John Ericksen

Candidate

Computer Science

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Melanie E. Moses, Chair

Rafael Fierro, Member

George Matthew Fricke, Member

Kim Linder, Member

Jared Saia, Member

Aerial Robotic Studies of Volcanic CO_2 Emissions

BY

John Ericksen

B.S., Computer Science, Western Washington University 2004

M.S., Computer Science, University of New Mexico, 2017

DISSERTATION

Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

Computer Science

The University of New Mexico

Albuquerque, New Mexico

May, 2025

DEDICATION

To my mother, Lena, who recognized my passion for computers at an early age and encouraged my curiosity. Her unwavering support laid the foundation for my journey in technology. I am forever grateful for the path she set me on.

ACKNOWLEDGMENTS

I want to thank the many people who made this dissertation possible. This research has been supported by so many enthusiastic and generous individuals, truly making it a group effort. First and foremost, I extend my sincere thanks to my advisors Melanie Moses and Matthew Fricke who worked as a team to advise me through this process. Melanie and Matthew's guidance helped me become a better writer, a better scientist, and an accomplished scholar. I could not have done this without their support.

I would also like to acknowledge the members of the Moses Biological Computation Lab, including Humayra, Jake, Sarah, Vanessa, Wayne, Will, Josh, Julie, Akil, and Calvin whose support, ideas, feedback, and camaraderie during countless lab meetings provided continual inspiration and encouragement.

I want to thank my step-father, Marcus Shimazu Takahashi who took my brother and me under his wing when we were young and fostered in us a love of science and engineering. Without his encouragement, I would not be on the path that I am on today.

I want to thank my Wife Crystal Ericksen and Step-Son Isaak Ziff for their unwavering love, patience, and support throughout my Ph.D. journey. You both sacrificed so much to make this dissertation possible, and I am eternally grateful for your partnership in this accomplishment.

I want to thank Jarrett Jones, who was instrumental in the early development of the Dragonfly UAS platform. His mentorship was invaluable in teaching me how to build and pilot quadrotor drones.

I would like to recognize the support of Carter Frost throughout the VolCAN project, both in hardware and software implementations. He was also a great help during multiple field expeditions, enabling our work to move forward smoothly.

I would also like to thank Abir Islam for generously partnering with me in developing the Sketch algorithm. His insights into spatial mathematics and theoretical modeling were crucial in advancing the algorithm design and empirical analysis.

I am indebted to Abhinav Aggarwal, who collaborated with me to develop the LoCUS algorithm, which was a springboard for this work. His perspective, not only on LoCUS but also on other endeavors, was able to uncover novel contributions to Computer Science within my own research.

I would like to thank Tobias Fischer for initiating our collaboration to study volcanoes using Unmanned Aircraft Systemss (UASs), an idea that grew into the VolCAN project and ultimately shaped much of this dissertation. When it came time to locate an active volcano for field research, Tobias was instrumental in making the necessary connections and arrangements. During our expedition to La Palma to study the Tajogaite eruption, Tobias hilariously told me, "You asked me for a plume, so I found you a plume."

I want to thank Rafael Fierro for his direction in electrical engineering, which this dissertation touches on. His expertise in quadrotor design, dynamics, and theory were a great support throughout the project. Each interaction with Rafael left me thinking that I need to talk to Rafael more.

I want to thank Stephanie Forrest who, before my official Ph.D. journey, helped guide my research and develop my skills as an author. Stephanie challenged me to approach my research with scientific rigor, which laid the foundation for this dissertation.

I extend my appreciation to Kevin Aubert for his determined efforts to analyze the Sketch algorithm. His partnership brought out the best in my work and significantly advanced the Sketch empirical analysis.

Finally, I want to thank the management team at Honeywell Federal Manufacturing and Technologies for encouraging my educational journey, from my Master's degree through this Ph.D. In particular, I extend my deepest appreciation to Pam Kissock, David Pilcher, David Cochrell, Virginia Palka, and Kerry Evans. Their unwavering support and guidance made it possible for me to pursue higher education, which included a two year fellowship to work on this research full-time. I could not have achieved any of this without them.

