Long-Endurance Sensing and Mapping Using a Hand-Launchable Solar-Powered UAV

Philipp Oettershagen, Thomas Stastny, Thomas Mantel, Amir Melzer, Konrad Rudin, Pascal Gohl, Gabriel Agamennoni, Kostas Alexis and Roland Siegwart

Abstract This paper investigates and demonstrates the potential for very long endurance autonomous aerial sensing and mapping applications with AtlantikSolar, a small-sized, hand-launchable, solar-powered fixed-wing unmanned aerial vehicle. The platform design as well as the on-board state estimation, control and path-planning algorithms are overviewed. A versatile sensor payload integrating a multi-camera sensing system, extended on-board processing and high-bandwidth communication with the ground is developed. Extensive field experiments are provided including publicly demonstrated field-trials for search-and-rescue applications and long-term mapping applications. An endurance analysis shows that AtlantikSolar can provide full-daylight operation and a minimum flight endurance of 8 h throughout the whole year with its full multi-camera mapping payload. An open dataset with both raw and processed data is released and accompanies this paper contribution.

1 Introduction

The field of aerial robotics has seen rapid growth in the last decade. Prerequisite technologies have developed to the point that we are not far from the day when utilization of aerial robots is prevalent in our society. With an application range that includes infrastructure inspection [13], surveillance for security tasks [6], disaster relief [8, 25], crop monitoring [7], mapping [1], and more, Unmanned Aerial Vehicles (UAVs) already provide added value to several critical and financially significant applications, and are widely acknowledged for their potential to achieve a large impact in terms of development and growth. Examples of compelling existing use-cases include the mapping of the Colorado flood area in 2003 [4], the 3D reconstruction of the "Christ the Redeemer" statue in Brazil and the Matterhorn mountain reconstruction [20], and the live offshore flare inspection that took place in the North Sea [3].

P. Oettershagen (✉) · T. Stastny · T. Mantel · A. Melzer · K. Rudin · P. Gohl · G. Agamennoni · K. Alexis · R. Siegwart
Autonomous Systems Lab, ETH Zurich, Zurich, Switzerland
e-mail: philipp.oettershagen@mavt.ethz.ch

D.S. Wettergreen and T.D. Barfoot (eds.), *Field and Service Robotics*,
Springer Tracts in Advanced Robotics 113, DOI 10.1007/978-3-319-27702-8_29

While these are impressive achievements, there are still major factors that limit the applicability of UAVs. One such factor is their relatively low endurance. Indeed, long-endurance flight capabilities are crucial for applications such as large-scale Search-and-Rescue support, industrial pipeline monitoring, atmospheric research, offshore inspection, precision agriculture and wildlife monitoring. This new class of problems exposes a practical limitation in the majority of currently available aerial robot configurations.

Solar-powered flight is a key enabling technology for long-endurance operations. By harnessing the sun's energy and storing solar power during the day, flight times can be significantly prolonged. In cases of extreme designs, sustained flight can even be achieved through night time and/or cloudy conditions. An existing example of extreme endurance is the QinetiQ Zephyr UAV (22 m wingspan), which broke records, sustaining flight for 2 weeks [24]. However, scaling down from the high-altitude "pseudo satellite" class to more manageable, rapidly deployable and low-altitude designs is not trivial.

Motivated by the increasing industrial, scientific and societal demand for persistent automatic aerial sensing and surveillance, long-endurance, solar-powered fixed-wing aircrafts have been a research priority in the Autonomous Systems Lab (ASL) at ETH Zurich. With the most recent development being the AtlantikSolar UAV, our aim is to extend the current technological state of the art with a robust and versatile platform capable of significantly longer term sensing and mapping on the order of days or even weeks. Figure 1 depicts the AtlantikSolar UAV and its sensing capabilities. The detailed design of this UAV platform has been described in [18]. This paper extends our previous design-oriented work by investigating and characterizing possible application scenarios for our platform. More specifically, we present a set

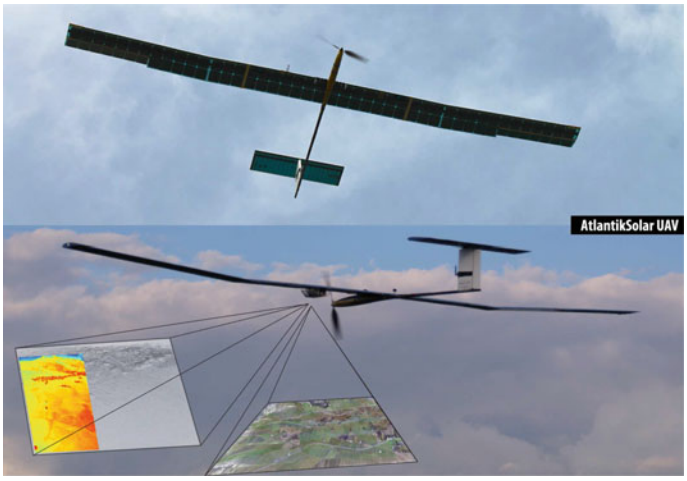


Fig. 1 The AtlantikSolar UAV is capable of very long-endurance operation in missions including mapping, surveillance, victim detection and infrastructure inspection

of field trials that are enabled by a diverse sensor payload recently integrated into the UAV. This on-board sensor payload includes RGB and grayscale camera systems and a thermal vision sensor in combination with a complete suite of sensors that enable the vehicle to navigate autonomously.

The remainder of this paper is organized as follows: We present a description of the AtlantikSolar vehicle in Sect. 2, its sensing and mapping capabilities in Sect. 3, field experiment results in Sect. 4, and derived conclusions in Sect. 5. We also provide a detailed discussion of our experiences from both search-and-rescue as well as mapping missions, and release a dataset containing raw as well as post-processed data.

2 AtlantikSolar Unmanned Aerial Vehicle

2.1 Platform Overview

The AtlantikSolar UAV (Fig. 2, Table 1) is a small-sized, hand-launchable, low-altitude long-endurance (LALE), solar-powered UAV optimized for large-scale aerial mapping and inspection applications. A detailed overview of the conceptual design of AtlantikSolar is given in [18]. The design methodology is based on the work in [10, 16] with extensions on optimizing solar-powered UAVs for a range of deteriorated meteorological conditions (e.g. cloud obstruction of sun radiation) as given in [18]. The platform owes much of its configuration to the optimization of power consumption. Lightweight composite materials are used in the fabrication of a torsionally resistant cylindrical carbon fibre spar, tapered carbon fibre tail boom, and fibreglass fuselage body. The AtlantikSolar prototype UAV used for the flight tests in this paper features 88 SunPower $E60$ cells with an efficiency of $\eta_{sm} = 0.23$. Energy is stored in 2.9 kg of cylindrical high energy density Li-Ion batteries (Panasonic NCR18650b,

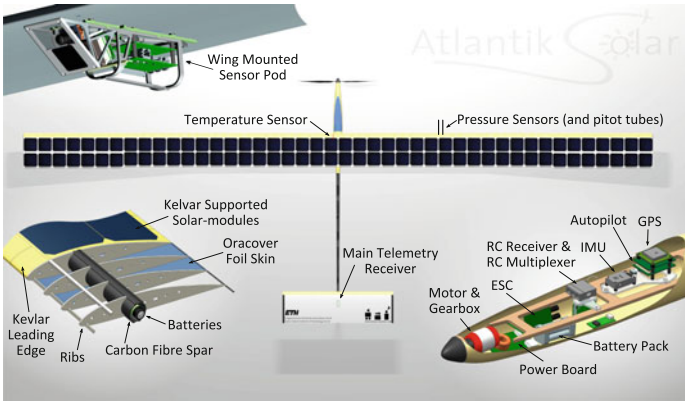


Fig. 2 AtlantikSolar system overview

Table 1 Summary of AtlantikSolar design and performance characteristics

Specification	Value/unit
Wing span	5.65 m
Mass	7.5 kg
Nominal cruise speed	9.7 m s^{-1}
Max. flight speed	20 m s^{-1}
Min. endurance (no payload)[a]	13 h
Design endurance (no payload)	10 days

[a]On battery-power only

243 Wh kg^{-1}, 700 Wh total) that are integrated into the wing spar for optimal weight distribution. The two ailerons, the elevator and the rudder are driven by brushless Volz DA-15N actuators with contactless position feedback. The propulsion system consists of a foldable custom-built carbon-fibre propeller, a 5:1 reduction gearbox and a 450 W brushless DC motor.

A Pixhawk PX4 Autopilot, an open source/open hardware project started at ETH Zurich [21], is the centerpiece of the avionics system. It employs a Cortex M4F microprocessor running at 168 MHz with 192 kB RAM to perform autonomous flight control and state estimation. Major hardware modifications include the integration of the ADIS16448 IMU and the Sensirion SDP600 differential pressure as well as re-writing of the estimation and control algorithms.

2.2 Operational Concept

AtlantikSolar is hand-launched to enable rapid deployment and operation in remote or uneven terrain. It is operated by a two-person team consisting of the safety-pilot and an operator for high-level mission management through the ground control station (GCS) interface (QGroundControl [23]). The GCS allows automatic loitering and autonomous waypoint following of user-defined or pre-computed paths. For visual-line-of-sight operation, the primary (434 MHz) telemetry link is sufficient, but an Iridium satellite link is also integrated to act as a backup link in the event of primary radio loss or beyond-visual-line-of-sight operation (Fig. 3). The UAV is equipped with a wing-mounted sensor pod, but provides additional payload capacity and versatility within its total payload budget of $m_{pld,max} \approx 800$ g. AtlantikSolar also integrates four high-power LEDs for night operations.

