Using a UAV for Destructive Surveys of Mosquito Population

An Nguyen, Dominik Krupke, Mary Burbage, Shriya Bhatnagar, Sándor P. Fekete, and Aaron T. Becker

***Abstract*— This paper introduces techniques for mosquito population surveys in the field using electrified screens (bug zappers) mounted to a UAV. Instrumentation on the UAV logs the UAV path and the GPS location, altitude, and time of each mosquito elimination. Hardware experiments with a UAV equipped with an electrified screen provide real-time measurements of (former) mosquito locations and mosquito-free volumes. Planning a trajectory for the UAV that maximizes the number of mosquito kills is related to the Traveling Salesman Problem, the Lawn Mower Problem and, most closely, Milling with Turn Cost. We reduce this problem to considering variants of covering a grid graph with minimum turn cost, corresponding to optimized energy consumption. We describe an exact method based on Integer Programming that is able to compute provably optimal instances with over 1,500 pixels. These solutions are then implemented on the UAV.**

I. Introduction

Mosquito-borne diseases kill millions of humans each year [1]. Because of this threat, governments worldwide track mosquito populations. Tracking individual mosquitoes is difficult because of their small size, wide-ranging flight, and preference for low-light. Tracking studies of individual mosquitos have chosen to use small ($1.2\,\mathrm{m} \times 2.4\,\mathrm{m}$) indoor regions [2], or mating swarms backlit against a solid background [3].

The dominant tools for tracking mosquito populations are stationary traps that are checked at weekly intervals (*e.g.* Encephalitis Vector Surveillance traps and/or gravid traps [4]). Recent research has focused on making these traps smaller, cheaper, and capable of providing real-time data [5], [6]; however, they still rely on attracting mosquitoes to the trap. This paper presents an alternate solution using an electrified bug-zapping screen mounted on an unmanned aerial vehicle (UAV) as shown in Fig. 1 to seek out the mosquitoes in their habitat. As the UAV follows a path, it sweeps out a volume of air, temporarily removing all the mosquitoes in this volume. By monitoring the voltage across this screen, we can track individual mosquito contacts. UAVs have strict energy budgets, so optimized flight patterns are of crucial importance. As a consequence, putting the UAV to good use requires methods for computing trajectories that minimize energy consumption along the way, but maximize the total volume of mosquitoes at visited locations.

This paper is arranged as follows. After a review of related work in §II, we describe a design and rationale for a UAV with bug zapper in §III. We next present a path planning optimization strategy in §IV. We then describe hardware experiments with the UAV in §V and conclude with directions for future research in §VI.

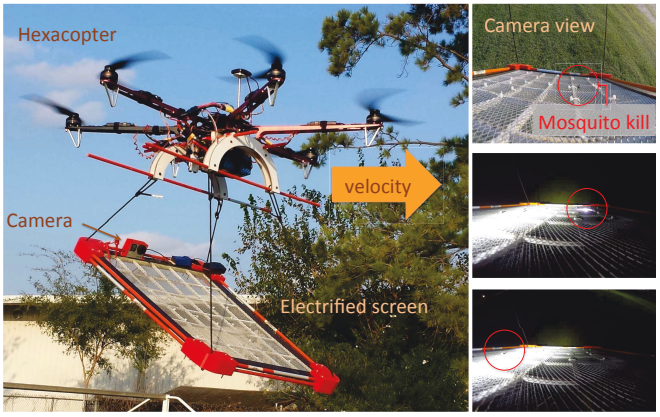

Fig. 1. A hexacopter UAV carrying a $48\,\mathrm{cm} \times 61\,\mathrm{cm}$ rectangular bug-zapping screen. An onboard micro controller monitors the voltage across the screen and records the time, GPS location, humidity, and altitude for each mosquito strike. At right are three frames recorded by the onboard camera showing mosquito hits, during the day (top) and at twilight. See attachment for videos of flight experiments [7].

II. Related Work

Robotic Coverage: Robotic coverage has a long history. The basic problem is one of designing a path for a robot that ensures the robot visits within r distance of every point on the workspace. For an overview see [8]. This work has been extended to use multiple coverage robots in a variety of ways, including using simple behaviors for the robots [9], [10].

Mosquito Control Solutions: Mosquito control also has a long history of efforts associated both with monitoring mosquito populations [11] and with eliminating mosquitoes. The work involves both draining potential breeding grounds and destroying living mosquitoes [12]. An array of insecticidal compounds has been used with different application methods, concentrations, and quantities, including both larvicides and compounds directed at adult mosquitoes [13].

Various traps have been designed to capture and/or kill mosquitoes with increasing sophistication in imitating human bait, as designers strive to achieve a trap that can rival the attraction of a live human [14]. In recent history, methods have also included genetically modifying mosquitoes so that they either cannot reproduce effectively or cannot transmit diseases successfully [15], and with the recent genomic mapping of mosquito species, new ideas for more targeted work have been formulated [16].

Popular methods to control mosquitoes such as insecticides are effective, but they have the potential to in-

*This work was supported by the National Science Foundation under Grant No. [IIS-1553063] and [IIS-1646607].

A. Nguyen, M. Burbage, S. Bhatnagar, and A. Becker are with the ECE Department at the University of Houston, TX. an.nguyen.vn@ieee.org, atbecker@uh.edu. S. Fekete and D. Krupke are with the Dept. of Computer Science, TU Braunschweig, Germany, s.fekete@tu-bs.de, d.krupke@tu-bs.de

troduce long-term environmental damage and mosquitoes have demonstrated the ability to become resistant to pesticides [17]. Traditional electrified screens (bug zappers) use UV light to attract pests but have a large bycatch of non-pest insects [18].

Robotic Pest Management: As GPS technology has flourished and data processing has become cheaper and more readily available, researchers have explored options for implementing the new technologies in breeding ground removal [19] and more effective insecticide dispersion [20]. Low-cost UAVs for residential spraying are under development [21]. Even optical solutions have been considered, including laser containment [22] or, by extension, exclusion and laser tracking and extermination [23].