Aerial Robotic Studies of Volcanic CO_2 Emissions

by

John Ericksen

B.S., Computer Science, Western Washington University 2004

M.S., Computer Science, University of New Mexico, 2017

Ph.D, Computer Science, The University of New Mexico, 2025

ABSTRACT

Volcanic systems are inherently complex, involving dynamic interactions among magma flow, gas emissions, and atmospheric dispersion. This dissertation focuses on developing and analyzing autonomous UAS algorithms for efficiently surveying volcanic CO_2 plumes, introducing several novel methods: the LoCUS algorithm, a swarm coordination and self-healing algorithm that supports gradient-based plume tracking, a transect-based technique that employs a 2D Gaussian fit to calculate CO_2 plume flux, and the Sketch algorithm for rapid plume boundary tracing. By treating multiple UAS as a single scientific instrument, these methods leverage swarm algorithms to use in-situ data in ways impossible with individual drones. Validated through simulations and field experiments at sites such as the Valles Caldera supervolcano in New Mexico and the Tajogaite eruption in La Palma, these techniques effectively find plume sources, calculate maximum CO_2 plume flux, and map plume areas, all the while mitigating operational risks. Conducted under the VolCAN project, this research provides powerful tools for volcano monitoring and hazard prediction, with broader implications for studying environmental phenomena.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We live in a world shaped by volcanic activity. Volcanoes are responsible for creating more than 80 percent of the earth's surface including land masses above and below the water. Volcanic activity unlocks nutrients from the earth's core, producing some of the most fertile soil on the planet. Additionally, volcanic systems offer geothermal energy as a clean and abundant energy source. For these and other reasons, humans are drawn to live in volcanic-active regions. However, living in these areas is not without risk. Over the last 500 years, volcanic activity has resulted in the deaths of over a quarter million people due to the unpredictability of eruptions [30]. Therefore, better forecasting of these eruptions is one of the three Grand Challenges recently highlighted by the National Academies because of the significant human risk presented by these eruptions [63].

The challenge of monitoring volcanic activity presents fundamental problems in distributed sensing and spatial computing. A key gas emitted by volcanoes is CO_2, which is degassed by subterranean magma. Increased CO_2 emissions, particularly in ratio to SO_2, is a precursor indicator of increased volcanic activity and a hallmark of an imminent eruption [11, 8, 50, 76]. In contrast to SO_2, which may be remotely sensed using an ultraviolet spectrometer from ground-based or hand-held detectors [74], CO_2 must be sampled in-situ because of the high degree of ambient interference [18]. This requirement for direct sampling in hazardous environments makes volcanic CO_2 monitoring an ideal application for distributed robotic systems. UASs are an ideal collection platform as they are largely expendable and can be quickly positioned within plumes to gather data.

This research advances distributed robotics and spatial computing through the development of novel algorithms for coordinated sensing and mapping, demonstrated here in surveying volcanic CO_2 plumes. The primary contributions advance distributed robotics through novel algorithms for coordinated sensing, which we demonstrate by gathering in-situ CO_2 data. To achieve this, we study and implement autonomous algorithms based on Computer Science theory, including search optimization, flocking dynamics, gradient descent, and asymptotic analysis. These algorithms are thoroughly tested in simulation to ensure their expected behavior in a fully controlled environment. Finally, to bridge the reality gap between simulation and real-world conditions, we validate these algorithms on physical UAS robot platforms in a range of real-world conditions. By testing our algorithms in real-world scenarios, we can ensure that they work effectively in a range of different environments, and we can improve our understanding of how to adapt and optimize them for different use cases.