2.3 Enabling Technologies for Autonomous Navigation

2.3.1 Robust Long-Term State Estimation

To provide reliable and drift-free long-term autonomous operation, a light-weight EKF-based state estimator, as presented in [11], is implemented on the autopilot.

Fig. 3 Communications and ground control

It fuses data from a 10-DoF Inertial Measurement Unit (IMU) with GPS-Position, GPS-velocity and airspeed measurements in order to successively estimate position, velocity, orientation (attitude and heading), QFF as well as accelerometer and gyro biases. Robustness against temporal GPS losses is enhanced through the inclusion of airspeed measurements from a differential barometer. To increase flight safety, the algorithm estimates the local three-dimensional wind vector and employs an internal aircraft aerodynamics model to estimate the current sideslip angle and Angle of Attack (AoA), which can in turn be used by the flight controller to apply implicit flight regime limits, as in the case of the authors' previous work [17].

2.3.2 Flight Control

AtlantikSolar's flight control system features automatic tracking of waypoints along pre-defined paths, allows extended loitering around areas of interest and implements safety-mechanisms such as automatic Return-To-Launch (RTL) in case of prolonged remote control or telemetry signal losses. The baseline control is a set of cascaded PID-controllers for inner-loop attitude control [2]. Output limiters are applied to respect the aircraft flight envelope, dynamic pressure scaling of the control outputs is used to adapt to the changing moment generation as a function of airspeed and a coordinated-turn controller allows precise turning. Altitude control is based on a Total Energy Control System that also allows potential energy gains in thermal updraft while it implements safety mechanisms such as automatic spoiler deploy-

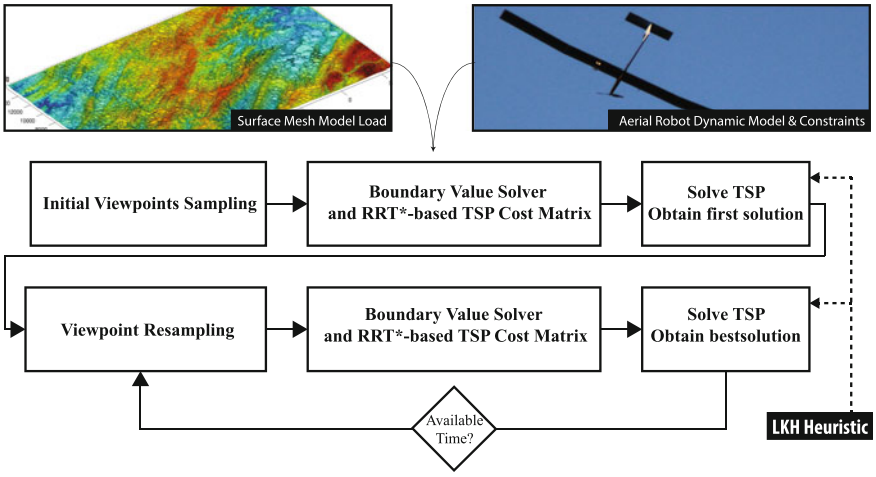


Fig. 4 Summary of the employed 3D inspection path-planning algorithm

ment during violation of maximum altitude limits. Waypoint-following is performed using an extended version of the $\mathcal{L}_1$-nonlinear guidance logic [19]. The detailed implementation and verification of our autopilot is described in [18].

2.3.3 Inspection Path-Planner

An inspection path-planning algorithm is integrated into the system in order to enable automated inspection and mapping of large scale 3D environments. The algorithm is inherently tailored for structural inspection and computes full coverage and collision-free paths subject to a model of the nonholonomic constraints of the vehicle. The overall approach is illustrated in Fig. 4, while a detailed description is available in the authors' previous work [1]. It essentially corresponds to an explicit algorithm that computes an inspection path based on a mesh-model representation of the desired world. It iteratively tries to compute viewpoint configurations that provide full coverage while at the same time employing the Lin-Kernighan heuristic [12] in the search for the best route that visits all of them subject to the motion constraints of the vehicle. Via a viewpoint resampling technique that employs randomized sampling, the designed algorithm allows for an iterative improvement of the path cost while always retaining complete coverage. Fast collision-free navigation is achieved via a combination of a Boundary Value Solver for the considered vehicle model with the RRT* [9] motion planner.

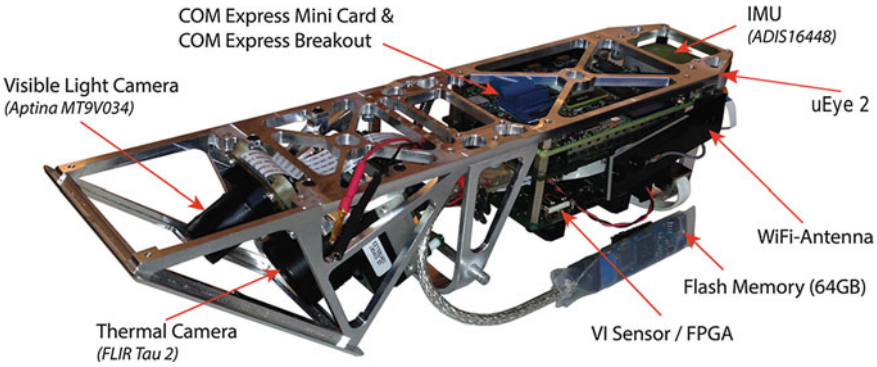


Fig. 5 The sensor pod as it is currently used on the AtlantikSolar UAV. The pod's fairing has been omitted for better visibility of the components

3 Sensor Pod

The sensor pod (see Fig. 5) features a grayscale (Aptina MT9V034) camera with a high dynamic range and a long-wavelength infrared (LWIR) camera (FLIR Tau 2) for thermal imaging, both mounted with an oblique field of view (FOV), as well as a nadir facing RGB camera (uEye XS 2). An IMU (Analog Devices ADIS16448) is also included, measuring linear accelerations, angular velocities, and the magnetic field in all three axes. All sensors are integrated with a Skybotix VI-sensor [27], allowing tight hardware synchronization and timestamping of the acquired data [15]. Furthermore, a state of the art embedded computer (Kontron COMe-mBT10), with an Intel Atom CPU (4 cores, 1.91 GHz) and a thermal design power (TDP) of 10 W, is interfaced with the VI-sensor and the PX4 autopilot board of the UAV. The on-board Atom computer further communicates with the PX4 in order to receive all global pose estimates and raw sensor data and transmit waypoints. The acquired data is processed on-board and communication with the ground control station is achieved over Wi-Fi. As shown in Fig. 5, all components are mounted on a lightweight aluminum construction ensuring a rigid connection between the cameras and the IMU, thus guaranteeing high quality extrinsic calibration of the sensors, a key element for accurate visual-inertial localization.

The on-board computer runs a standard Ubuntu Linux operating system, allowing quick adaptation to different kinds of missions. Furthermore, it enables rapid testing of new algorithms, e.g. for localization and mapping. It has been utilized to evaluate monocular localization [10] while the original stereo version of the VI-sensor is actively used for localization of rotary-wing UAVs in possibly GPS-denied environments [14]. Within the framework of the research projects ICARUS and SHERPA [8, 26], the described sensor pod is used for area mapping, victim detection, and situational awareness tasks. The data of the visible light cameras is combined with the pose estimates and fed to post-processing software [20] to derive accurate 3D recon-

structions of the environment. Active research is ongoing for aerial victim detection at altitudes on the order of 100 m.

4 Flight Experiments

AtlantikSolar is a key component of several research projects and has actively participated in multiple large-scale demonstration events. Within this paper, indicative results from the ICARUS project [8] public field-trials event at Marche-en-Famenne, Belgium and a long-endurance mapping mission in Rothenthurm, Switzerland are presented along with flight endurance related tests and evaluations. A dataset is also released and documented to accompany this paper. It contains the vehicle state estimates, IMU and GPS raw data, the camera frames from all the on-board modules as well as post-processed reconstructions of the environment for the field-trials described in Sect. 4.2. This rich dataset is publicly available at [5].

4.1 Search-and-Rescue Application Demonstration

During the ICARUS project field-trials in Marche-en-Famenne [8], the AtlantikSolar UAV was commanded to autonomously execute inspection paths that ensured the complete coverage of a predefined area in order to assist the area monitoring, mapping, victim detection and situational awareness necessities of Search-and-Rescue rapid response teams. Employing the path-planner overviewed in Sect. 2.3.3 and based on the long-endurance capabilities of the UAV, the area was scanned repeatedly over multiple hours. An example inspection path is depicted in Fig. 6 and corresponds to an optimized solution for the case of the oblique-view mounted thermal camera, FOV (56°, 60°) for the horizontal and vertical axes, respectively. The mounting orientations of the grayscale and the thermal camera are identical, but the FOV of the grayscale camera is larger in all directions (70°, 100°), thus the planned path provides full coverage for both vision sensors.

During the execution of these inspection paths, the two camera-system and the pose estimates of the aircraft were uniformly timestamped and recorded in a ROS bag. Subsequently, post-processing of the grayscale images was conducted in order to derive a dense point-cloud of the area using the Pix4D software [20]. An image of the derived result is shown in Fig. 7, while additional results of autonomously executed inspection paths may be found in our previous work [1].