III. Hardware Design

This section examines the components of the mosquito UAV system, shown in Fig. 1. This includes the UAV, electrified screen, surveying electronics, and a discussion of the energy budget.

A. UAV

The UAV is a custom-built, $177\,\mathrm{cm}$ wingspan hexacopter, controlled by a Pixhawk flight controller running ArduPilot Mega flight software. The UAV has a 3DR GPS module using the UBlox NEO-7 chipset.

B. Screen Design

The mosquito screen is designed to eliminate high density mosquito populations. This screen was constructed from two expanded aluminum mesh panels, spaced apart by $3\,\mathrm{mm}$ thick ABS grid. These mesh panels have $12\,\mathrm{mm}$ diamond-shaped openings, and is held taught by nylon bolts around the perimeter. The bottom mesh panel is offset by half a diamond (6 mm) to the right to ensure all insects greater than 6 mm cannot pass through the net. The top mesh is held at the reference voltage and the bottom mesh is energized to $1.8\ kV$ above the reference voltage.

The perimeter is reinforced by two sets of $7\,\mathrm{mm}$ diameter fiberglass rods that are inset into 3D printed corner fixtures. These rods protect the frame from getting damaged from any side, and allows the UAV to land without damaging the net.

Once assembled, the net weighs $0.948\,\mathrm{kg}$ and has an overall area of $0.194\,\mathrm{m}^2$, with the spacer occupying $0.0325\,\mathrm{m}^2$. This makes the effective net area $0.161\,\mathrm{m}^2$.

C. Screen Location

The UAV carries the bug-zapping screen, which is suspended by paracord rope at each corner. The location of this screen determines the efficacy of the mosquito UAV, measured in mosquitoes detected per second of flight time. The following describes a simplified analysis to optimize the screen location.

For manufacturing ease, the electrified screen is a rectangle with a width of d_s. The screen is suspended a distance h_s beneath the UAV flying at height h_d. We chose to suspend the screen beneath the UAV to avoid the weight of the rigid frame that would be required if the screen were above the UAV and because most mosquito species prefer low flight [24]. This screen can be suspended at any desired angle θ in comparison to horizontal, as shown in Fig. 2. Two key parameters are the distance h_s and the optimal angle θ. The goal is to clear the greatest volume of mosquitoes per second, a volume defined by the UAV forward velocity v_f and the cross-sectional area $h_m \times d_s$ cleared by the screen, as shown in Fig. 3.

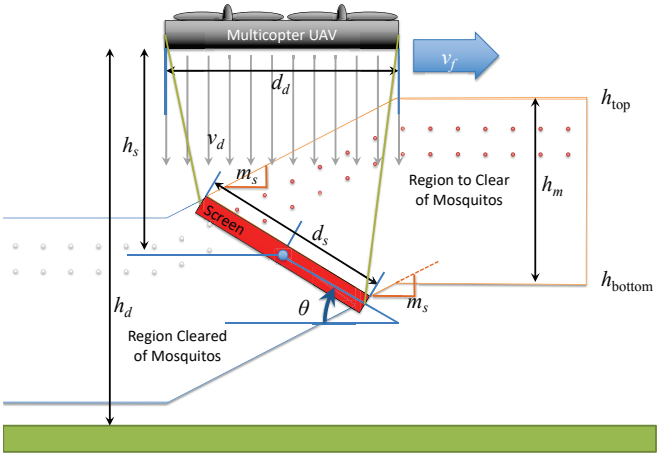

Fig. 2. The UAV suspends a rectangular bug-zapping screen beneath it. Propwash pushes incoming mosquitoes downwards, and the UAV clears a volume $h_m \times d_s \times v_f$ each second. Circles show two mosquitoes at equal time intervals relative to the UAV.

To hover, the UAV must push sufficient air down with velocity v_d to apply a force that cancels the pull of gravity. The UAV and screen combined have mass m_d and its cross section can be approximated as a square with a side length of d_d. The mass flow of air through the UAV's propellers is equal to the product of the change in velocity of the air, the density of the air ρ_a, and the cross sectional area.

We assume that air above the UAV is quiescent, so the change in velocity of the air is $v_d\ \mathrm{m/s}$.

$$\text{Force gravity} = (\text{mass flow}) \cdot \text{air velocity}$$
$$m_d \cdot g = (v_d \cdot \rho_a \cdot d_d^2) \cdot v_d \tag{1}$$

Then the required *propwash*, the velocity of air beneath the UAV, for hovering is

$$v_d = \sqrt{\frac{m_d g}{\rho_a d_d^2}} \tag{2}$$

The flight testing site in Houston, Texas is $15\,\mathrm{m}$ above sea level. At sea level the density of air ρ_a is $1.225\,\mathrm{kg/m^3}$. The UAV and instrumentation combined weigh $5.1\,\mathrm{kg}$ with a width of $0.75\,\mathrm{m}$. The acceleration due to gravity is $9.871\,\mathrm{m/s^2}$. Substituting these values gives $v_d = 8.5\,\mathrm{m/s}$.

Due to propwash, an initially hovering mosquito will fall when under the UAV at a rate of v_d. Relative to the UAV, the mosquito moves horizontally at a rate of $-v_f$. As shown in Fig. 2, we can extend lines with slope $-v_d/v_f$ from the screen's trailing edge to h_{top} and from the leading edge to

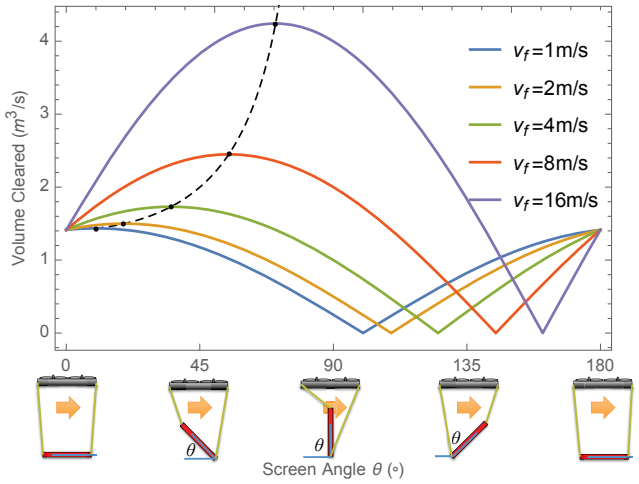

Fig. 3. The volume cleared by a UAV is a function of screen angle θ and forward velocity v_f. Dotted line shows the optimal angle given in (4).