This research is conducted as part of the VolCAN grant, which focuses on applying UASs to study volcanic plumes to improve our understanding of volcanic behavior. The project focuses on four key areas. First, we need to identify the largest plumes quickly to focus our measurement efforts efficiently. Second, we must detect plume clusters to understand the spatial distribution of emissions. Third, we need to estimate plume areas to understand their extent and dispersion. Fourth, we must find the maximum CO_2 source to locate the primary emission points. These objectives converge on a central goal: determining the maximum CO_2 flux of volcanic sources. Flux, which represents the total amount of CO_2 released and varies over time with magma chamber conditions, is crucial for predicting volcanic activity and hazards.

To achieve these objectives, we develop several key components. The foundation of our work is a custom UAS platform designed for long-endurance flights and reliable CO_2 data collection near volcanic activity. This platform serves as both a testing environment and field deployment system, with validated capabilities for CO_2 detection and geospatial correlation against known ground-truth sources.

Bridging the gap between theory and practice, we implement a comprehensive software suite that operates in both Gazebo physics simulation and on physical UAS platforms. This includes the piloting software to guide the UAS in flight, a dashboard for monitoring and control over the swarm of UASs, and a virtual plume capable of providing similar plume readings to a volcanic CO_2 plume.

Building on this foundation, our algorithmic contributions advance several areas of computer science through novel approaches to distributed sensing. First, we develop the Loss-tolerant Cohesive UAV Swarm (LoCUS) [55] algorithm, which enables multiple UASs to operate as a single cohesive unit while maintaining fault-tolerant swarm behavior. Building on LoCUS, we implement and field-test gradient descent and rasterization algorithms that allow the swarm to efficiently track and map plume concentrations, demonstrating successful autonomous navigation in real-world conditions. We advance plume analysis techniques by developing a Gaussian fit method that accurately calculates CO_2 flux from real volcanic plumes, providing critical data for eruption forecasting. Finally, we implement and analyze the Sketch algorithm for asymptotically bounded plume boundary tracing, providing the first empirical validation of its theoretical performance and establishing its effectiveness for rapid plume characterization.

These advanced techniques collectively transform how we study and understand volcanic behavior. By combining autonomous UAS systems with sophisticated algorithms, we create new capabilities for predicting volcanic hazards and activity. This research not only advances the field of volcanology but also demonstrates the potential of autonomous systems in challenging environmental monitoring applications.

Chapter 2

LoCUS: A multi-robot loss-tolerant algorithm for surveying volcanic plumes

2.1 Publication Notes

Citation: Ericksen, John, et al. ”LOCUS: A multi-robot loss-tolerant algorithm for surveying volcanic plumes.” 2020 Fourth IEEE International Conference on Robotic Computing (IRC). IEEE, 2020.

Publication date: 24 December 2020

Conference: 4th IEEE International Conference on Robotic Computing

Publisher: IEEE

Formatting: The original published text has been preserved as much as possible while still adhering to the formatting requirements of this dissertation.

Data and Software Availability: The code used in this paper is publicly available at https://tinyurl.com/tne7tzu.

Funding: This work was supported by funding from the following: the Department of Energy's Kansas City National Security Campus, operated by Honeywell Federal Manufacturing & Technologies, LLC under contract number DE-NA0002839, a James S. McDonnell Foundation Complex Systems Scholar Award and DARPA award #FA8650-18-C-6898 for funding. We thank the UNM Center for Advanced Research Computing, supported in part by the National Science Foundation, for providing the high performance computing resources used in this work.

2.2 Abstract

Measurement of volcanic CO_2 flux by a drone swarm poses special challenges. Drones must be able to follow gas concentration gradients while tolerating frequent drone loss. We present the LoCUS algorithm as a solution to this problem and prove its robustness. LoCUS relies on swarm coordination and self-healing to solve the task. As a point of contrast we also implement the MoBS algorithm, derived from previously published work, which allows drones to solve the task independently. We compare the effectiveness of these algorithms using drone simulations, and find that LoCUS provides a reliable and efficient solution to the volcano survey problem. Further, the novel data-structures and algorithms under-

pinning LoCUS have application in other areas of fault-tolerant algorithm research.