Fig. 6 Inspection path full area coverage using the oblique-view mounted thermal vision and grayscale cameras of the AtlantikSolar sensor pod. The *colored* mosaic was derived using an additional very large field of view nadir-facing camera (HDR-AS100VW)

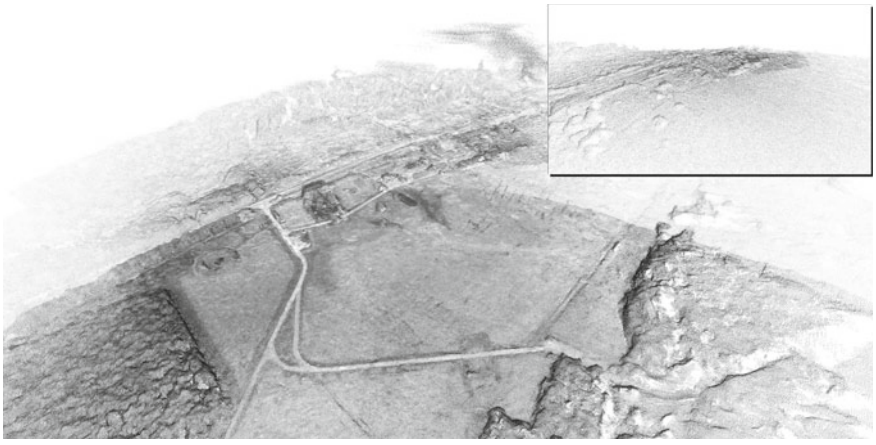


Fig. 7 Reconstructed dense point cloud based on the combination of the oblique-view grayscale camera images with the vehicle position estimates. The reconstruction was achieved using the Pix4D mapping software

4.2 Area Coverage Application Demonstration

In this specific field experiment, the AtlantikSolar UAV's capabilities for long-term area coverage, inspection and mapping were evaluated. Within 6 h of flight, the system performed multiple lawn-mowing and other paths like those presented in Fig. 8. With a camera frame recording rate set at $F_c = 1$ Hz, synchronization with the vehicle pose estimates and properly designed waypoint distances to ensure coverage and sufficient overlap for all cameras, a solid reconstruction result was achieved. Within this flight,

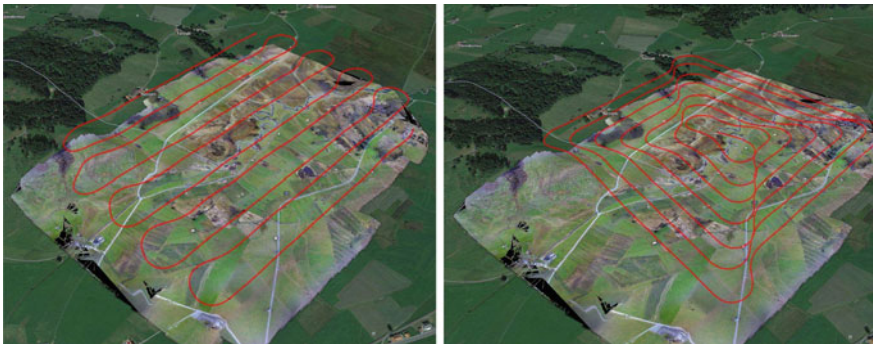


Fig. 8 The lawn-mowing path executed by the AtlantikSolar UAV overlayed on the reconstructed mosaic of the environment, incorporated in Google maps

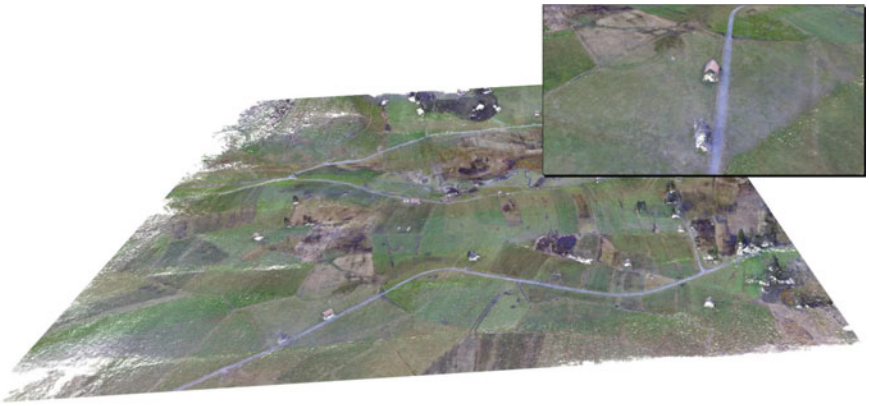


Fig. 9 The reconstructed point-cloud of the Rothenthurm area based on the combination of the RGB and grayscale camera data as well as the UAV pose estimates collected during the lawn-mowing path and subsequently processed using the Pix4D software

all three cameras were employed and Fig. 9 depicts the reconstructed point cloud using a combination of the geo-tagged nadir-facing RGB camera of the sensor pod with the, likewise, geo-tagged oblique-view grayscale images, while Fig. 10 shows false-colored thermal images that our team is currently aiming to employ for victim detection, extending previous work [22] at ASL. An open dataset containing 1 h of raw data and post-processed results is released to accompany this paper and may be found at [5].

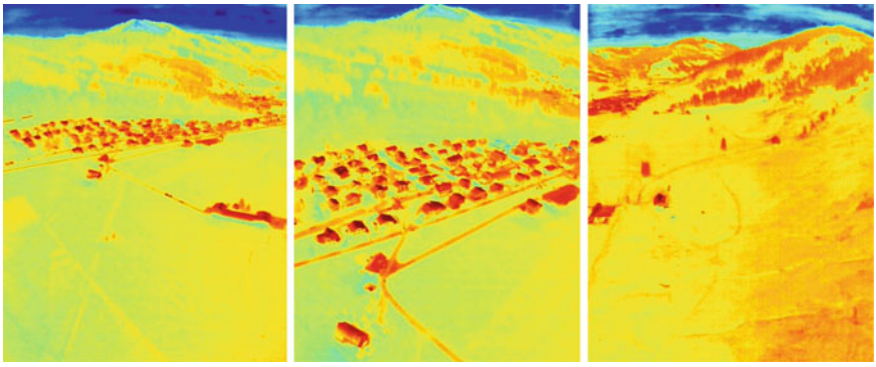


Fig. 10 False-colored thermal camera images recorded using the on-board sensor pod of the AtlantikSolar UAV

4.3 Full-Payload Flight Endurance and Range

After having shown a flight endurance of more than 12 h without payload in Summer 2014 [18], the area coverage demonstration in Rothenthurm on November 21st was used to determine AtlantikSolar's maximum flight endurance with the full sensor pod payload of $m_{payload} = 610\,\text{g}$ during winter conditions. Figure 11 shows the corresponding power income, output, and battery state. The average power consumption during the flight is $P_{aircraft} = 69.7\,\text{W}$ plus $P_{payload} = 15\,\text{W}$ for the sensor pod. After take-off at 10:25 a.m. local time at 94 % battery state-of-charge (SoC), the heavily attitude-dependent solar power income increases but reaches only a maximum of 80 W at noon due to the limited insolation in winter. Nevertheless, as indicated by the SoC, the system power is mostly drawn from the solar panels for more than 3 h of the flight. Power income decreases towards the afternoon: The solar panel maximum power point trackers (MPPTs) are still operating, but the panel voltage has decreased significantly and the MPPTs deliver currents below the measurement threshold. However, the remaining SoC during landing shortly before sunset (4:28 p.m. local time) is still 52 %. Extrapolating using the total power consumption of $P_{tot} = P_{aircraft} + P_{payload} = 69.7\,\text{W} + 15\,\text{W} = 84.7\,\text{W}$ yields an additional 4.32 h of remaining flight endurance assuming zero-radiation conditions and thus a total flight endurance of ca. 10 h with full payload for the installed 700 Wh battery during these winter conditions.