h_{bottom}.

$$h_{\text{top}} = h_d - h_s + \frac{d_s}{2}\sin(\theta) + \frac{d_d + d_s\cos(\theta)}{2}\frac{v_d}{v_f}$$
$$h_{\text{bottom}} = h_d - h_s - \frac{d_s}{2}\sin(\theta) + \frac{d_d - d_s\cos(\theta)}{2}\frac{v_d}{v_f}$$
$$h_m = h_{\text{top}} - h_{\text{bottom}} = d_s\left(\frac{v_d}{v_f}\cos(\theta) + \sin(\theta)\right) \quad (3)$$

The optimal angle is therefore a function of forward and propwash velocity:

$$\theta = \text{ArcTan}\left(\frac{v_f}{v_d}\right) \quad (4)$$

To ensure the maximum number of mosquitoes are collected, the screen must be sufficiently far below the UAV $h_s > \frac{d_s}{2}\sin(\theta) + \frac{d_d + d_s\cos(\theta)}{2}\frac{v_d}{v_f}$ and the bottom of the screen must not touch the ground, $h_d > h_s + \frac{d_s}{2}\sin(\theta)$.

There are practical limits to h_s as well. Tests with $h_s > 2\,\text{m}$ were abandoned because the long length caused the screen to act as a pendulum, introducing dynamics that made the system difficult to fly.

Changing the flying height h_d of the UAV will target different mosquito populations because mosquitoes are not distributed uniformly vertically. Gillies and Wilkes demonstrated that different species of mosquitoes prefer to fly at different heights [24].

D. Wind Tunnel Verification of Net Angle

This section describes experiments run in a wind tunnel to verify the simplified net angle analysis in the previous section. Smoke streaklines were used to visualize the flow of air as it passed by the UAV. Due to space constraints in the wind tunnel, a free-flying phantom 4 was used instead of the hexacopter used for carrying the zapper. The wind tunnel was set to a $3\,\text{m/s}$ flow speed, and the UAV manually flown in approximately stable hovering. The solo UAV is $0.3 \times 0.3 \times 0.2\,\text{m}$. The windtunnel has a $1\,\text{m} \times 1\,\text{m}$ cross section. As seen from Fig. 4, the proposed screen position captures free flowing air and air entrained by the UAV propellers. This test encouraged us to mount the net as close to the UAV as possible, so that air, and flying mosquitos, entrained by the propellers are pushed into the net.

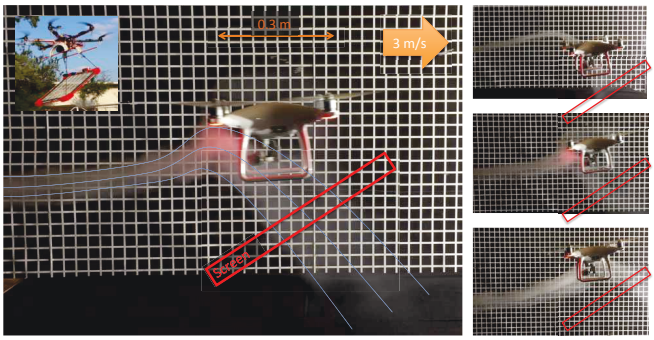

Fig. 4. Frames from wind tunnel test with free-flying UAV at $3\,\text{m/s}$ windspeed with smoke for streaklines [7]. As shown in the frames at right, the proposed screen position (in red) captures free flowing air and air entrained by the UAV propellers. Each black square is $25.4\,\text{mm}$ in width.

E. Data Logger

The electrical detection and logging system is powered by a 9 V lithium ion battery applied directly to the controller and two AA 3 V lithium ion batteries applied to the power circuit for the screen. The controller uses a GPS shield for monitoring the location and altitude as well as a real time clock to timestamp each data point collected from the system. A Raspberry Pi 3 is used for data logging, sensors include a GPS sensor (NEO-6M Ublox), a capacitive humidity sensor, a thermistor (DHT22), and an INA219 high side, 12-bit DC current sensor for monitoring the supply-side current delivered to the net. The net current draw is logged at 100 Hz, while GPS and weather sensor data is logged at 1Hz. All data is stored on an onboard SD card.

F. Energy Budget

Tests with an oscilloscope show that in the steady state, a $30.5\,\text{cm} \times 61\,\text{cm}$ screen and electronics have a power consumption of $3.6\,\text{W}$. During a zap, the screen voltage monitoring circuit shorts briefly when the mosquito contacts the screen. Figure 5 shows the time sequences for battery and screen voltages, current, and power during five mosquito zaps. Multiplying voltage by current to find the instantaneous power ($p = iv$) and integrating the area under the power curve show a total energy consumption of $4.2\,\text{mJ}$ for each zap. Recharging the screen requires more power and is represented in the latter part of the curves. The overall recovery time is about $160\,\text{ms}$. Most of the energy is consumed charging and maintaining the charge on the screen rather than in zapping the mosquitoes.