2.3 Introduction

More than 10% of the world's population live in the destructive zone of volcanoes, and a quarter of a million people have perished in volcanic eruptions in the last 500 years [30]. Volcanoes emit unknown amounts of CO_2 and other climate changing gasses, but only 10 of the approximately 300 currently active volcanoes are characterised by long-term datasets that enable any assessment of temporal CO_2 variability [13]. Measuring volcanic CO_2 flux would enable predictions of eruptions, minimizing loss of life and economic impact, as well as informing our understanding of greenhouse gas-driven climate change.

Satellite remote sensing of CO_2 is infeasible, so sampling is currently performed by ground based sensors or aerial surveys with piloted aircraft [51]. These techniques are costly, dangerous, and produce temporally and spatially coarse measurements. Unpiloted Aerial Vehicles (UAVs) present an emerging solution [99] that reduces risk to volcanologists and has the potential to markedly increase sampling resolution within volcano plumes.

An international team of research universities recently demonstrated that UAVs can feasibly sample CO_2 from an active volcano in Papua New Guinea [97]. We developed the dragonfly drone for this task. The dragonfly is capable of measuring CO_2 in real time and has a flight duration of 1.00 h. However, drone loss was very common. Sudden and violent thermal updraughts, acidic plumes, and rugged cliffs were some of the many conditions that destroyed UAVs. Further, the remoteness of many survey sites and battery life restrictions necessitate brief missions with small swarms. These hazardous and difficult conditions motivate the need for reliable performance and surveillance algorithms that maximize the chance of completing the CO_2 surveillance task even with the loss of drones, short flight times, and small swarm sizes.

A key task for volcano surveillance is to locate the maximum CO_2 flux (max flux) in a dynamic gas plume. We propose the LoCUS algorithm to maintain a spatially dispersed swarm of drones that can simultaneously measure CO_2 concentrations at different locations and communicate those measurements across the entire swarm. We use deductive arguments to prove the loss-tolerance properties of LoCUS, and we test its performance and fault tolerance in simulations. In particular we show LoCUS guarantees that failed drones are replaced within flight-time proportional to the square root of swarm size, while preserving the swarm symmetry essential to efficient gradient following.

We hypothesise that maintaining a dispersed team of robots that can simultaneously measure CO_2 at different spatial locations will provide a better estimate of the CO_2 gradient, allowing fast navigation to the

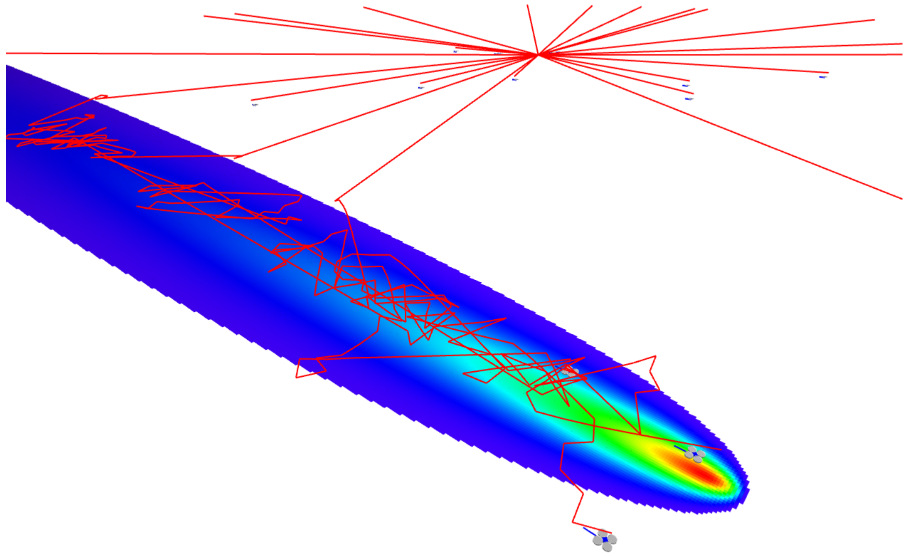


Figure 2.1: **MoBS simulation with 16 drones and a smooth plume.** The red lines trace each drone's independent search for the plume using golden ratio spokes from the center of the arena. After each drone contacts the plume, it switches to a Moth pheromone inspired search algorithm to find the max flux.