The recorded power consumption of $P_{tot} = 84.7\,\text{W}$ was taken as the input for the flight endurance simulation in Fig. 12. Assuming launch of the airplane exactly at sunrise, full-daylight flight endurance is provided throughout the full year including winter under most atmospheric conditions. More specifically, full-daylight flight capability is only lost when $CLR = P_{Solar}/P_{Solar,ClearSky}$ is smaller than ca. 0.3 in summer and ca. 0.15 in winter, which corresponds to severe cloud coverage or fog that may hinder flight operations independently of energy considerations. The maximum endurance of AtlantikSolar with the full payload is 22.4 h on June 21st, which

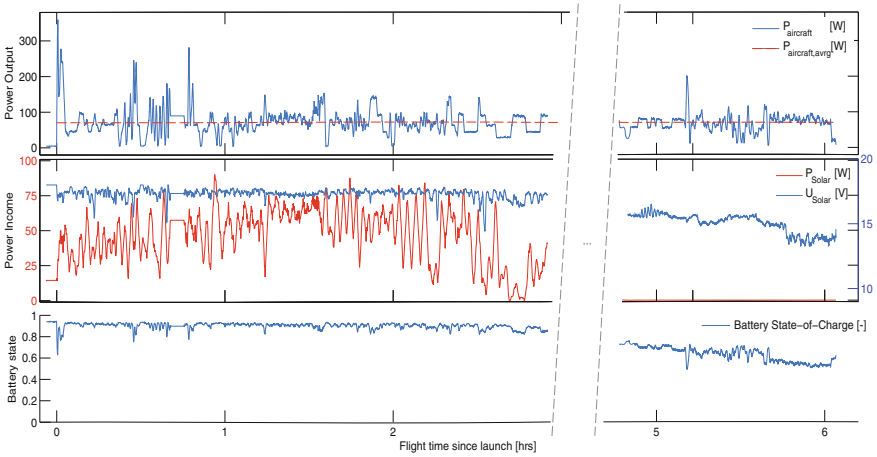


Fig. 11 Power income, output and battery state-of-charge for the Rothenthurm mapping flight. AtlantikSolar covered 243 km ground-distance, was airborne for 6 h 3 min and landed shortly before sunset (4:28 p.m. local time) with 52 % battery capacity remaining

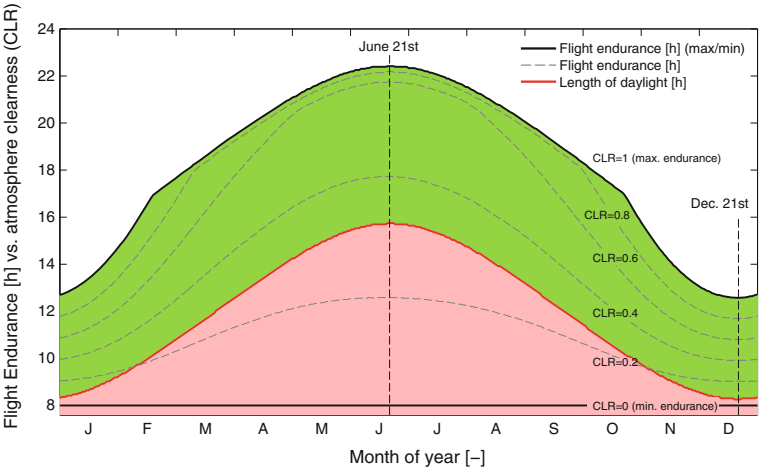


Fig. 12 AtlantikSolar's flight endurance with 610 g/15 W payload versus atmospheric clearness (CLR) for $\phi = 47°$ N (Rothenthurm, CH) when assuming launch at sunrise with SoC = 100 %. Full daylight flight is possible throughout the whole year (*green area*), and only severe cloud coverage can reduce endurance below the daylight duration (*red area*). In all cases 8 h of minimum endurance are achieved

means that perpetual flight is not possible. Note that in all atmospheric conditions, a minimum endurance of 8 h can be guaranteed through battery-powered flight alone. At the chosen airspeed of $v_{air} = 11.02$ m/s, AtlantikSolar can thus cover a ground distance of 317 km (min. endurance) to 888 km (max. endurance). Note that this airspeed provides the maximum range (optimal glide ratio), but is not the power-

optimal airspeed (lowest rate of sink). Flying strictly at the power-optimal airspeed found in [18] would e.g.increase the endurance to 23.9 h on June 21st, with battery energy depleting shortly before sunrise. This means that perpetual flight with the full sensor payload can theoretically be achieved through minor aircraft optimizations, e.g. through a slight increase of the available battery capacity. However, note increasing endurance through power-optimal airspeed selection in the non-perpetual flight endurance case comes at a cost of range, and should be considered per application.

5 Conclusions

In this work, we have demonstrated a significant leap in long-endurance, low-altitude aerial sensing and mapping. Utilizing optimized solar aircraft design methodologies, low power consumption electronics, a robust autonomous navigation framework, and a versatile, modular, and self-contained sensor payload, the AtlantikSolar system, as a whole, provides a baseline to address quickly approaching societal needs related to long-term aerial robotic operations. Extensive field-trial experience indicates that solar power is a promising solution towards providing long endurance to small-sized, low-altitude UAVs, and integrated sensor suites, when used in tandem with autonomous navigation and planning methods, can provide wealth of valuable information to end users in an efficient manner. Still, there is great room for improvement, especially in the directions of autonomous navigation close to terrain, where a combination of advanced perception and planning algorithms have to be employed. Also in terms of superior robustness, as required for multi-hour or even multi-day flight.

Acknowledgments This work was supported by the European Commission projects ICARUS (#285417) and SHERPA (#600958) under the 7th Framework Programme. Further information at http://www.atlantiksolar.ethz.ch/.

References

1. Bircher, A., Alexis, K., Burri, M., Oettershagen, P., Omari, S., Mantel, T., Siegwart, R.: Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. In: 2014 IEEE International Conference on Robotics and Automation (ICRA) (2015) (accepted)
2. Brian, L.S., Frank, L.L.: Aircraft Control and Simulation. Wiley Interscience (1992)
3. Cyberhawk—Remote aerial inspection and land surveying specialists (2015). http://www.thecyberhawk.com/
4. Falcon UAV (2015). http://www.falconunmanned.com/
5. FSR 2015—Solar-powered UAV Sensing and Mapping Dataset (2015). http://projects.asl.ethz.ch/datasets/doku.php?id=fsr2015
6. Girard, A., Howell, A., Hedrick, J.: Border patrol and surveillance missions using multiple unmanned air vehicles. In: 43rd IEEE Conference on Decision and Control, 2004. CDC (2004)

7. Hunt, E.R., Hively, W.D., Fujikawa, S.J., Linden, D.S., Daughtry, C.S.T., McCarty, G.W.: Acquisition of nir-green-blue digital photographs from unmanned aircraft for crop monitoring. Remote Sens. **2**(1), 290–305 (2010)
8. ICARUS: Unmanned Search and Rescue (2015). http://www.fp7-icarus.eu/
9. Karaman, S., Frazzoli, E.: Incremental sampling-based algorithms for optimal motion planning. CoRR (2010). http://www.abs/1005.0416
10. Leutenegger, S.: Unmanned solar airplanes: Design and algorithms for efficient and robust autonomous operation. PhD thesis, ETH Zurich (2014)
11. Leutenegger, S., Melzer, A., Alexis, K., Siegwart, R.: Robust state estimation for small unmanned airplanes. In: IEEE Multi-conference on Systems and Control (2014)
12. Lin, S., Kernighan, B.W.: An effective heuristic algorithm for the traveling-salesman problem. Oper. Res. **21**(2), 498–516 (1973)
13. Metni, N., Hamel, T.: A UAV for bridge inspection: visual servoing control law with orientation limits. Autom. Constr. **17**(1), 3–10 (2007)
14. Nikolic, J., Burri, M., Rehder, J., Leutenegger, S., Huerzeler, C., Siegwart, R.: A UAV System for Inspection of Industrial Facilities. In: IEEE Aerospace Conference (2013)
15. Nikolic, J., Rehder, J., Burri, M., Gohl, P., Leutenegger, S., Furgale, P.T., Siegwart, R.Y.: A Synchronized Visual-Inertial Sensor System with FPGA Pre-Processing for Accurate Real-Time SLAM. In: IEEE International Conference on Robotics and Automation (ICRA) (2014)
16. Noth, A.: Design of solar powered airplanes for continuous flight. PhD thesis, ETH Zurich (2008)
17. Oettershagen, P., Melzer, A., Leutenegger, S., Alexis, K., Siegwart, R.: Explicit Model Predictive Control and $\mathcal{L}_1$-Navigation Strategies for Fixed-Wing UAV Path Tracking. In: 22nd Mediterranean Conference on Control & Automation (MED) (2014)
18. Oettershagen, P., Melzer, A., Mantel, T., Rudin, K., Lotz, R., Siebenmann, D., Leutenegger, S., Alexis, K., Siegwart, R.: A Solar-Powered Hand-Launchable UAV for Low-Altitude Multi-Day Continuous Flight. In: IEEE International Conference on Robotics and Automation (ICRA) (2015)
19. Park, S., Deyst, J., How, J.P.: A new nonlinear guidance logic for trajectory tracking. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, pp. 16–19 (2004)
20. Pix4D (2015). http://pix4d.com/
21. Pixhawk Autopilot (2015). http://pixhawk.org/
22. Portmann, J., Lynen, S., Chli, M., Siegwart, R.: People detection and tracking from aerial thermal views. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1794–1800 (2014)
23. QGroundControl (2015). http://www.qgroundcontrol.org/
24. QinetiQ: QinetiQ files for three world records for its Zephyr Solar powered UAV. QinetiQ Press Release (2010). http://www.qinetiq.com/media/news/releases/Pages/three-world-records.aspx
25. Rudol, P., Doherty, P.: Human body detection and geolocalization for uav search and rescue missions using color and thermal imagery. In: Aerospace Conference, 2008 IEEE, pp 1–8 (2008)
26. SHERPA Project (2015). http://www.sherpa-project.eu/
27. Skybotix AG (2015). http://www.skybotix.com/

Aerial Vehicle Path Planning for Monitoring Wildfire Frontiers

Ryan C. Skeele and Geoffrey A. Hollinger

Abstract This paper explores the use of unmanned aerial vehicles (UAVs) in wildfire monitoring. To begin establishing effective methods for autonomous monitoring, a simulation (FLAME) is developed for algorithm testing. To simulate a wildfire, the well established FARSITE fire simulator is used to generate realistic fire behavior models. FARSITE is a wildfire simulator that is used in the field by Incident Commanders (IC's) to predict the spread of the fire using topography, weather, wind, moisture, and fuel data. The data obtained from FARSITE is imported into FLAME and parsed into a dynamic frontier used for testing hotspot monitoring algorithms. In this paper, points of interest along the frontier are established as points with a fireline intensity (British-Thermal-Unit/feet/second) above a set threshold. These interest points are refined into hotspots using the Mini-Batch K-means Clustering technique. A distance threshold differentiates moving hotspot centers and newly developed hotspots. The proposed algorithm is compared to a baseline for minimizing the sum of the max time untracked $J(t)$. The results show that simply circling the fire performs poorly (baseline), while a weighted-greedy metric (proposed) performs significantly better. The algorithm was then run on a UAV to demonstrate the feasibility of real world implementation.