IV. Path Planning

A. Modelling

The data on the distribution of mosquitoes is given for a two-dimensional grid environment; the grid size is induced by the size of the screen, the available data and the desired resolution of the extracted map. For each pixel $p_i \in P$, we

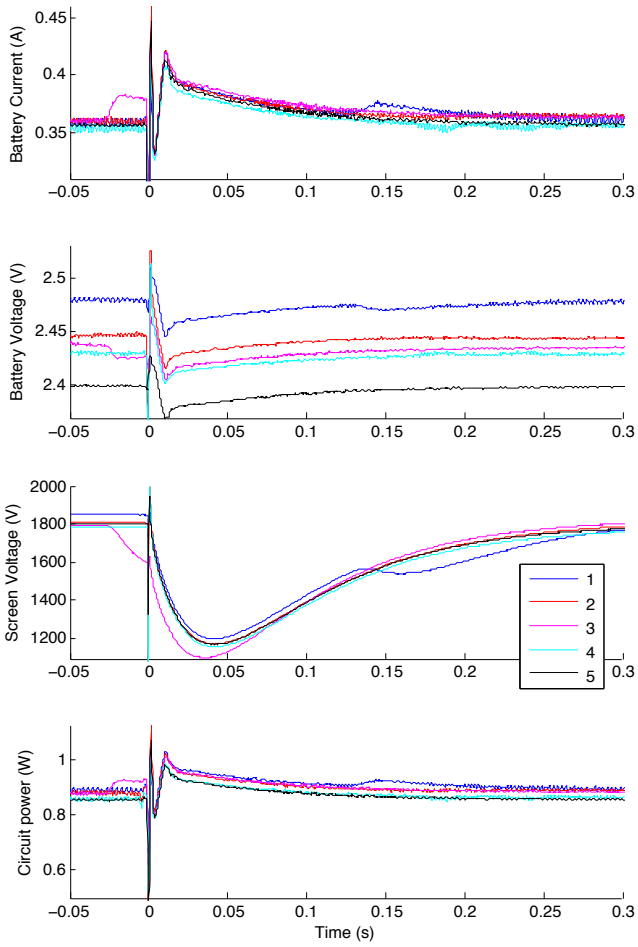

Fig. 5. Current, voltage, and power traces for five *Culex quinquefasciatus* mosquitoes as each contacts the bug-zapping screen at $t = 0$. Contact causes a brief short that recovers in 160 ms.

are given a relative value $c(p_i)$ that describes the estimated a-priori density of mosquitoes, based on data obtained from boustrophedon (back and forth) scans of the area by the UAV; this implies that only a subset of pixels carry a significant value. Visiting one of the pixels corresponds to sampling and mapping the actual density distribution of mosquitoes. For a dense distribution of mosquitoes (which is the case for the instances relevant for pest control), multiple visits to the same pixel do not contribute additional knowledge. As a consequence, the objective is to maximize the sampling value of the set $S \subseteq P$ of visited pixels, i.e., $\max_{S \subseteq P} \sum_{p_i \in S} c(p_i)$ within the available battery capacity; this may be over the course of a single closed trajectory, or over a combination of multiple roundtrips.

Planning good trajectories for a UAV is not subject to the same curvature constraints of an ordinary aircraft because UAVs can turn on the spot. However, turns are a critical aspect of path planning due to their impact on energy consumption. Battery capacity is *the* limiting factor for UAV flight time. As shown in Fig. 6, the power output for a desired trajectory is non-uniform. Flying along a straight path is relatively inexpensive but turning is energy intensive.

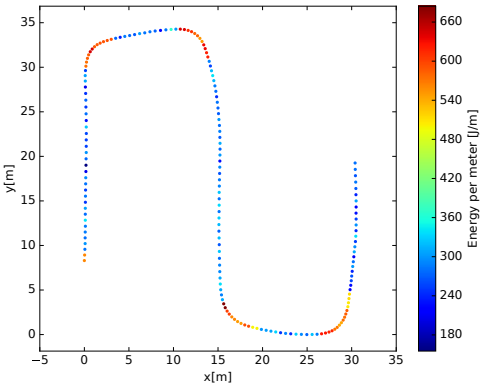

Fig. 6. Turns are expensive. See our related video at `https://youtu.be/SFyOMDgdNao` for details, and [25] for an accompanying abstract.

As a consequence, we must consider the total turn costs associated with changing direction, as measured by the turn angle. As we are not limited by trajectory curvature, we refer to straight-line connections and a finite set of 2ω different headings for visiting vertices. For the most natural case of orthogonal grids $\omega = 2$. When surveying non-isolated mosquito hotspots (whose size greatly exceeds the size of the UAV), we are not dealing with isolated pixels and the modeling error of this restriction is small.

Now we consider different trajectory types. A *cycle* is a roundtrip of a subset $S \subseteq P$ that visits all points in S and returns to the origin, a *cycle cover* of P is a set of cycles that together visit all points in P, and a *tour* is a single cycle that visits all points in P. A *subset cycle cover* for $S \subset P$ is a cycle cover that covers at least the points in S, while a *subset tour* is a tour of at least the points in S. For any of these structures, we are interested in cycle covers or tours of *minimum total travel cost*. The travel/battery cost is a linear combination of the number of pixel transitions (distance) and the weighted number of turns, corresponding to the total turn angle. In addition, a *minimum turn-cost penalty cycle cover* or a *minimum turn-cost penalty tour* visits a subset $R \subset P$, such that the sum of total travel cost and the sum $\sum_{i \notin R} c(p_i)$ of values of unvisited pixels is minimized.

B. Computational Complexity

Finding optimal covering paths that map a given region is closely related to the famous *Traveling Salesman Problem (TSP)*, which asks to minimize total length of a single tour that covers all of a given set of locations. The TSP is one of the classic NP-hard problems, so we cannot expect a general method that finds a provably optimal solution for any instance in polynomially bounded time. A generalization of the TSP is the *Lawnmower Problem* (see Arkin et al. [26], which considers coverage by a tool of nontrivial size. For the objective of minimizing the total cost (in particular, the turn cost), Arkin et al. [27] showed that finding minimum-turn tours in grid graphs is NP-hard, even if a minimum-turn cycle cover is given. The complexity of finding a set of multiple cycles that cover a given set of locations at minimum total turn cost had remained elusive for many years; *Problem 53*

in *The Open Problems Project* asks for the complexity of finding a minimum-cost (full) cycle cover in a 2-dimensional grid graph. This is not obvious: large parts of a solution can usually easily be deduced by local information and 2-factor techniques. Arkin et al. showed [27], [28] that the full coverage variant in *thin* grid graphs (which do not contain a 2×2 square, so every pixel is a boundary pixel) is solvable in polynomial time. In separate work [29], two of us were able to resolve this issue by showing that finding a cycle cover of minimum turn cost is NP-hard.