CO_2 source. We further hypothesise that the benefit of spatially dispersed measurements outweighs the increased complexity resulting from coordination and self-healing. To test these hypotheses, we develop an alternative approach which allows multiple UAVs to independently search for the maximum CO_2 flux. We compare LoCUS to MoBS, an algorithm that combines a ballistic search algorithm for multiple agents without communication[7][4] and a gas gradient following algorithm for robots inspired by Moth pheromone tracking[146][24].

2.4 Related Work

An algorithm for reliably locating max flux using a remote-sampling robotic-platform requires the following:

1. *Search:* A search pattern to explore an area to make initial contact with the plume.

2. *Plume Gradient Following:* After plume contact is made, the platform follows the gas plume to the source.

3. *Failure Resistance:* The collection of robots needs to respond to failures to maintain a cohesive structure.

Schleich et. al. [127] proposes *searching* an area using a fully-connected swarm of drones and compares this against a random and pheromone-following approach. They find that a fully-connected swarm satisfies base-station connectivity requirements while achieving slightly better survey performance for larger swarm sizes. This motivates LoCUS, as keeping the swarm in contact provides benefits that outweigh the overhead of maintaining swarm connectivity.

Neumann et. al. [106] compares 3 algorithms for *plume gradient following*: the surge-cast algorithm, the Dung Beetle (zig-zag) algorithm, and the pseudo-gradient algorithm using a single robot agent. Through the author's experiments, in both simulation and physical robots, they validate all three algorithms promising for micro UAVs each under different ciricumstances. Our approach uses multiple robots for plume gradient following, with MoBS closely resembling the surge-cast algorithm and LoCUS resempling pseudo-gradient algorithm across the swarm formation.

Chen et. al. [40] apply a Particle Swarm Optimization algorithm to follow a gas plume gradient in an indoor environment. This approach requires full swarm connectivity to communicate global arena information throughout the swarm. This motivates keeping the swarm connected with coordinated movement for gradient descent.

In [35], Cabrita et. al. investigate locating the max flux using Gaussian parameter estimation leveraging a simulated annealing error minimisation approach. They test this algorithm successfully on a swarm of 5 robots. We implement a similar model in MoBS and LoCUS, but we only use the local gradient to navigate the plume in the case of MoBS, and the gradient that spans the swarm's full extent in LoCUS. We use their simple linear fit to determine the direction of the CO_2 gradient.

Flocking algorithms are effective at coordinating movement while being *failure resistant*. Souissi et. al. [132] and Yang et. al. [150] propose leader based approaches for moving a swarm flock while maintaining a given shape and detecting and recovering from failures. Their algorithms keep the swarm together during movement. LoCUS, on the other hand, makes theoretical guarantees about swarm symmetry as drones are lost, given a

