of unique vertices traversed by UAV i and $V_{\text{total}}^{(i)}$ is the total number of vertices in the path for UAV i. In the coverage problem, the number of unique vertices is also the total the number of vertices in the graph $\sum_{i=1}^{M} V_{\text{unique}}^{(i)} = N_{\text{vertex}}$. Likewise, the total number of vertices is just the number of steps taken by all the UAVs in aggregate, $\sum_{i=1}^{M} V_{\text{total}}^{(i)} = N_{\text{path}}$. An efficiency of 1 implies that no UAV needs to backtrack ($V_{\text{repeat}}^{(i)} = 0$ for all i) over any vertex to complete the path. We compared POPCORN with the sweep-style method common in the literature (*7*, *10*–*12*, *21*), as well as a recent cellular decomposition polygon coverage path-planning method (*42*). The sweep routes were planned using the area coverage feature already implemented in the UGCS software (*67*). The polygon coverage method (*42*) first divides the desired area into polygons and then uses a sweep method to plan a coverage path over each polygon. These routes are then stitched together via a TSP solver. The intermediate polygonal division step allows the planner to locally optimize the sweep direction for each polygonal cell, which leads to shorter paths in most cases. Despite the increased efficiency of polygon coverage method over the naïve sweep method, both methods still suffer considerable backtracking because the core planning strategy in both methods is to use a sweep pattern. Performance comparisons between our POPCORN algorithm, the standard sweep method, and the polygon decomposition (*42*) method are given in Fig. 5. Efficiency for all methods is calculated as the ratio between the nonrepeated section and total path length. Table 1 shows the path efficiency for our method, as well as the sweep and polygon methods; the most efficient set of routes is in bold. POPCORN obtains the best efficiency in all cases but one, with an average efficiency improvement of 13.7% over the sweep planner and 9.0% over the polygon coverage method.

Using the battery drain data and the transit distance between the colony and field camp, we can find the flight length limit for a zone, which can be thought of as a "distance budget" because of the predicted remaining battery. Zones that were far from the field camp (Z2, East Crozier) or needed fewer UAVs (Z5, Royds) had a smaller budget than the other zones. If a set of routes exceeds

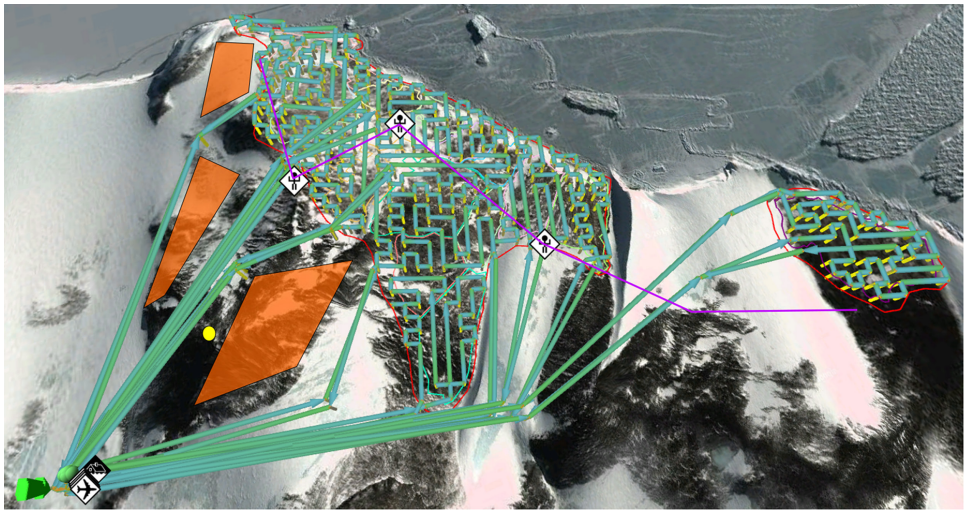

Fig. 4. Survey routes over Cape Crozier. All routes for the Crozier area are shown here. The paths in blue-green were planned by our method and executed with a team of UAVs to conduct aerial surveys of the Cape Crozier and East Rookery penguin colonies. No-fly zones in orange mark the area of mountains or nesting skuas. Takeoff and landing zones at the field camp are marked by the green arrow in the lower left corner. To comply with USAP safety guidelines, the safety pilots were stationed at the yellow dot to maintain line of sight to the UAVs, and the ground observer followed the magenta path.