C. Mathematical Optimization

A powerful approach for finding optimal solutions to instances of NP-hard problems is the use of Integer Programming (IP). While solving an IP still requires exponential time in the worst case, using carefully crafted mathematical models in combination with specific algorithm engineering and available IP solvers enables solving instances of considerable size to provable optimality. For our purposes, we can describe the problem as follows.

a) Penalty Cycle Covers: The set P of pixels corresponds to a given grid graph $G(P, E)$ in which each pixel $p_j \in P$ is adjacent to the set $N(p_j)$ of pixels in P that share an edge with p_j. Each vertex $p_j \in P$ has a scalar reward $c(p_j)$ for visiting (or penalty for not visiting), and a function $\text{cost}_j(i,k) \in \mathbb{Z}_0^+$ that maps the cost of traveling from p_i to p_j to p_k, where $p_i, p_k \in N(p_j)$ are adjacent pixels to p_j. This cost is symmetric, i.e. $\text{cost}_j(i,k) = \text{cost}_j(k,i)$. The integer program uses two types of variables: integer variables $x_{ijk} = x_{kji}$ that state how often passage $p_i - p_j - p_k$ or $p_k - p_j - p_i$ is used and Boolean variables y_j that indicate that the pixel $p_j \in V$ is not covered, i.e., the penalty is paid. This results in the following formulation:

$$\min \sum_{p_j \in P} \sum_{p_i, p_k \in N(p_j)} \text{cost}_j(i,k) \cdot x_{ijk} + \sum_{p_j \in P} c(p_j) \cdot y_j \quad (5)$$

with constraints

$$1 \leq 4 \cdot y_j + \sum_{p_i, p_k \in N(p_j)} x_{ijk} \leq 4, \quad \forall p_j \in P \quad (6)$$

$$2x_{jij} + \sum_{p_k \in N(p_i), p_k \neq p_j} x_{jik} = 2x_{iji} + \sum_{p_k \in N(p_j), p_k \neq p_i} x_{ijk} \quad , \quad \forall \{p_i, p_j\} \in E \quad (7)$$

$$x_{ijk} \in \mathbb{N}_0, y_j \in \mathbb{B}, \quad \forall p_j \in P, \{p_i, p_k\} \subseteq N(p_j) \quad (8)$$

The objective function in Eq. 5 minimizes the total cost of the cycles and the uncovered pixels. Eq. 6 enforces a pixel to be covered or the *not covered* variable to be set to `true`. Arkin et al. [27] showed that no pixel needs to be visited more than four times, otherwise a simple local optimization can be performed. Eq. 7 enforces the transitions between two adjacent pixels to match. Eq. 8 enforces that the variables are integers or booleans.

We can solve a wide spectrum of instances with different kinds of probability distributions up to a size of 1500 pixels to provable optimality. Optimal solutions for different densities scalings of an instance with 1783 pixels are shown in Fig. 7. To solve larger instances the optimality constraint can be relaxed or the grid graph can be split and the subgraphs solved separately.

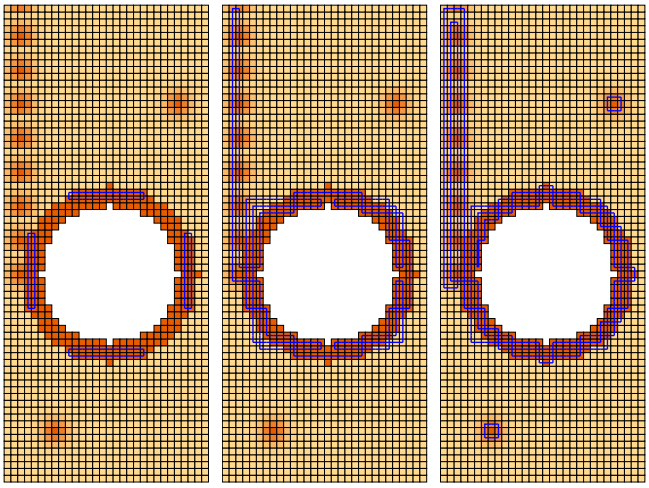

Fig. 7. Optimal cycle covers with different density scaling. The middle has twice and the right instance has four times the density as the left. In these instances, the cost of a 90° is five times that of a straight pixel transition.

b) Tours: Computing a minimum cycle cover may result in several subcycles that need to be visited separately, which is appropriate for the use of several UAVs or when several separate roundtrips by the same UAV are convenient. If we want to determine connected roundtrips by a single UAV, we need to connect the components of a cycle cover to a tour. This can be achieved via integer programming by adding additional constraints for separating these *subtours*.

This separation of subtours is more complicated than for the classic TSP because there may be tours that cross but are not connected. Instead of connecting two subtours, one subtour can also be discarded.