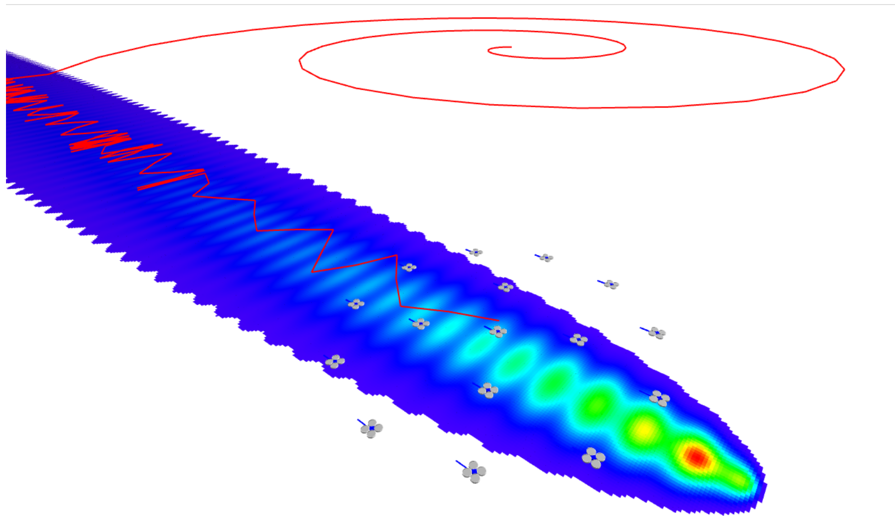


Figure 2.2: **LoCUS simulation with 16 drones and a perturbed plume.** The red lines trace the swarm's Archimedes Spiral search for the plume. After contacting the plume, the swarm follows leverages its simultaneous spatially dispersed measurements to descend the gradient to the max flux.

small collection of drones in close enough proximity that all drones can maintain communication with each other. Our approach could be applied to heal traditional flocking algorithms like the one presented in [143].

Paliotta et. al. [110] present a *plume gradient following* agent based model for three fully networked agents [29][27][23]. We extend this structured plume gradient following approach with LoCUS by increasing the swarm size, tuning agent capabilities to mimic our dragonfly robotic platform, and adding a fault recovery mechanism.

2.5 Derivation and Analysis of LoCUS

The LoCUS algorithm ensures a fully connected swarm with efficient recovery from drone failures. A LoCUS swarm is able to be controlled as a single unit, by directing all members of the swarm at once. We first discuss the basic algorithm assuming no failures, and then discuss how the swarm recovers from drone failures and resumes its mission.

Let N be the total number of autonomous drones in the system. Each drone has a unique ID in $\{1, \ldots, N\}$ and a communication radius $\mathsf{R}_{\mathsf{max}}$ and a safety radius $\mathsf{R}_{\mathsf{min}}$. Each drone can communicate with any drone within $\mathsf{R}_{\mathsf{max}}$ distance, but requires a minimum distance between any two drones in the swarm to be $\mathsf{R}_{\mathsf{min}}$ to avoid collisions.

2.5.1 Balanced Range-Limited Trees

Definition 1 *Given $R_{min}, R_{max} > 0$ and an integer $n > 0$, an (R_{min}, R_{max})-Range-Limited Tree on n nodes is a rooted tree, where the distance between any two nodes is at least R_{min} and at most s. In particular, a maximal (R_{min}, R_{max})-Range-Limited Tree is one in which the distance between the parent node and any of its children is R_{max}. The ratio $\rho = R_{max}/R_{min}$ is the* spread *of this tree.*

As with standard k-ary trees, we can define the height of a Range-Limited Tree $\mathcal{T}$ node in terms of the heights of its children. We define