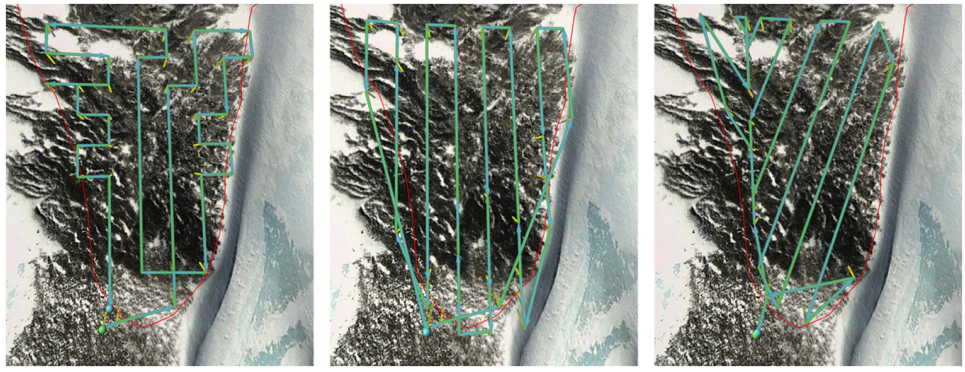

Fig. 5. Comparison of route planning methods. Comparison of our method, POPCORN (left), sweep-style path (middle), and polygon coverage (right) (*42*) over Cape Crozier zone Z5. For this zone, our method produces a 30 and 14% shorter path than the sweep and polygon methods, respectively, when planned from the same start and end points.

Table 1. Route efficiency. Lengths that exceed the distance allowance are underlined. Bold entries indicate the most efficient set of routes.

Area		Croz Z0	Croz Z1	Croz Z2	Croz Z3	Croz Z4	Croz Z5	E. Croz	Royds
Transit		5.8 km	4.9 km	5.6 km	4.3 km	7.0 km	1.9 km	9.6 km	0.2 km
Limit		5.2 km	5.2 km	4.8 km	5.5 km	5.3 km	2.9 km	4.7 km	3.3 km
POPCORN	Length	4.6 km	4.7 km	3.5 km	3.6 km	4.4 km	2.5 km	3.9 km	1.1 km
	N_{vertex}	104	106	80	82	103	56	89	51
	N_{path}	108	110	84	84	104	58	92	55
	η_{path}	**0.967**	**0.964**	**0.952**	**0.976**	**0.990**	**0.955**	**0.967**	**0.927**
Sweep	Length	5.9 km	5.4 km	3.8 km	4.6 km	5.0 km	3.6 km	4.6 km	1.3 km
	Backtrack	1.1 km	1.1 km	0.35 km	0.67 km	0.78 km	0.27 km	0.86 km	0.06 km
	η_{path}	0.821	0.801	0.908	0.926	0.843	0.926	0.813	**0.953**
Polygon (*42*)	Length	5.7 km	4.7 km	4.3 km	4.2 km	4.5 km	2.9 km	5.7 km	1.1 km
	Backtrack	0.83 km	0.47 km	0.51 km	0.43 km	0.63 km	0.30 km	0.46 km	0.12 km
	η_{path}	0.854	0.900	0.881	0.898	0.861	0.896	0.921	0.895

this limit, then at least one UAV will not have enough battery to complete the route and return home safely. This limit and the total transit distance, as well as the aggregate route lengths, are also given (in kilometers) with the length underlined if it exceeds the distance allowance. POPCORN never exceeded this distance allowance, whereas the other planners exceeded the allowance in at least two of eight cases. Each zone specifies a section of the survey where a ground observer can maintain visual line of sight to the UAV over the entire course of the route. The larger zones had a size of just over 100 vertices, which corresponds to about eight New York City blocks. The paths planned with POPCORN shown in Fig. 3 backtracked on four vertices (about 120 m) in the worst case.

Figure 5 shows a side-by-side comparison of the standard sweep style and polygon coverage method against our method. Not accounting for transit portion of the path, the sweep method results in a total path over 1 km longer than the path produced by our method. The polygon method produced a path 400 m longer than the one produced by our method. This 400-m reduction saves 100 s of flight time or 10.4% of the total battery available. Across all instances, POPCORN has an average reduction of 469 m over the polygon method, shortening flights by an average of 2 min or 12.5% of the 16-min battery life.

Although the polygon coverage method (*42*) mostly produced shorter and more efficient routes than the sweep planner, in some cases, it generated nearly degenerate (large aspect ratio) polygons during its decomposition step. The route segments planned in these nearly degenerate polygons resulted in considerable backtracking once stitched together for the final route. POPCORN results in an average reduction of 11.5% in path length when compared with the polygon method. Figure 6 shows the comparison between our method and the same polygon coverage method over all zones in the main Crozier colony. Although the sweep method produces adequate coverage, it still suffers from a large portion of backtracking as shown in Fig. 5. The polygonal coverage (*42*) method reduces this slightly by changing the sweep direction at various points but still suffers from backtracking when stitching together adjacent regions, which can be seen in Fig. 6. Figure 3 shows that, like continuous space

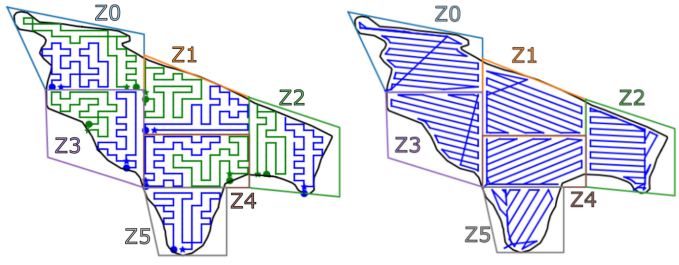

Fig. 6. Comparison of route planning methods. Comparison of our method, POPCORN (left) and the polygon coverage (right) (*42*) over all West Crozier zones (Z0 to Z5). Although the polygon method contours better to the edges of the survey area, it can lead to degenerate polygons resulting in more backtracking once the paths are stitched together. This can be seen in the top left of Z0 and bottom right of Z2. POPCORN results in an average reduction of 11.5% in path length when compared with the polygon method. Our method also automatically splits the paths over multiple UAVs, whereas the polygon methods are for a single UAV.

planners akin to the polygon coverage method (*42*), POPCORN still effectively contours to the boundary of the survey region, despite its discrete formulation.