We first consider a constraint (Eq. 9) that is able to separate any given solution with multiple subtours. Let Q be the pixels of a selected subtour. Let $p_\ell \in Q$ be a pixel with high density and no other subtours crossing it and $p_{\ell'} \notin Q$ be another covered pixel with high density. These two pixels are used for 'defusing': if one of them is no longer covered, the constraint is automatically fulfilled. We denote by Q_s the pixels that are covered only by straight paths in the subtour. $T(p_j)$ describes the turn variables of a pixel p_j. x' refers to the variable assignment in the current solution.

$$1 \leq y_\ell + y_{\ell'} + \sum_{p_i, p_k \in N(p_\ell), x'_{i\ell k} = 0} x_{i\ell k} + \sum_{t \in T(v), v \in Q_s - p_\ell} t + \sum_{p_j \in Q \setminus (Q_s + p_\ell), p_i \neq p_k \in N(p_j), x'_{ijk} = 0} x_{ijk} \quad (9)$$

While this constraint suffices for capturing the mathematical conditions, its practical performance is unsatisfactory for connecting distant subtours. A better approach is described in the following; this is not always sufficient but more efficient in connecting distant cycles. We use the same definitions as for the previous constraint, but consider an additional set Q' that is a superset of Q. Q are the pixels of a subtour and

p_ℓ is a valuable pixel. Q' is a superset of Q (possibly equal to Q). $p_{\ell'}$ is a valuable pixel outside of Q'. The constraint enforces that either p_ℓ or $p_{\ell'}$ is uncovered, or there is a path on the margin of Q' that connects p_ℓ and $p_{\ell'}$.

$$y_\ell + y_{\ell'} + \sum_{x \in \text{Leaving}(Q')} x \geq 1 \tag{10}$$

We use two ways to choose Q' for these subtour elimination constraints: $Q' = Q$ which is similar to the classical TSP constraint or Q' is the Voronoi cell of the subtour.

D. Computational Results

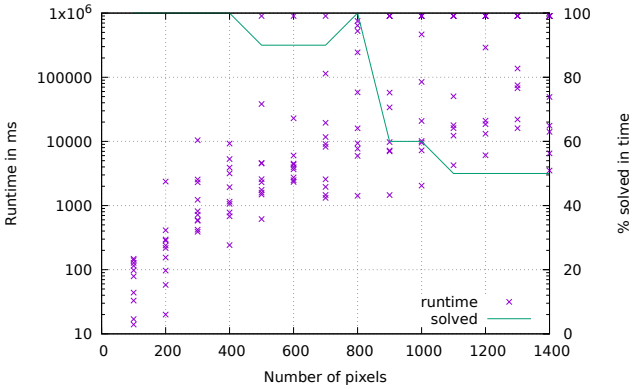

Fig. 8. Runtime of solving benchmark instances to optimality. Shown are the times of ten instances for each size, with a timeout at 900 s, as well as the percentage of solved instances. Only the number of turns is minimized in these instances.

We evaluated the effectiveness of our optimization method by testing it on a suite of benchmark instances based on random natural grid graphs with random densities; the probability of a pixel to be added during test instance generation is correlated with its neighborhood, resulting in smoother boundaries which are more natural than purely random instances. The tests were carried out for 10 instances for each size in the range up to 1400 pixels. We used modern desktop computers equipped with an *Intel(R) Core(TM) i7-6700K CPU* @ 4.00 GHz and 64 GB of RAM. The integer programs were computed with CPLEX version 12.5.0.0 and the parameters `EpInt=0`, `EpGap=0`, `EpOpt=1e-9`, and `EpAGap=0`. Fig. 8 shows runtimes for solving penalty cycle cover to optimality. Instances that took longer than 15 min were aborted. As shown in the figure, even at 1400 pixels we were still able to solve half of the instances to provable optimality. Even for the aborted instances, the computed solutions were within a few percentage points of the provable lower bound, meaning that they were nearly optimal.

Fig. 9 shows an example of iteratively computing an optimal tour with the described integer program. This example took less than a minute of total computing time. It assumed that 90° turns cost five times as much as a straight pixel transition (distance).

V. Experiments

The results for representative flights are described below. Figure 10 compares the energy consumption for three coverage schemes for a region including a large obstacle in the center. A boustrophedon path requires 50 turns, 187 kJ, 160 s, and 181 m. A hand-designed path requires 45 turns, 214 kJ, 155 s, and 178 m. A path computed using the optimal penalty cycle cover requires only 33 turns, 184 kJ, 133 s, and 176 m.

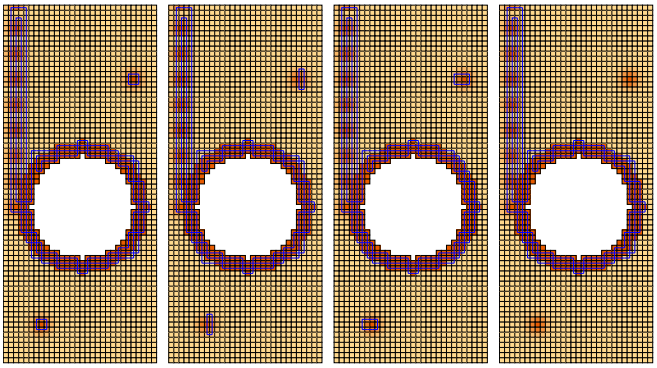

Fig. 9. Left is an optimal penalty cycle cover. Cycles (blue) cover all areas with high density. After three applications of the tour constraints, a single cycle remains (right). In the intermediate solutions, the subcycles first try to evade the new constraints by reshaping. The final tour omits two of the small hotspots because the cost of integrating them into the single tour is prohibitively expensive.

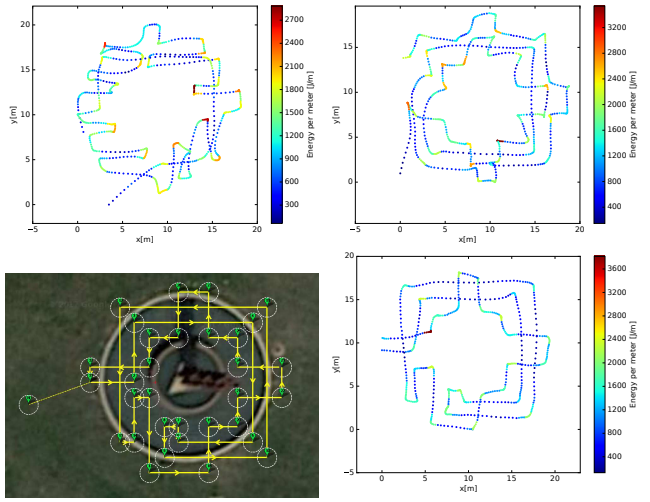

Fig. 10. Paths for an environment surrounding a fountain, which poses an obstacle for the UAV. (Top left) The energy consumption during a real world flight for a boustrophedon path. (Top right) The energy consumption during a real world flight for a hand-designed loop path. (Bottom left) The optimal penalty cycle cover path. (Bottom right) The energy consumption during a real-world flight along the optimal path.