1 Introduction

Recent developments in sensing technology have made possible low cost, reliable unmanned aerial vehicles (UAVs). These field robots are being implemented in various application domains, but specifically show promise in applications hazardous

This work was supported in part by NASA grant NNX14AI10G and ONR grant N00014-14-1-0509.

R.C. Skeele (✉) · G.A. Hollinger
Robot Decision Making Laboratory, School of Mechanical, Industrial & Manufacturing Engineering, Oregon State University, Corvallis, OR 97331, USA
e-mail: skeeler@onid.oregonstate.edu

G.A. Hollinger
e-mail: geoff.hollinger@oregonstate.edu

D.S. Wettergreen and T.D. Barfoot (eds.), *Field and Service Robotics*,
Springer Tracts in Advanced Robotics 113, DOI 10.1007/978-3-319-27702-8_30

for humans. Studying wildfires has an obvious benefit when considering the human cost spent combating them. One of the main issues in combating wildfires is monitoring the progression of the fire over time [13]. Live fire frontier monitoring can help produce quicker decisions and result in better resource allocation and fire management [12]. During wildfires, the information available to the Incident Commander (IC) is critical. Current methods of tracking a fire involve a human pilot flying several miles away from the fire and verbally reporting to the IC what trends they see in the fire. Satellite imaging is also available but is often rendered useless by smoke. In 2012, there were a total of 67,774 fires, destroying 9.3 million acres, and costing over $1.9 billion to suppress in the U.S. alone [6]. Large aircraft can negatively affect the fireline, for example if flown too low (below 1,000 ft), wake vortices from the windtips produce wind gusts which can cause torching and spotting [9].

This paper presents tests of different hotspot monitoring algorithms to gather important information for the Incident Commander (IC) managing the wildfire. This research aims to help improve a UAV's effectiveness in gathering valuable information for the IC. To simulate wildfires, a program developed by the Department of Agriculture and Forest Service is used. FARSITE is a free program used by the U.S. Forest Service, National Park Service, and more specifically ICs, to predict the fire's behavior using data on the topography, weather, wind, moisture, and fuel [8]. FARSITE exports various characteristics of the fire. While our simulation (FLAME) uses fireline intensity data (BTU/ft/s), other fire metrics like flame length and rate of spread also be incorporated. See Fig. 1 for a fireline intensity map of a simulated fire.

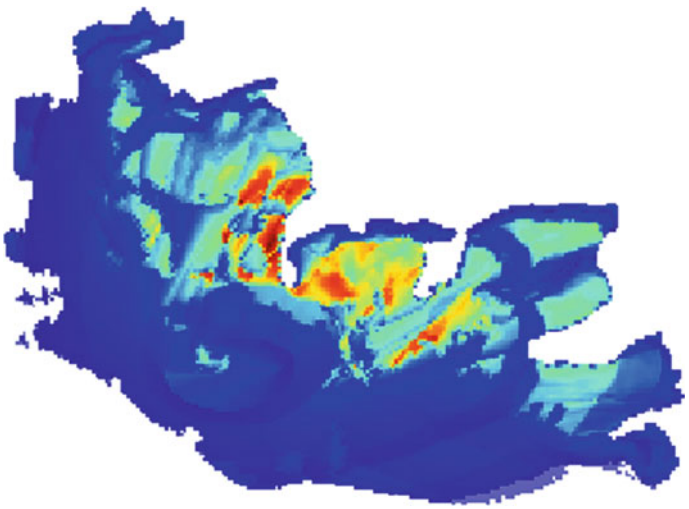


Fig. 1 Wildfire simulation example (*red areas* correspond to hotter areas of the fire). We propose a weighted-greedy algorithm for optimizing the monitoring trajectories of aerial vehicles in wildfire scenarios

Robotic monitoring has become a hot research topic in recent years, due to robots playing a more integral role in collecting environmental data. This has led to a variety of monitoring algorithms [4, 14, 18–20]. Wildfires are highly unpredictable, acting as a unique dynamic frontier. Dynamic monitoring has been explored [2, 18], but fire frontier monitoring is a largely unexplored domain. Our simulator (FLAME) models a dynamic fire frontier and uses techniques like Mini-Batch K-Means Clustering to achieve a similar problem formulation as related monitoring research.

Tracking the most volatile locations on the fireline will give valuable information for the IC. These hotspots will be intelligently monitored by the UAV, using algorithms for minimizing the time hotspots are left unmonitored.

While wildfires were the chosen domain, this research is not limited to wildfires. Similar application domains with dynamic frontiers include: algae blooms, pollution spills, and military battles [16, 22]. These similar domains can also be analyzed using the techniques developed in this paper.

The main novelties of this paper include: (1) a simulation (FLAME) which uses realistic fire modeling software for accurate fire characteristics, (2) a novel fire tracking algorithm which outperforms existing methods, (3) the first investigation into adaptive monitoring of hotspots along a dynamic frontier, and (4) hardware experiments demonstrating the ability to implement our work with existing technology. Taken together, these contributions provide a new approach to the general problem of monitoring dynamic frontiers.

The remainder of this paper is organized as follows. First, we will establish the current state of similar research (Sect. 2). Following that, we will overview the problem and assumptions we made during our investigation (Sect. 3). Next, we describe the simulation and a novel approach to frontier monitoring (Sect. 4). Finally, the algorithm is described in detail in (Sect. 5), and the simulation results are presented (Sect. 6), and the hardware experiments are discussed (Sect. 7).

2 Related Work

Work on autonomous information gathering began with early work in sequential hypothesis testing [23], which focused on determining which experiments could efficiently classify the characteristics of an unknown. This line of research developed into more general approaches and evolved into the field of active perception [1]. Similar insights led to using optimization techniques on robotic information gathering problems, and researchers later developed algorithms for minimizing long-term information uncertainty [4, 5].

Robotic systems are becoming more commonly used as mass data-gathering tools by scientists [7, 10, 18]. Robots are already collecting large datasets on environmental change. Algae blooms, pollution, and other climate variables are application domains for persistent monitoring techniques. Persistent/adaptive monitoring in robotics is currently a growing research topic. Prior work has explored different approaches to monitoring stationary and dynamic feature points. While these adaptive sampling

techniques focus on optimizing uncertainty levels in static [4, 14, 19] and in dynamic environments [18, 20], prior work often focuses on systems in obstacle-free environments. Some research has examined collision avoidance [11, 21], but adaptive sampling along dynamic frontiers remains an ongoing research problem.

In [3], fire frontier tracking was integrated into a simulation for determining UAV tracking accuracy of the fire perimeter. Their UAVs follow a circular path around the fire similar to our baseline. However, our metric is to track the most active parts of the fire. We compare the baseline against our weighted-distance algorithm. Our research presents the first investigation into adaptive monitoring of hotspots along a dynamic frontier. We span the domains of hotspot monitoring and dynamic frontier tracking to evaluate path planning techniques in our FLAME simulator. This line of work allows us to test new algorithms in real-world scenarios.

3 Problem Formulation

We will now formally introduce the problem domain and the assumptions we made. We will also introduce the metric we use to evaluate our algorithm against the baseline.

We assume that GPS and communication between the IC and the vehicle are always available. This means the UAV can always localize itself and never needs to return to the starting location to transfer collected data. We assume the UAV always has the simulated fire frontier in order to find the hotspot locations. Additionally, the UAV is assumed, for comparison purposes, to have unlimited endurance.

Each hotspot has a corresponding time since last tracked by the UAV and the maximum time its been left untracked (ϕ) in the past. The sum of ϕ of all hotspots was chosen as the metric to evaluate the effectiveness of an algorithm. In this paper, fireline intensity is used as the crucial information needed by the IC. The intensity is monitored through the clustering into hotspots, directly relating to the goal of providing the IC with up-to-date information about the fire progression.

$$J(t) = \sum_{i=0}^{hotspots} \phi_i \,, \tag{1}$$

where ϕ is max time untracked.

The goal is to minimize the metric $J(t)$, which corresponds to timely hotspot monitoring, through an optimized trajectory for the UAV.

4 FLAME Simulation

We will now explain our simulation and how we developed each of the different components. Figure 2 should be used as a reference of the state transitions in the simulation. There are two aspects to the weighted-greedy algorithm, picking which hotspot to go to, and how to get there.

Fire data is generated using FARSITE, the wildfire simulator currently used by ICs during wildfire management [8]. The data is exported in the form of time of arrival, and a measurable characteristic of the fire. In this work we use the fireline intensity at each location. As stated above, the task is to minimize the sum of max time untracked (ϕ) over all hotspots. At mission start, the UAV must first find the fire and begin identifying the hotspot regions.