Robustness to early battery depletion

By optimizing for a looped path, we show that the drone can exit the path at any point and have enough reserve battery to return to the start of the survey and then proceed to the takeoff location in the event the UAV needs to be recalled. The route's start points were set such that the UAVs would start the survey at no lower than 80% battery and exit with at least 30%. Assuming worst-case battery usage (3σ), we can determine a no-return distance for any battery percentage within the survey window. This distance is calculated as the total time remaining at the current percentage times the maximum allowable flight speed, 4 m/s, over the survey area. Figure 7 shows the battery usage data taken from the flight logs of the 11 routes over West Crozier. All routes are designed to exit at or near (one grid point away) the start point. Every route is under the no-return line, indicating that they can safely return to the start of the survey and then proceed home in a way that would not disturb any penguins.

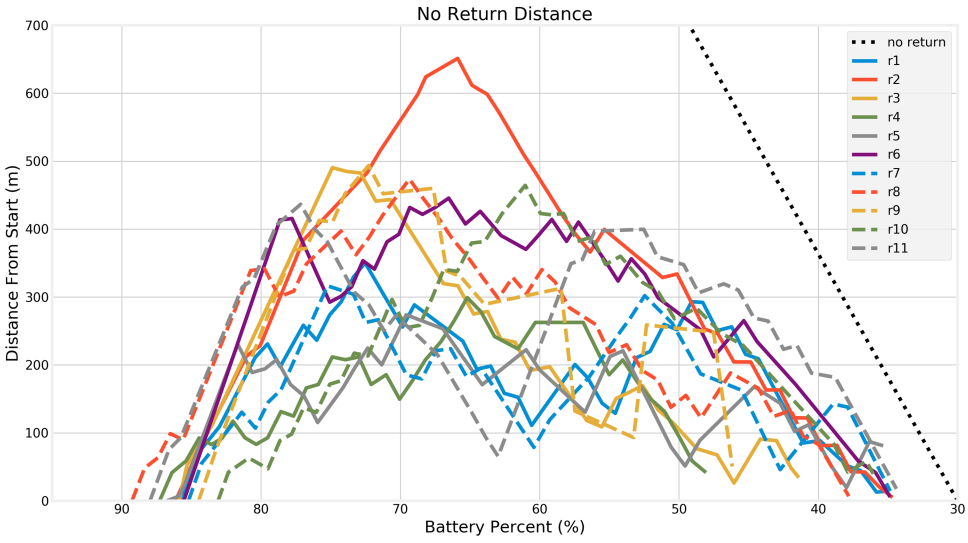

Fig. 7. Point of no-return bounds. No-return bounds with each of the 11 flights over West Crozier. The no-return (dotted) line marks the maximum distance as a function of battery level the UAV can be from the start of the survey and still return home safely.

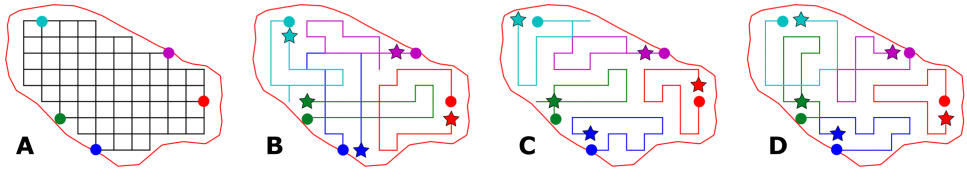

Fig. 8. Example path-planning instance. A set of paths for a multirobot coverage problem with five robots. The initial/final locations are shown on the lattice grid in (**A**), and (**B**) to (**D**) show the set of routes produced with the corresponding maximum path length, T_{max}.

Over the course of the entire season, UAVs were safely recalled in the case of camera errors and telemetry loss or in the case of incoming inclement weather.

CONCLUSION

Our method provides a general and flexible path planning and solution method for a team of aerial UAVs to complete a coverage task. This planning algorithm allows scientists to survey larger areas quickly and robustly using multiple UAVs. In addition, using multiple UAVs in an autonomous system allows for a higher frequency of faster surveys with less overhead and higher resolution when compared with helicopter surveys, thus better capitalizing on small weather windows. The UAVs also do not need to backtrack over the coverage region to return to base because we can enforce cyclical paths. Multirotor UAVs are constrained by their short flight time, so avoiding such backtracking effectively increases the amount of coverage area per flight, as seen in Fig. 5. Furthermore, our algorithm naturally leads to a "safe emergency recall" property for the UAVs. The cyclical path, coupled with a maximum path length constraint, causes the UAVs to fly to farther points in the grid first and work their way backward toward closer points. This allows for safe recall in the case of faster than expected battery drainage or some other hardware fault, given that the UAV will always have an "exit" trajectory available that is as close as possible to the base. This safety property of our planner is highlighted in Fig. 6.