A boustrophedon (back-and-forth) path with 2 m spacing was generated to cover a region 120 m×15 m at height 1.5 m. The path was generated using Mission Planner software from ardupilot.org [30].

For each trial the UAV took off from a resting position on top of the screen. Flight began manually, with a piloted takeoff of the UAV. After establishing a stable hover at 3 m, control was switched to the autonomous flight plan. The pilot monitored the flight with the ability to switch to manual operation in case of potential crashes due to GPS error or hazards in the flight plan. Mosquito strikes detected by the

Fig. 11. The UAV's path for flight 3 is in red. Strikes collected along this path are represented by yellow dots.

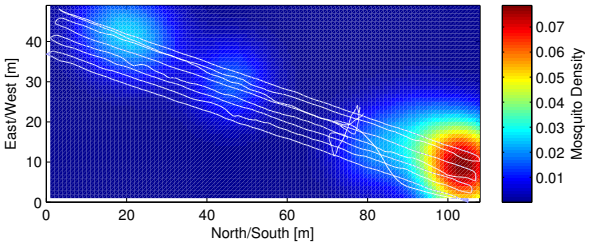

Fig. 12. Density map showing mosquito distribution on the field, overlaid by flight path 4 in white.

data logger were verified using a GoPro Hero 4 Silver camera attached at the top of the net, as shown in Fig. 1. At night and twilight, the sparks could be detected both visually and audibly from the recorded video. During the day, the sparks were loud enough to observe over the audio channel of the videos.

The UAV flew eight missions on this field, covering the same path. It was mainly flown in the early morning and late afternoon, when mosquito activities are more active. Three flights were flown at noon and early afternoon to ensure that mosquito activities during these periods were not ignored. However, only two mosquito strikes were observed during this period. The path covered is about $1\,\mathrm{km}$ long and typically takes $12\,\mathrm{min}$.

Over the eight missions on this field, there were a total of 11 mosquito strikes. Figure 11 shows the mission's flight path and the map of all collected strikes. The mosquito strikes are concentrated at the north and south ends of the field, where there are more trees. A density map was generated from the collected strikes' position by representing each strike by a Gaussian distribution with the norm on the strike's location and a σ of $10\,\mathrm{m}$. Figure 12 shows the density map generated by summing these Gaussian distributions.

These results not only tell where mosquitoes were but also show where mosquitoes were not. This is a key difference from stationary traps such as [5], [6]. Figure 13 shows the UAV during a dawn flight test near the ocean.

VI. Conclusion and Future Work

This paper presented an approach for finding optimal tours given turn costs and an energy budget, inspired by a mosquito-killing UAV with limited battery life. Initial experiments with the UAV and electrified screen track the location of a mosquito-killing UAV as it patrols a field and maps mosquito kills.

Fig. 13. The UAV and screen during a flight trial near the ocean.

Many refinements to the algorithm could be pursued in future work, including changes to both the mosquito-biasing algorithm and the robot flight simulation. The model may be expanded to continuous space, three dimensions, and to arbitrary turn angles. These and other considerations will make a more realistic model for future work.

Further testing of the multi-copter UAV is indicated and will allow more extensive testing of the robustness and accuracy of the hardware design. New sensors that can identify and detect flying insects [5] may be added to the UAV and enable it to proactively steer toward insect swarms and identify insects in realtime.

The concept may be extended to a non-destructive population survey in which the screen could be replaced with a net and, with appropriate lighting, the camera used to record capture events. Teams of UAVs could work together to map areas more quickly and, by measuring gradients of the distribution, quickly find large mosquito populations.

Acknowledgment

The authors acknowledge the helpful advice and feedback from Martin Reyna Nava, MS, Medical Entomologist and Technical Operations Manager and Mustapha Debboun, Ph.D, BCE, Director Mosquito Control Division, of the Harris County Public Health & Environmental Services, Mosquito Control Division. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1553063 and IIS-1646607. Thank you to Professor Daniel Araya for designing and running the wind tunnel verification tests.

References

[1] C. J. Murray, L. C. Rosenfeld, S. S. Lim, K. G. Andrews, K. J. Foreman, D. Haring, N. Fullman, M. Naghavi, R. Lozano, and A. D. Lopez, "Global malaria mortality between 1980 and 2010: a systematic analysis," *The Lancet*, vol. 379, no. 9814, pp. 413–431, 2012.

[2] J. E. Parker, N. Angarita-Jaimes, M. Abe, C. E. Towers, D. Towers, and P. J. McCall, "Infrared video tracking of anopheles gambiae at insecticide-treated bed nets reveals rapid decisive impact after brief localised net contact," *Scientific Reports*, vol. 5, 2015.

[3] S. Butail, N. Manoukis, M. Diallo, A. S. Yaro, A. Dao, S. F. Traoré, J. M. Ribeiro, T. Lehmann, and D. A. Paley, "3d tracking of mating events in wild swarms of the malaria mosquito anopheles gambiae," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 720–723.

[4] G. M. Williams and J. B. Gingrich, "Comparison of light traps, gravid traps, and resting boxes for west nile virus surveillance," *Journal of Vector Ecology*, vol. 32, no. 2, pp. 285–291, 2007.

[5] Y. Chen, A. Why, G. Batista, A. Mafra-Neto, and E. Keogh, "Flying insect classification with inexpensive sensors," *Journal of Insect Behavior*, vol. 27, no. 5, pp. 657–677, 2014.

[6] A. Linn, "Building a better mosquito trap," *International Pest Control*, vol. 58, no. 4, p. 213, 2016.