Tracking a hotspot is done by calculating the distance between a previous set of hotspots relative to a new set. To determine when a hotspot moved as the fire progressed, a threshold is implemented. If a hotspot is not within the distance threshold of any previous hotspots, it is then classified as a new hotspot. Even after careful tuning, this approach can still lead to some untracked hotspots where the hotspot existence is too short for any response by the UAV.

To identify hotspots, all points along the frontier with a fireline intensity above a normalized threshold are parsed using a clustering technique called Mini-Batch K-means [15]. K-means clustering was chosen because it directly relates the number of interest points (how active the fire is) to the number of cluster centers (hotspots). The desirable amount of clusters (K) changes as the fire evolves. We actively determine the K value for adaptive hotspot extraction with the following formula. With K as

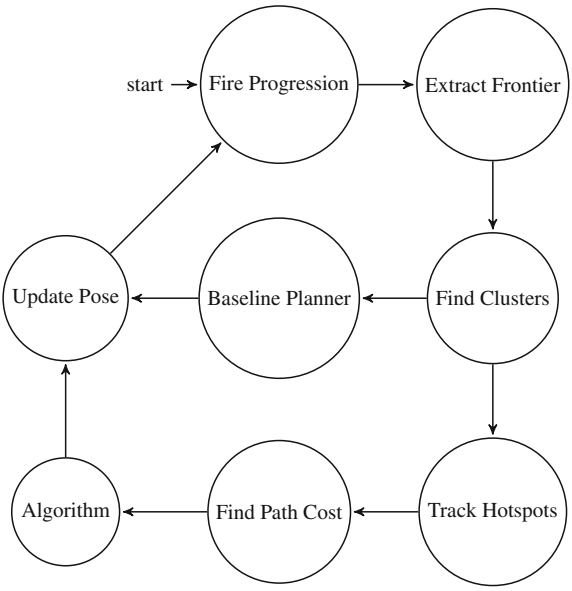


Fig. 2 State diagram of FLAME

the number of centers, and N as the number of interest points,

$$K = \sqrt{N/2} \tag{2}$$

FLAME uses A* path planning for generating paths from the UAV to hotspot locations around the fire. This method works better for estimating path cost over a simple Euclidean distance estimate due to the spherical tendency of the fire spread. Other similar methods where explored to increase efficiency, such as Jump Point Search. Jump Point Search gives respectable speed gains in environments with large open spaces, while the UAVs path remained mostly along the fire frontier. Methods like wall-following could provide faster simulation times, but lack expandability to more complex frontiers, and provide less accurate path costs. Due to the shape of the fire, any benefits of these alternatives were determined to be inconsequential. It was therefore determined to use the A* search algorithm as the UAVs path planner.

A cost map is passed to the A* algorithm, and is generated by applying a blur to the map of the fire up to that point in time and assigning a high cost to areas within the fire. This helps ensure the path generated for the UAV is not within dangerous proximity of the fire, but can still be navigated close enough to monitor the hotspots. The algorithms were tested over seven different fires generated in FARSITE. The baseline and proposed algorithm are described in pseudo code in Algorithms 1 and 2. A state diagram of FLAME is provided in Fig. 2. The algorithm state is weighted, but may be replaced with any tracking algorithm for testing.

5 Algorithms

The proposed algorithm is evaluated against a baseline in the following tests. The following sections will describe each algorithm and how it was implemented in the FLAME simulation. The first monitoring technique described is used as the baseline comparison. It exemplifies current tactics utilized in real world wild fire monitoring, and prior research UAV fire monitoring [3]. This is compared to our proposed approach, a weighted-greedy algorithm that moves to the hotspot that has remained untracked the longest with a tunable parameter of distance from the UAV. Figure 3 should be used as a reference of the difference between the two algorithms behaviors.

5.1 Baseline

Traveling parallel to the dynamic fire frontier is used as a baseline model. Calculating a 90° transformation of the vector from the UAVs current location to the nearest point on the fire frontier gives the travel vector of the UAV. Maximum and minimum distance thresholds are imposed on the UAV so it can then move along the frontier

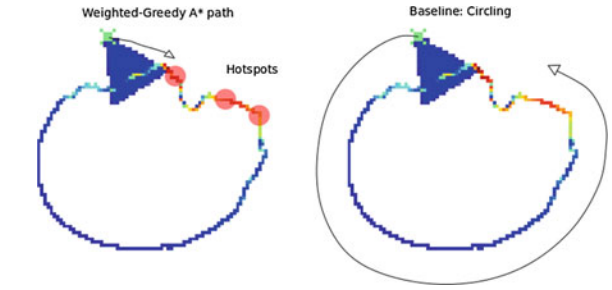


Fig. 3 **a** UAV monitoring the fire using proposed algorithm identifies and tracks the most important part of the fire. **b** UAV monitoring the fire by constant circling will continue regardless of the state of the fire

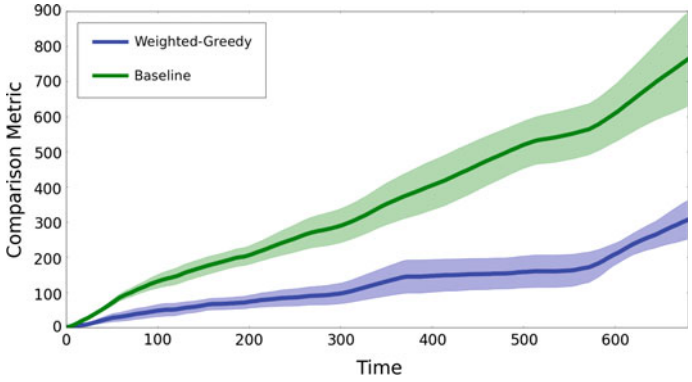
monitoring hotspots while maintaining a safe distance from the fire. We use this as a baseline comparison based on the work of [3].

Algorithm 1 Baseline Algorithm

Inputs: UAV_Location, frontier
for all points in frontier **do**
    points.distance = $\sqrt{(\text{points.x - UAV\_location.x})^2 + (\text{points.y - UAV\_location.y})^2}$
end for
closest_point = min(points.distance)
vector_to_nearest = ([UAV_location.x - closest.point.x], [UAV_location.y - closest.point.y])
normalized_vector = vector_to_nearest / distance_to_nearest
if dist_to_nearest > max_distance_to_fire **then**
    travel_vector = vector_to_nearest
else if dist_to_nearest < min_distance_to_fire **then**
    travel_vector = -vector_to_nearest
else
    travel_vector = (-vector_to_nearest.x, vector_to_nearest.y)
end if
path = travel_vector

Algorithm 2 Weighted Algorithm

Inputs: hotspots{location, time_untracked}, α, UAV_location
for all h in hotspots **do**
    h.path, h.path_cost = ASTAR(h.location, UAV_location)
    h.score = h.time_untracked $- \alpha *$ path_cost(h)
    if hotspot.score > target_hotspot.score **then**
        target_hotspot = h
    end if
end for
path = target_hotspot.path



(a) Weighting parameter α = .5 Hotspot Threshold β = .35

Fig. 4 Wildfire simulation, where the comparison metric is J(t) or the sum of max time untracked of all hotspots. Lower is better. The weighting parameter α is set at 0.5. The normalized threshold β, for a spot along the fire to be considered an interest point, is set to 0.35. A lower β corresponds to more hotspot locations. Error bars are one SEM

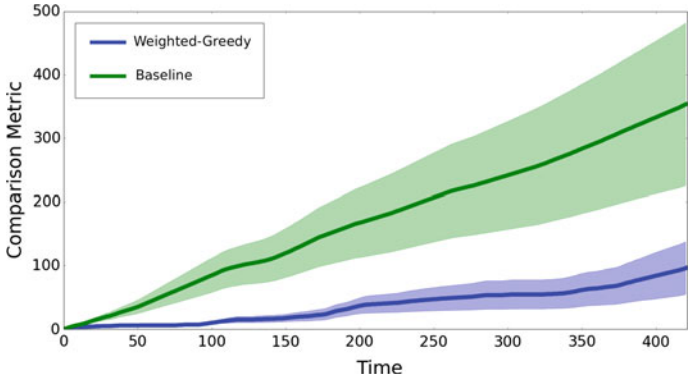
5.2 Weighted-Greedy

The weighted-greedy algorithm checks the time untracked of every live hotspot, calculates the distance to it, and targets the one with the highest score. Unlike the baseline, the weighted-greedy algorithm makes target decisions based on the current state of the hotspots.