We demonstrated the effectiveness of our algorithm by implementing it on a team of UAVs and conducting field experiments resulting in reduction from 2 days to 3 hours to survey the colony compared with a human-piloted single-drone survey. The added speed in the survey process provides robustness against the rapid changes in weather and allows for more data to be gathered during the breeding season. Other applications requiring quick aerial surveys in extreme conditions—such as surveys of other wildlife populations, disaster sites, or forests and brush lands to detect wildfires in high-risk weather conditions—could also benefit from our algorithm. Our efforts will culminate in a survey planning tool for any number of UAVs that has the capability for online replanning to adapt in real time to changing survey conditions.

MATERIALS AND METHODS

Satisfiability modulo theory

We encode the path-planning problem as an SMT using a one-hot encoding scheme similar to (*51*). Let $X_{i,v,t} \in \{0,1\}$ be a Boolean variable where $i \in \{1, \ldots, N_{\text{robots}}\}$, $v \in V = \{1, \ldots, N_{\text{vertex}}\}$, and $t \in \{0, \ldots, T_{\text{max}}\}$. If $X_{i,v,t} = 1$, it implies that robot i is at vertex $v \in V$ at step t. The parameter T_{max} is the longest path allowed by any robot, e.g., if $T_{\text{max}} = 20$, then robot will traverse at most 20 vertices. Although the value for T_{max} can be different for each robot, e.g., a heterogeneous robot system, for simplicity, we assume that T_{max} is the same for each robot and that there are enough robots to cover the graph. The graph $G(V, E)$ is constructed on the basis of the survey area and the requisite density derived from the image overlap requirements. Various logical statements are set to describe the desired behavior allowed by the robots. The solver attempts to assign each Boolean variable, $X_{i,v,t}$, either 0 (False) or 1 (True) such that these statements, formulated in predicate logic, evaluate to True. These logical statements can be thought of as constraints, similar to how an optimization problem is formulated.

To enforce the physical constraints of the coverage problem, we first require that

$$\forall (i,t)\ \exists\ v\ X_{i,v,t} = 1$$

is only true for only one v. This statement constrains the robot i to only be at one point v at any given step t. Next, we enforce the dynamics, allowing robot i to move to vertex y at step $t + 1$ only if it was at vertex v at step t, such that there is an edge, $(v{\sim}y)$, between the two vertices

$$\neg X_{i,v,t} \bigvee_{v \sim y} X_{i,y,t+1} = 1$$

To promote progress through the graph, we only connect the start/end vertex to itself and all other vertices have no self-loops. Next, we require that

$$\forall v \; \exists (i,t) \; X_{i,v,t} \; = \; 1$$

which enforces the coverage task constraint requiring that, for every vertex v in the graph, there is at least one step t where there is some robot i that occupies that space. If we were to enforce that there exists only one pair (i, t) where this constraint is met, then this would be equivalent to the TSP. Inter-robot collision avoidance constraints can also be added by requiring that there is at most one i such that

$$\forall v, t \; X_{i,v,t} \; = \; 1$$

Last, we encode the start and end points. Let v_i^0 be the start and end vertex for robot i

$$X_{i,v_i^0,0} \; = \; X_{i,v_i^0,T_{\max}} \; = \; 1$$

We desire a closed loop as a path to ensure that the robot does not have to retraverse the same area to return to the launch site. If a closed-loop path is not desired, then separate boundary conditions can be set.

Because the maximum path length is informed by the battery usage of the robot, finding a feasible path implies that the robot will have enough battery to reach the end of the path. In the case of an emergency recall, the robot will have enough battery to safely return home because the Euclidean distance between any point on the path to the end is at most the length of the rest of the path. When the robot cannot safely complete the survey (e.g., inclement weather area or loss of visual), this feature allows the UAV to safely return to the start of the survey path and proceed back to the launch point while obeying any sensitive wildlife or geographic constraints. This becomes more important if the battery consumption rate was underestimated. This feature encourages the UAV to move closer to the start of the path as it reaches the end of the survey, as can be seen in Fig. 7.

Variable reduction

Although these four constraints completely encode the multirobot coverage problem at an SMT, the number of Boolean variables can be reduced by exploiting the problem structure. Given that each robot can only move one adjacent vertex every time step, binary variables that represent unreachable states can be removed from the problem to reduce the solution time. Algorithm 1 describes the forward-backward reduction method.

Table 2. Example cases, numbers of variables, and solution times given before and (after) reduction.