[7] S. Bhatnagar, A. Nguyen, D. Krupke, S. P. Fekete, and A. T. Becker, "Uav for destructive surveys of mosquito population (video)," Feb. 2018. [Online]. Available: https://youtu.be/OTQSR03Bv5g

[8] H. Choset, "Coverage for robotics - a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 113–126, October 2001.

[9] D. Spears, W. Kerr, and W. Spears, "Physics-based robot swarms for coverage problems," *The International Journal of Intelligent Control and Systems*, vol. 11, no. 3, 2006.

[10] S. Koenig, B. Szymanski, and Y. Liu, "Efficient and inefficient ant coverage methods," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1, pp. 41–76, Oct. 2001.

[11] J. A. Dennett, A. Bala, T. Wuithiranyagool, Y. Randle, C. B. Sargent, H. Guzman, M. SIIRIN, H. K. Hassan, M. Reyna-Nava, T. R. Unnasch *et al.*, "Associations between two mosquito populations and west nile virus in harris county, texas, 2003–06," *Journal of the American Mosquito Control Association*, vol. 23, no. 3, p. 264, 2007.

[12] R. Peter, P. Van den Bossche, B. L. Penzhorn, and B. Sharp, "Tick, fly, and mosquito control—lessons from the past, solutions for the future," *Veterinary parasitology*, vol. 132, no. 3, pp. 205–215, 2005.

[13] W. H. Organization, "Guidelines for laboratory and field testing of mosquito larvicides," *World Health Organization communicable disease control, prevention and eradication WHO pesticide evaluation scheme*, 2005.

[14] D. V. Maliti, N. J. Govella, G. F. Killeen, N. Mirzai, P. C. Johnson, K. Kreppel, and H. M. Ferguson, "Development and evaluation of mosquito-electrocuting traps as alternatives to the human landing catch technique for sampling host-seeking malaria vectors," *Malaria Journal*, vol. 14, no. 1, p. 1, 2015.

[15] J. M. Marshall and C. E. Taylor, "Malaria control with transgenic mosquitoes," *PLoS Med*, vol. 6, no. 2, p. e1000020, 2009.

[16] C. A. Hill, F. C. Kafatos, S. K. Stansfield, and F. H. Collins, "Arthropod-borne diseases: vector control in the genomics era," *Nature Reviews Microbiology*, vol. 3, no. 3, pp. 262–268, 2005.

[17] M. O. Ndiath, S. Sougoufara, A. Gaye, C. Mazenot, L. Konate, O. Faye, C. Sokhna, and J.-F. Trape, "Resistance to ddt and pyrethroids and increased kdr mutation frequency in an. gambiae after the implementation of permethrin-treated nets in senegal," *PLoS One*, vol. 7, no. 2, p. e31943, 2012.

[18] ScienceDaily. (1997, Jul.) Snap! Crackle! Pop! Electric bug zappers are useless for controlling mosquitoes, says UF/IFAS pest expert. [Online]. Available: http://www.sciencedaily.com/releases/1997/07/970730060806.htm

[19] P. Anupa Elizabeth, M. Saravana Mohan, P. Philip Samuel, S. Pandian, and B. Tyagi, "Identification and eradication of mosquito breeding sites using wireless networking and electromechanical technologies," in *Recent Trends in Information Technology (ICRTIT), 2014 International Conference on*. IEEE, 2014, pp. 1–6.

[20] B. Hur and W. Eisenstadt, "Low-power wireless climate monitoring system with rfid security access feature for mosquito and pathogen research," in *Mobile and Secure Services (MOBISECSERV), 2015 First Conference on*. IEEE, 2015, pp. 1–5.

[21] J.-T. Amenyo, D. Phelps, O. Oladipo, F. Sewovoe-Ekuoe, S. Jadoonanan, S. Jadoonanan, T. Tabassum, S. Gnabode, T. D. Sherpa, M. Falzone *et al.*, "Medizdroids project: Ultra-low cost, low-altitude, affordable and sustainable uav multicopter drones for mosquito vector control in malaria disease management," in *Global Humanitarian Technology Conference*. IEEE, 2014, pp. 590–596.

[22] C. Boonsri, S. Sumriddetchkajorn, and P. Buranasiri, "Laser-based mosquito repelling module," in *Photonics Global Conference (PGC), 2012*. IEEE, 2012, pp. 1–4.

[23] J. Kare and J. Buffum, "Build your own photonic fence to zap mosquitoes midflight [backwards star wars]," *IEEE Spectrum*, vol. 5, no. 47, pp. 28–33, 2010. [Online]. Available: http://spectrum.ieee.org/consumer-electronics/gadgets/backyard-star-wars

[24] M. Gillies and T. Wilkes, "The vertical distribution of some West African mosquitoes (Diptera, Culicidae) over open farmland in a freshwater area of The Gambia," *Bulletin of entomological research*, vol. 66, no. 01, pp. 5–15, 1976.

[25] A. T. Becker, M. Debboun, S. P. Fekete, D. Krupke, and A. Nguyen, "Zapping Zika with a mosquito-managing drone: Computing optimal flight patterns with minimum turn cost," in *32nd Symposium on Computational Geometry (SoCG)*, ser. LIPIcs-Leibniz International Proceedings in Informatics, vol. 77, 2017, pp. 62:1–62:5.

[26] E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell, "Approximation algorithms for lawn mowing and milling," *Comput. Geom.*, vol. 17, no. 1-2, pp. 25–50, 2000.

[27] E. M. Arkin, M. A. Bender, E. D. Demaine, S. P. Fekete, J. S. B. Mitchell, and S. Sethia, "Optimal covering tours with turn costs," *SIAM Journal on Computing*, vol. 35, no. 3, pp. 531–566, 2005.

[28] ——, "Optimal covering tours with turn costs," in *Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA01)*, 2001, pp. 138–147.

[29] S. P. Fekete and D. Krupke, "Computing optimal covering tours and cycle covers with turn cost," 2017, manuscript.

[30] ardupilot.org. (2016) Motion planner overview. [Online]. Available: http://ardupilot.org/planner/docs/mission-planner-overview.html