This is done using the following formula where $\mathcal{H}$ is the target hotspot, $\mathcal{T}$ is the time untracked of each hotspot and $\mathcal{C}$ is the path cost to each hotspot:

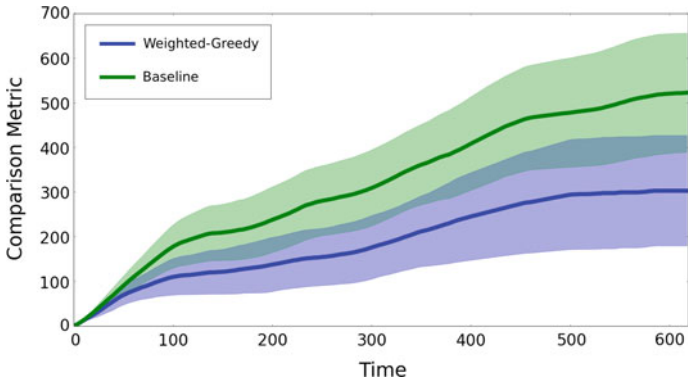
$$\mathcal{H} = \underset{h}{\operatorname{argmin}}\ \mathcal{T}_h - \alpha * \mathcal{C}_h \tag{3}$$

The proposed algorithm accounts for the distance to each hotspot when choosing the targeted hotspot. The weighting factor α is a parameter evaluated in Figs. 4, 5, and 6. The use of a weighting factor addresses some sub-optimality of using just a greedy algorithm. The weighting parameter helps intelligently pick a hotspot that may not be the longest untracked but is closer to the vehicle. A greedy algorithm will immediately move towards the hotspot with the longest time left untracked, disregarding any nearby hotspots that may not have been untracked for nearly quite as long.



(a) Weighting parameter $\alpha = .5$ Hotspot Threshold $\beta = .25$

Fig. 5 Wildfire simulation, where the comparison metric is J(t) or the sum of max time untracked of all hotspots. Lower is better. The weighting parameter α is set at 0.5. The normalized threshold β, for a spot along the fire to be considered an interest point, is set to 0.25. A lower β corresponds to more hotspot locations. Error bars are one SEM



(a) Weighting parameter $\alpha = .5$ Hotspot Threshold $\beta = .45$

Fig. 6 Wildfire simulation, where the comparison metric is J(t) or the sum of max time untracked of all hotspots. Lower is better. The weighting parameter α is set at 0.5. The normalized threshold β, for a spot along the fire to be considered an interest point, is set to 0.45. A lower β corresponds to more hotspot locations. Error bars are one SEM

6 Results

Using our FLAME simulator, we can compare our proposed weighted-greedy approach with traditional methods of monitoring of wildfires. The simulation was run on an Intel i7-4702HQ processor with 8 gigabytes of RAM. The UAV's decision and planning methods took an average of 0.74 s to complete. This is fast enough for a UAV to implement in the field (see Sect. 7).

In comparison to the baseline, the weighted algorithm provided substantial improvement over the course of the trials. The plots in Figs. 4, 5, and 6 show the two monitoring algorithms performance with different parameter settings. As previously discussed, the weighting parameter (α) is multiplied by path cost to the hotspot location. The hotspot cutoff β is the normalized threshold for a spot along the fire to be considered an interest point. This directly affects the total number of hotspots. Tests ran with a lower β will generate a higher number of hotspots for the UAV to track. Time on the X axis begins at first cluster appearance during the simulation. The Y axis shows the results of the comparison metric $J(t)$.

The averaged score over the seven fires are depicted as the bold lines. Around each line, the standard error of the mean is represented by the shading. Figure 4 shows the simulation results with a hotspot threshold $\beta = 0.35$. The plot shows the results with a corresponding α value of 0.5. Our proposed algorithm performs significantly better than currently used approach.

In Fig. 5 the simulation is run with a β equal to 0.25. The β value (0.25) is the lowest used and Fig. 5 shows the performance of both algorithms in an environment with the corresponding large set of hotspots.

Figure 6 depicts the simulation results with a hotspot threshold β at 0.45. This trial uses the highest β (fewest number of hotspots), and shows the plots performance with α value at 0.5. The standard error of the mean (SEM) for both algorithms is significantly higher in this test environment. The results demonstrate the algorithms ability to outperform the baseline in environments with only few clusters, or many clusters. In all cases presented here the proposed algorithm showed significant improvement over traditional wildfire monitoring methods. Our algorithm better tracks the dynamic regions of a dynamic frontier, providing valuable data to better track the frontier.

An interesting characteristic of the frontier monitoring is that it may be simplified into a 1 dimensional problem. Each timestep the UAV must decide between two options, if it wishes to move clockwise or counter-clockwise. It will be worth further investigation into leveraging this characteristic.

7 Hardware Experiments

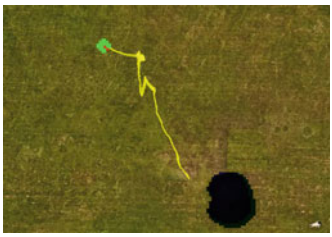
To demonstrate the feasibility of the proposed algorithm, we implemented the algorithm on a live test. To test on hardware we set up the FLAME simulation as a ground station that acted as live satellite data would for a real fire. The algorithm then sent a live stream of coordinates to a UAV to monitor the fire. While a real fire was not used for purpose of this test (for safety reasons), we are able to demonstrate that a UAV can effectively perform these tasks.

We converted FLAME into a ROS package to use the MAVROS plugins [17]. MAVROS acted as a communication bridge between FLAME and the flight controller on the UAV. This allowed us to update the UAVs path in time with the simulation. We used a tethered IRIS+ quadcopter as the platform for these experiments Fig. 7.

(a) A 3d view of our flight log in Google Earth showing the flight distance and trajectories.

(b) Experiment set up, computer running live simulation and IRIS+ flying autonomously (tethered).

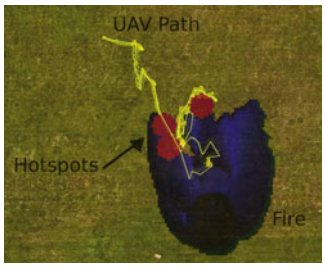
Fig. 7 Experimental setup and flight log results



(a) The UAV first starts off a safe distance away from the fire and must travel to the frontier.

(b) Upon reaching the frontier and identifying a hotspot the UAV stays outside the burn area as it grows.

(c) The UAV moves from one hotspot to another to reduce the time untracked.



(d) Final fire size and flight log of our field experiments.

Fig. 8 Four images demonstrating the algorithm path planning during field tests

We ran the experiment for over 10 min, about half the max flight time of the vehicle. The experiment was performed outdoors in about a 60 ft $\times$ 60 ft area. The simulation coordinates were scaled and transformed to GPS degrees to support sending waypoints. We present the path of the vehicle around the fire in Fig. 8.

The UAV successfully followed the trajectories generated in the simulation to the best locations along the fire to monitor it as it spread. This illustrates our ability to begin introducing robotic monitoring into these dynamic monitoring situations and gather valuable data from it.

8 Conclusion

In this paper, we have introduced FLAME, a simulation developed for testing monitoring techniques on a dynamic frontier, or more specifically a wildfire. The two algorithms tested in the simulation have demonstrated that there is significant benefit in a weighted-greedy over the baseline method of flying around the fire frontier. Using Mini-Batch K-Means Clustering for identifying hotspots, our proposed weighted-greedy algorithm optimized for $J(t)$, the sum of max time untracked of all hotspots. Three different normalized hotspot thresholds (β) (0.25, 0.35, 0.45) were used. Data results showed the weighted-greedy algorithm with significant improvements over the baseline.

These algorithms depend on global knowledge of the fire, or more specifically where the hotspots are. Future work will include implementing a probabilistic model of hotspot locations and studying the exploration/exploitation trade-off for tracking and updating the model. In this paper we assume the UAVs have unlimited flight time. However, the cost of flight with limited endurance is an important factor. Additionally, hotspots are not all equal, and things such as risk to critical areas will need to be considered. Continuation of the project will also focus on implementation of multiple UAVs and the introduction of common fire monitoring challenges, including smoke and adverse weather conditions.

References

1. Bajcsy, R.: Active perception. Proc. IEEE **76**(8), 966–1005 (1988)
2. Bertozzi, A.L., Kemp, M., Marthaler, D.: Determining environmental boundaries: asynchronous communication and physical scales. In: Cooperative Control, pp. 25–42. Springer (2005)
3. Casbeer, D.W., Beard, R., McLain, T., Li, S.M., Mehra, R.K.: Forest fire monitoring with multiple small uavs. In: Proceedings of the American Control Conference 2005, pp. 3530–3535. IEEE (2005)
4. Cassandras, C., Ding, X.C., Lin, X.: An optimal control approach for the persistent monitoring problem. In: 2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), pp. 2907–2912 (2011)
5. Cassandras, C., Lin, X., Ding, X.: An optimal control approach to the multi-agent persistent monitoring problem. IEEE Trans. Autom. Control **58**(4), 947–961 (2013)
6. Center, N.I.F.: Federal fire fighting costs (2015). http://www.nifc.gov/fireInfo/fireInfo_documents/SuppCosts.pdf. Accessed 26 Sep 2014
7. Dunbabin, M., Marques, L.: Robots for environmental monitoring: significant advancements and applications. IEEE Robot. Autom. Mag. **19**(1), 24–39 (2012)
8. Farsite: Fire, fuel and smoke (2014). http://www.firelab.org/project/farsite. Accessed 27 Sep 2014
9. Group, N.W.C.: Wildland fire suppression tactics reference guide (1996). http://www.coloradofirecamp.com/suppression-tactics/suppression-tactics-guide.pdf. Accessed 26 Sep 2014
10. Hollinger, G., Choudhary, S., Qarabaqi, P., Murphy, C., Mitra, U., Sukhatme, G., Stojanovic, M., Singh, H., Hover, F.: Underwater data collection using robotic sensor networks. IEEE J. Sel. Areas Commun. **30**(5), 899–911 (2012)