$T_{\max}$	Variables (with reduction)	Solution time (with reduction)
20	7400 (3896)	5.82 s (2.42 s)
19	7030 (3526)	11.39 s (1.32 s)
18	6660 (3156)	18.24 s (9.48 s)
17	6290 (2786)	12.6 s (5.22 s)

Algorithm 1. Forward-backward reduction

for each robot i do:
  for each vertex v do:
    $d \leftarrow$ Distance(v, v_i^0)
    for $t \in \{0,\ldots,d-1\}$ do:
      Remove$(X_{i,v,t})$
      Remove$(X_{i,v,T_{\max}-t})$

The `Distance()` function returns number of time steps needed to travel between two vertices. In a rectilinear grid world, this `Distance()` function can be thought of as the $L1$, or Manhattan distance. The `Remove()` function simply removes the Boolean variable by constraining it to be False before the solver assigns the values.

Sequential SMT

The SMT instances do not optimize an objective function directly and instead only find an anytime feasible solution. To transform a feasibility framework into an optimization one, we use the standard method of descending (or ascending) on an objective function until the problem is infeasible. To find the shortest path, Algorithm 2 performs this iterative process by repeatedly solving SMT instances where $T_{\max}$ is successively reduced each step according to some search schedule until the user terminates or the problem is infeasible. From the construction of the problem, each successive iteration guarantees that the resulting paths are not longer than the previous iterate, assuming that a solution is found. As an example, we provide a linear search schedule in for $T_{\max}$ in Algorithm 2, but, in practice, any reasonable search schedule is applicable. For the `Solve()` method, we use the freely available `z3` solver (*61*).

Algorithm 2. Sequential SMT

SMT $\leftarrow$ *Init*(*problemData*)
  sol $\leftarrow$ *null*
  while *True* do:
    if *Solve*(*SMT*, $T_{\max}$) is feasible:
      sol $\leftarrow$ *getSol*(*SMT*)
      $T_{\max} \leftarrow T_{\max} - 1$
    else:
      Return *sol*

Example instance

Consider five UAVs tasked with surveying an arbitrarily shaped area. The coverage requirement is such that the coverage graph has 74 vertices, and the robots enter through a node at the periphery of the survey area. As $T_{\max}$ is reduced, the paths are constrained to be shorter; thus, the paths become more spatially efficient. Although each problem instance has initially around 6500 Boolean variables, after applying Algorithm 1, about 3500 variables are removed before the problem is passed to the solver. Table 2 shows that the effect Algorithm 1 has on solution times, and Fig. 8 provides the paths for this example instance. The paths start at the circle symbol and end at the star symbol, which is one step away from the start vertex.

SMT solution times do not always decrease with the number of variables. However, Algorithm 1 removes variables that represent impossible states for the robots; thus, the solver will explore less branches when searching for a feasible assignment.

SUPPLEMENTARY MATERIALS

robotics.sciencemag.org/cgi/content/full/5/47/eabc3000/DC1
Section S1. Survey computation details

Section S2. Field survey protocol and materials
Table S1. Survey planning solution times.

REFERENCES AND NOTES

1. J. Yu, S. M. LaValle, Planning optimal paths for multiple robots on graphs. in *IEEE International Conference on Robotics and Automation* (IEEE, 2013), pp. 3612–3617.
2. A. Nedjati, I. Gokhan, B. Vizvari, J. Arkat, Complete coverage path planning for a multi-UAV response system in post-earthquake assessment. *Robotics* **5**, 26 (2016).
3. A. Barrientos, J. Colorado, J. D. Cerro, A. Martinez, C. Rossi, D. Sanz, J. Valente, Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *J. Field Robot.* **28**, 667–689 (2011).
4. H. Choset, Coverage for robotics–a survey of recent results. *Ann. Math. Artif. Intell.* **31**, 113–126 (2001).
5. E. Galceran, M. Carreras, A survey on coverage path planning for robotics. *Robot. Auton. Syst.* **61**, 1258–1276 (2013).
6. T. Cabreira, L. Brisolara, P. R. Ferreira Jr., Survey on Coverage Path Planning with Unmanned Aerial Vehicles. *Drones* **3**, 4 (2019).
7. G. S. C. Avellar, G. A. S. Pereira, L. C. Pimenta, P. Iscold, Multi-UAV routing for area coverage and remote sensing with minimum time. *Sensors* **15**, 27783–27803 (2015).
8. D. Ainley, R. E. LeResche, W. J. L. Sladen, *Breeding Biology of the Adélie Penguin* (University of California Press, 1983).
9. A. Otto, N. Agatz, J. Campbell, B. Golden, E. Pesch, Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks* **72**, 411–458 (2018).
10. H. Choset, P. Pignon, Coverage path planning: The boustrophedon cellular decomposition, in *Field and Service Robotics* (Springer, 1998), pp. 203–209.
11. W. H. Huang, Optimal line-sweep-based decompositions for coverage algorithms, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2001), pp. 27–32.
12. I. Maza, A. Ollero, Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms, in *Distributed Autonomous Robotic Systems* (Springer, 2007), pp. 221–230.
13. T. M. Cabreira, C. D. Franco, P. R. Ferreira, G. C. Buttazzo, Energy-aware spiral coverage path planning for UAV photogrammetric applications. *IEEE Robot. Autom. Lett.* **3**, 3662–3668 (2018).
14. S.-W. Moon, D. H.-C. Shim, Study on path planning algorithms for unmanned agricultural helicopters in complex environment. *Int. J. Aeronaut. Space Sci.* **10**, 1–11 (2009).
15. O. Artemenko, O. J. Dominic, O. Andryeyev, A. Mitschele-Thiel, Energy-aware trajectory planning for the localization of mobile devices using an unmanned aerial vehicle, in *25th International Conference on Computer Communication and Networks (ICCCN)* (IEEE, 2016), pp. 1–9.
16. S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, G. Nikolakopoulos, Cooperative coverage path planning for visual inspection. *Control. Eng. Pract.* **74**, 118–131 (2018).
17. L. Nam, L. Huang, X. J. Li, J. Xu, An approach for coverage path planning for UAVs, in *2016 IEEE 14th International Workshop on Advanced Motion Control (AMC)* (IEEE, 2016), pp. 411–416.
18. T. M. Cabreira, P. R. Ferreira, C. D. Franco, G. C. Buttazzo, Grid-based coverage path planning with minimum energy over irregular-shaped areas with UAVs, in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)* (IEEE, 2019), pp. 758–767.
19. Y. Li, H. Chen, M. J. Er, X. Wang, Coverage path planning for UAVs based on enhanced exact cellular decomposition method. *Mechatronics* **21**, 876–885 (2011).
20. J. Valente, J. Del Cerro, A. Barrientos, D. Sanz, Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach. *Comput. Electron. Agric.* **99**, 153–159 (2013).
21. A. Hodgson, D. Peel, N. Kelly, Unmanned aerial vehicles for surveying marine fauna: assessing detection probability. *Ecol. Appl.* **27**, 1253–1267 (2017).
22. C. Phan, H. H. T. Liu, A cooperative UAV/UGV platform for wildfire detection and fighting, in *7th International Conference on System Simulation and Scientific Computing* (IEEE, 2008), pp. 494–498.
23. M. Brown, D. G. Lowe, Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vis.* **74**, 59–73 (2007).
24. E. Adel, M. Elmogy, H. El-Bakry, Image stitching based on feature extraction techniques: A survey. *Int. J. Comput. Appl.* **99**, 1–8 (2014).
25. D. G. Lowe, Object recognition from local scale-invariant features, in *Proceedings of the Seventh IEEE International Conference on Computer Vision* (IEEE, 1999), pp. 1150–1157.
26. R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision* (Cambridge University Press, ed. 2, 2004).
27. P. F. McLauchlan, A. Jaenicke, Image mosaicing using sequential bundle adjustment. *Image Vis. Comput.* **20**, 751–759 (2002).
28. Microsoft, Image composite editor; www.microsoft.com/en-us/research/project/image-composite-editor.
29. Agisoft, Metashape; www.agisoft.com.