11. Hollinger, G.A., Sukhatme, G.: Sampling-based motion planning for robotic information gathering. In: Robotics: Science and Systems (2013)
12. Koulas, C.E.: Extracting wildfire characteristics using hyperspectral, lidar, and thermal ir remote sensing systems. In: SPIE Defense, Security, and Sensing, pp. 72,983Q–72,983Q (2009)
13. Kremens, R., Seema, A., Fordham, A., Luisi, D., Nordgren, B., VanGorden, S., Vodacek, A.: Networked, autonomous field-deployable fire sensors. In: Proceedings of the International Wildland Fire Safety Summit (2001)
14. Lan, X., Schwager, M.: Planning periodic persistent monitoring trajectories for sensing robots in gaussian random fields. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 2415–2420 (2013)
15. Lloyd, S.: Least squares quantization in pcm. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
16. Marthaler, D., Bertozzi, A.L.: Tracking environmental level sets with autonomous vehicles. In: Recent developments in cooperative control and optimization, pp. 317–332. Springer (2004)
17. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software, vol. 3, p. 5 (2009)
18. Smith, R.N., Schwager, M., Smith, S.L., Jones, B.H., Rus, D., Sukhatme, G.S.: Persistent ocean monitoring with underwater gliders: adapting sampling resolution. J. Field Robot. **28**(5), 714–741 (2011)
19. Smith, S.L., Rus, D.: Multi-robot monitoring in dynamic environments with guaranteed currency of observations. In: 2010 49th IEEE Conference on Decision and Control (CDC), IEEE, pp. 514–521 (2010)
20. Smith, S.L., Schwager, M., Rus, D.: Persistent robotic tasks: monitoring and sweeping in changing environments. IEEE Trans. Robot. **28**(2) (2012)
21. Soltero, D.E., Smith, S., Rus, D.: Collision avoidance for persistent monitoring in multi-robot systems with intersecting trajectories. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3645–3652 (2011)
22. Susca, S., Bullo, F., Martínez, S.: Monitoring environmental boundaries with a robotic sensor network. IEEE Trans. Control Syst. Technol. **16**(2), 288–296 (2008)
23. Wald, A.: Sequential tests of statistical hypotheses. Ann. Math. Stat. **16**(2), 117–186 (1945)

Part V
Underground

Multi-robot Mapping of Lava Tubes

X. Huang, J. Yang, M. Storrie-Lombardi, G. Lyzenga and C.M. Clark

Abstract Terrestrial planetary bodies such as Mars and the Moon are known to harbor volcanic terrain with enclosed lava tube conduits and caves. The shielding from cosmic radiation that they provide makes them a potentially hospitable habitat for life. This motivates the need to explore such lava tubes and assess their potential as locations for future human outposts. Such exploration will likely be conducted by autonomous mobile robots before humans, and this paper proposes a novel mechanism for constructing maps of lava tubes using a multi-robot platform. A key issue in mapping lava tubes is the presence of fine sand that can be found at the bottom of most tubes, as observed on earth. This fine sand makes robot odometry measurements highly prone to errors. To address this issue, this work leverages the ability of a multi-robot system to measure the relative motion of robots using laser range finders. Mounted on each robot is a 2D laser range finder attached to a servo to enable 3D scanning. The lead robot has an easily recognized target panel that allows the follower robot to measure both the relative distance and orientation between robots. First, these measurements are used to enable 2D (SLAM) of a lava tube. Second, the 3D range measurements are fused with the 2D maps via ICP algorithms to construct full 3D representations. This method of 3D mapping does not require odometry measurements or fine-scale environment features. It was validated in a building hallway system, demonstrating successful loop closure and mapping errors on the order of

X. Huang (✉) · J. Yang · M. Storrie-Lombardi · G. Lyzenga · C.M. Clark
Harvey Mudd College, Claremont, California, CA 91711, USA
e-mail: xhuang@hmc.edu

J. Yang
e-mail: jyang@hmc.edu

M. Storrie-Lombardi
e-mail: mstorrielombardi@hmc.edu

G. Lyzenga
e-mail: lyzenga@hmc.edu

C.M. Clark
e-mail: clark@hmc.edu

D.S. Wettergreen and T.D. Barfoot (eds.), *Field and Service Robotics*,
Springer Tracts in Advanced Robotics 113, DOI 10.1007/978-3-319-27702-8_31

0.63 m over a 79.64 m long loop. Error growth models were determined experimentally that indicate the robot localization errors grow at a rate of 20 mm per meter travelled, although this is also dependent on the relative orientation of robots localizing each other. Finally, the system was deployed in a lava tube located at Pisgah Crater in the Mojave Desert, CA. Data was collected to generate a full 3D map of the lava tube. Comparison with known measurements taken between two ends of the lava tube indicates the mapping errors were on the order of 1.03 m after the robot travelled 32 m.

1 Introduction

It is understood that within our solar system, Mars shares an environment similar in many respects to that of Earth, and it is possible that there might exist traces of life. The surface of Mars is relatively inhospitable and is constantly bombarded by cosmic radiation due to the thin atmosphere and lack of planetary magnetic field. Furthermore, the surface temperature ranges from 215 to 160 K from the equator to the poles. The temperature also fluctuates greatly within a day. Despite these harsh conditions, many scientists predict the existence of a saline groundwater system in the shallow subsurface of the planet, and therefore the subsurface may provide or may have provided a suitable environment for life. NASA's Astrobiology Roadmap objectives include investigating biosignatures in subsurface rocks, modeling subsurface habitable environments, and developing robotic drilling systems to access subsurface environments on Mars [11].

Lava tubes on Mars have gained considerable interest in the astrobiological community because they offer protection from the harsh conditions experienced on the planet's surface. There have been many attempts to characterize these lava tubes to determine the best sites for future exploration and to study the geomicrobiology in lava tubes. To achieve these goals remote-sensing techniques are required [11]. The lava tubes often have many openings, uneven terrain and variation in floor texture. Therefore, while radar instruments have already been used to drill to the subsurface to detect such characteristics, existing sensing methods often lack the resolution necessary to detect exact positions of interest in each individual lava tube.

These challenges motivate the goal of developing autonomous robots that can explore lava tubes and conduct in-situ scientific measurements. Such robots would need to construct 3D maps of the tubes to not only allow the robot to localize in-situ sample measurements with respect to a coordinate frame fixed to the tube, but also to enable the robot to localize itself with respect to the tube and carry out autonomous robot navigation.

Constructing 3D maps with robots has been well studied in the Simultaneous Localization and Mapping (SLAM) community. Many SLAM strategies have used a single robot that fuses odometry and range measurements via filtering algorithms to localize the robot and map the environment [1, 14]. While these methods are reliable, they are limited by the conditions of the exploration environment. The susceptibility

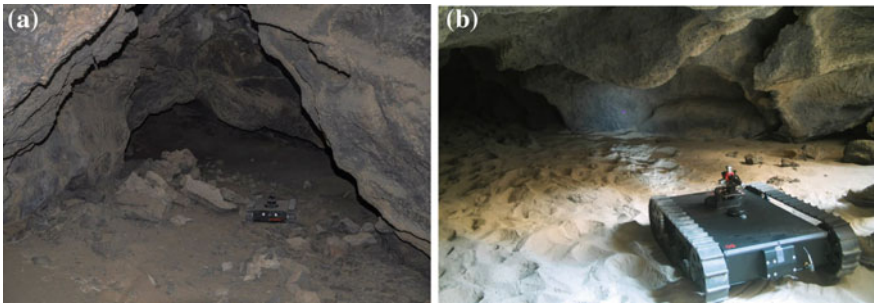


Fig. 1 Image of the Jaguar robot **a** at the entrance of a lava tube **b** on the sandy ground

of the encoder odometry measurements to error resulting from the fine sand found on the lava tube floor further challenges the SLAM problem Fig. 1.

Proposed here is a multi-robot mapping framework that allows robots to cooperatively map lava tubes which (a) have poor odometry measurements due to the fine sand of the tube floor, and (b) lack fine-scale features that reduce dead reckoning errors. Section 2 of this paper presents related work. A three-step solution called *Platoon SLAM* is proposed in Sect. 3, where in the first two steps range finder measurements of the relative distance and bearing-angle orientations between robots are used to update their positions, and in the last step these updated positions are used to seed ICP algorithm queries, that both localize the robot in 2D and construct maps in 3D. Implementation of these techniques are documented in Sect. 4, where results from hallway and lava tube mapping scenarios are presented. Finally, conclusions from these results are drawn in Sect. 5 and possible future work is proposed in Sect. 6.

2 Background

The problem of Simultaneous Localization and Mapping (SLAM) involves constructing a map of an unknown environment while localizing the position of the robot. SLAM is a maturing research area, with work most related to this project including advancements made in the sub-disciplines of 3D SLAM, ICP, 3D mapping in tube like structures, and multi-robot 3D SLAM.

A variety of approaches to 3D mapping in SLAM have been implemented that combine different localization and mapping techniques. Initially, 3D maps were built using multiple 2D scanners with different orientations to construct the 3D map. Thrun et al. [19] used measurements from two laser scanners, oriented perpendicular with respect to each other to form 3D point clouds. However other methods mentioned below give higher resolution of the generated 3D map, including visual SLAM using cameras or 3D range sensing methods are used in autonomous mapping [4, 9, 10, 14, 23, 24]. One popular 3D scanning method uses a pair of cameras with RGB-D cameras in 3D sensing [9]. This method is not well suited for the low lighting