30. C. Di Franco, G. Buttazzo, Energy-aware coverage path planning of UAVs, in *Proceedings of the 2015 IEEE International Conference on Autonomous Robot Systems and Competitions* (IEEE, 2015), pp. 111–117.
31. E. U. Acar, H. Choset, Sensor-based coverage of unknown environments: Incremental construction of morse decompositions. *Inte. J. Robot. Res.* **21**, 345–366 (2002).
32. E. U. Acar, H. Choset, J. Y. Lee, Sensor-based coverage with extended range detectors. *IEEE Trans. Robot.* **22**, 189–198 (2006).
33. M. Popović, T. Vidal-Calleja, G. Hitz, J. Chung, I. Sa, R. Siegwart, J. Nieto, An informative path planning framework for UAV-based terrain monitoring. *Auton. Robots* **44**, 1–23 (2020).
34. A. Singh, A. Krause, W. J. Kaiser. Nonmyopic adaptive informative path planning for multiple robots, in *Twenty-First International Joint Conference on Artificial Intelligence* (AAAI, 2009), pp. 1843–1850.
35. D. E. Soltero, M. Schwager, D. Rus, Generating informative paths for persistent sensing in unknown environments, in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*) (IEEE, 2012), pp. 2172–2179.
36. X. Lan, M. Schwager, Rapidly exploring random cycles: Persistent estimation of spatiotemporal fields with multiple sensing robots. *IEEE Trans. Robot.* **32**, 1230–1244 (2016).
37. K. Vivaldini, V. Guizilini, M. Oliveira, T. H. Martinelli, D. F. Wolf, F. Ramos, Route planning for active classification with UAVs, in *2016 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016), pp. 2563–2568.
38. J. Binney, A. Krause, G. S. Sukhatme, Informative path planning for an autonomous underwater vehicle in *2010 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2010), pp. 4791–4796.
39. M. Held, R. M. Karp, The traveling-salesman problem and minimum spanning trees. *Oper. Res.* **18**, 1138–1162 (1970).
40. M. Held, R. M. Karp, The traveling-salesman problem and minimum spanning trees: Part ii. *Math. Program.* **1**, 6–25 (1971).
41. G. Gutin, A. Yeo, The greedy algorithm for the symmetric tsp. *Algorithmic Oper. Res.* **2**, 97–99 (2007).
42. R. Bähnemann, N. Lawrance, J. Chung, M. Pantic, R. Siegwart, J. Nieto. Revisiting boustrophedon coverage path planning as a generalized traveling salesman problem. arXiv:1907.09224 [cs.RO] (22 July 2019).
43. T. Bektas, The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega* **34**, 209–219 (2006).
44. P. Toth, D. Vigo, *The Vehicle Routing Problem* (SIAM, 2002).
45. Y. Gabriely, E. Rimon, Spanning-tree based coverage of continuous areas by a mobile robot. *Ann. Math. Artif. Intell.* **31**, 77–98 (2001).
46. N. Agmon, N. Hazon, G. A. Kaminka, Constructing spanning trees for efficient multi-robot coverage, in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), pp. 1698–1703
47. D. Yurichev, Sat/smt by example (2019).
48. E. Scala, M. Ramirez, P. Haslum, S. Thiebaux, Numeric planning with disjunctive global constraints via SMT, in *26th International Conference on Automated Planning and Scheduling* (AAAI, 2016).
49. E. I. Grøtli, T. A. Johansen, Path planning for UAVs under communication constraints using splat! and MILP. *J. Intell. Robot. Syst.* **65**, 265–282 (2012).
50. D. Silver, Cooperative pathfinding, in *2005 Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (AAAI, 2005), pp. 117–122.
51. P. Surynek, Simple direct propositional encoding of cooperative path finding simplified yet more, in *Mexican International Conference on Artificial Intelligence* (Springer, 2014), pp. 410–425.
52. P. Surynek, Reduced time-expansion graphs and goal decomposition for solving cooperative path finding sub-optimally, in *Twenty-Fourth International Joint Conference on Artificial Intelligence* (AAAI, 2015), pp. 1916–1922.
53. W. N. N. Hung, X. Song, J. Tan, X. Li, J. Zhang, R. Wang, P. Gao, Motion planning with satisfiability modulo theories, in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), pp. 113–118.
54. Y. Shoukry, P. Nuzzo, I. Saha, A. L. Sangiovanni-Vincentelli, S. A. Seshia, G. J. Pappas, P. Tabuada, Scalable lazy SMT-based motion planning, in *2016 IEEE 55th Conference on Decision and Control (CDC)* (IEEE, 2016), pp. 6683–6688.
55. N. T. Dantam, Z. K. Kingston, S. Chaudhuri, L. E. Kavraki, Incremental task and motion planning: A constraint-based approach, in *Robotics: Science and Systems* (2016), vol. 12, p. 00052.
56. I. Saha, R. Ramaithitima, V. Kumar, G. J. Pappas, S. A. Seshia, Automated composition of motion primitives for multi-robot systems from safe LTL specifications, in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2014), pp. 1525–1532.
57. S. Nedunuri, S. Prabhu, M. Moll, S. Chaudhuri and L. E. Kavraki, SMT-based synthesis of integrated task and motion plans from plan outlines, in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), pp. 655–662.

58. F. Imeson and S. L. Smith, Multi-robot task planning and sequencing using the SAT-TSP language, in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), pp. 5397–5402.
59. F. Imeson, S. L. Smith, An SMT-Based Approach to Motion Planning for Multiple Robots With Complex Constraints. *IEEE Trans. Robot.* **35**, 669–684 (2019).
60. H. Zankl, A. Middeldorp, Satisfiability of non-linear (IR) rational arithmetic, in *International Conference on Logic for Programming Artificial Intelligence and Reasoning* (Springer, 2010), pp. 481–500.
61. L. De Moura, N. Bjørner, Z3: An efficient SMT solver, in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (Springer, 2008), pp. 337–340.
62. Y. Wang, N. T. Dantam, S. Chaudhuri, L. E. Kavraki, Task and motion policy synthesis as liveness games, in *26th International Conference on Automated Planning and Scheduling* (AAAI, 2016), pp. 536–540.
63. N. T. Dantam, S. Chaudhuri, L. E. Kavraki, The task-motion kit: An open source, general purpose task and motion-planning framework. *IEEE Robot. Autom. Mag.* **25**, 61–70 (2018).
64. Drone Deploy, Drone software for industry innovators; www.dronedeploy.com.
65. DJI, DJI Ground Station Pro; www.dji.com/ground-station-pro.
66. Dronecode, QGroundControl; qgroundcontrol.com.
67. SPH Engineering, Universal Ground Control Station (UGCS); www.ugcs.com.
68. A. Borowicz, P. McDowall, C. Youngflesh, T. Sayre-McCord, G. Clucas, R. Herman, S. Forrest, M. Rider, M. Schwaller, T. Hart, S. Jenouvrier, M. J. Polito, H. Singh, H. J. Lynch, Multi-modal survey of Adélie penguin mega-colonies reveals the Danger Islands as a seabird hotspot. *Sci. Rep.* **8**, 3926 (2018).
69. N. Ratcliffe, D. Guihen, J. Robst, S. Crofts, A. Stanworth, P. Enderlein, A protocol for the aerial survey of penguin colonies using UAVs. *J. Unmanned Veh. Syst.* **3**, 95–101 (2015).
70. M. E. Goebel, W. Perryman, J. Hinke, D. Krause, N. Hann, S. Gardner, D. LeRoi, A small unmanned aerial system for estimating abundance and size of Antarctic predators. *Polar Biol.* **38**, 619–630 (2015).
71. K. Christie, S. L. Gilbert, C. L. Brown, M. Hatfield, L. Hanson, Unmanned aircraft systems in wildlife research: current and future applications of a transformative technology. *Front. Ecol. Environ.* **14**, 241–251 (2016).
72. J. Linchant, J. Lisein, J. Semeki, P. Lejeune, C. Vermeulen, Are unmanned aircraft systems (UAS s) the future of wildlife monitoring? A review of accomplishments and challenges. *Mammal Rev.* **45**, 239–252 (2015).
73. J. F. McEvoy, G. P. Hall, P. G. McDonald, Evaluation of unmanned aerial vehicle shape, flight path and camera type for waterfowl surveys: disturbance effects and species recognition. *PeerJ* **4**, e1831 (2016).
74. Y. L. Maho, J. D. Whittington, N. Hanuise, L. Pereira, M. Boureau, M. Brucker, N. Chatelain, J. Courtecuisse, F. Crenner, B. Friess, E. Grosbellet, L. Kernaléguen, F. Olivier, C. Saraux, N. Vetter, V. A. Viblanc, B. Thierry, P. Tremblay, R. Groscolas, C. L. Bohec, Rovers minimize human disturbance in research on wild animals. *Nat. Methods* **11**, 1242–1244 (2014).
75. D. Ainley, *The Adélie Penguin: Bellwether of Climate Change* (Columbia Univ. Press, 2002).
76. M. A. LaRue, H. J. Lynch, P. O. B. Lyver, K. Barton, D. G. Ainley, A. Pollard, W. R. Fraser, G. Ballard, A method for estimating colony sizes of Adélie penguins using remote sensing imagery. *Polar Biol.* **37**, 507–517 (2014).
77. R. L. Penney, Territorial and social behavior in the Adélie penguin. *Antarctic Bird Stud.* **12**, 83–131 (1968).
78. M. LaRue, D. Ainley, M. Swanson, K. Dugger, O. Phil, B. Lyver, K. Barton, G. Ballard, Climate change winners: receding ice fields facilitate colony expansion and altered dynamics in an Adélie penguin metapopulation. *PLOS ONE* **8**, e60568 (2013).
79. P. K. Bricher, A. Lucieer, E. J. Woehler, Population trends of Adélie penguin (Pygoscelis adeliae) breeding colonies: A spatial analysis of the effects of snow accumulation and human activities. *Polar Biol.* **31**, 1397–1407 (2008).
80. M. Ferrer, J. Belliure, E. Minguez, E. Casado, K. Bildstein, Heat loss and site-dependent fecundity in chinstrap penguins (Pygoscelis antarctica). *Polar Biol.* **37**, 1031–1039 (2014).
81. J. Moreno, A. P. Møller, Extreme climatic events in relation to global change and their impact on life histories. *Curr. Zool.* **57**, 375–390 (2012).
82. UNAVCO, A less invasive, more accurate way to survey penguin colonies; www.unavco.org/highlights/2019/penguins.html.

Acknowledgments: We thank D. Ainley, K. Dugger, M. Elrod, D. Jongsomjit, P. Levinson, A. Lescroël, and V. Morandini for assistance with the field operations. **Funding:** This work was supported by NSF grant 1834986 with logistical support provided by the USAP. G.B. received support from NSF/OPP-1543498. **Author contributions:** The algorithm was designed by K.S. and M.S. Field operations were conducted by K.S., G.B., and A.S. The paper was written by K.S., G.B., A.S., and M.S. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** An open-source version of the codebase for this project can be found at github.com/k2shah/wadl/, and the version used in this work can be found at github.com/k2shah/wadl/releases/tag/0.0.1.

Submitted 2 May 2020
Accepted 23 September 2020
Published 28 October 2020
10.1126/scirobotics.abc3000

Citation: K. Shah, G. Ballard, A. Schmidt, M. Schwager, Multidrone aerial surveys of penguin colonies in Antarctica. *Sci. Robot.* **5**, eabc3000 (2020).

Science Robotics

Multidrone aerial surveys of penguin colonies in Antarctica

Kunal Shah, Grant Ballard, Annie Schmidt, and Mac Schwager

Sci. Robot. **5** (47), eabc3000. DOI: 10.1126/scirobotics.abc3000

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.