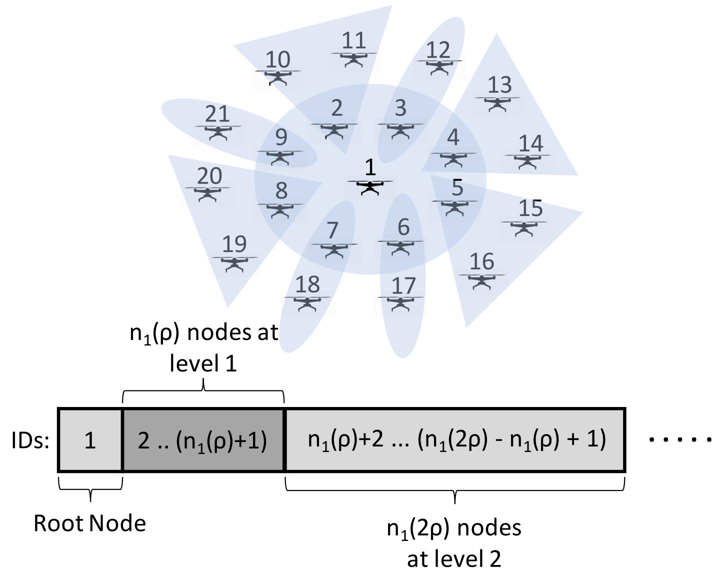


Figure 2.3: **Assignment of nodes to levels based on their IDs.** Since the number of nodes at each level is fixed, the assignment is deterministic and can be computed locally to determine placement in the swarm (see Section 2.5.1). The blue regions denote parent/child communication links.

the height of the root node as zero and then, recursively, the height of $\mathcal{T}$, denoted $\mathsf{height}\,(\mathcal{T})$, as $\mathsf{height}\,(\mathcal{T}) = 1 + \max_i \{\mathsf{height}\,(\mathcal{T}_i)\}$, where the maximum is over the height of all children $\mathcal{T}_i$ of $\mathcal{T}$. Similarly, we define the level of a node as $\mathsf{level}(\mathcal{T}_i) = 1 + \mathsf{level}(\mathcal{T}_i.parent)$, where $\mathcal{T}_i.parent$ is the parent node of $\mathcal{T}_i$. For this recurrence, the root node is defined to be level zero. Thus, the root node has the largest height in the tree but is located at the lowest level.

Definition 2 *Let $\mathcal{T}$ be a Range-Limited Tree. We say that $\mathcal{T}$ is* Balanced *if for every node in $\mathcal{T}$, the difference in the heights between any two of its children is at most one, i.e., for every node $\mathcal{T}_i \in \mathcal{T}$ with children $\mathcal{T}_i^{(1)}, \ldots, \mathcal{T}_i^{(m)}$, it must hold that $|\mathsf{height}\,(\mathcal{T}_i) - \mathsf{height}\,(\mathcal{T}_j)| \leq 1$ for all $i \neq j$.*

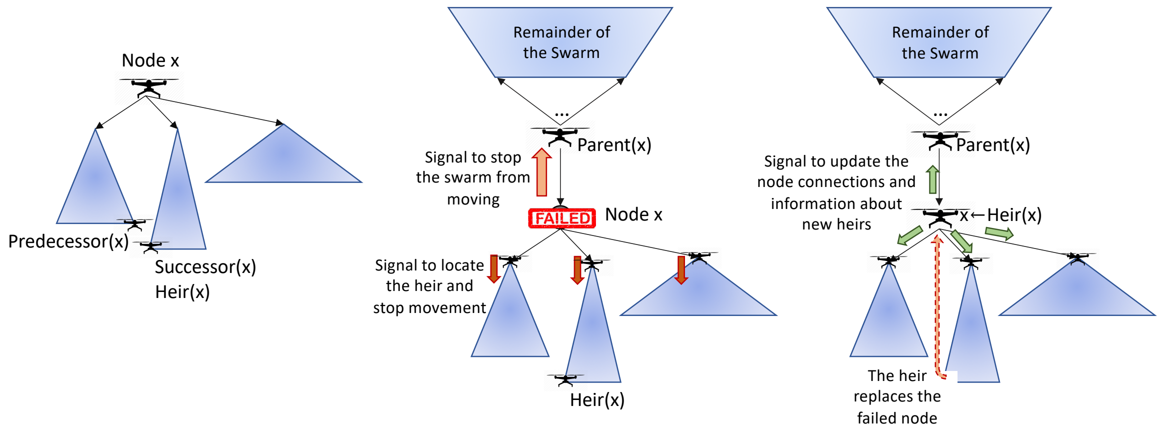


Figure 2.4: **A schematic of single failure recovery in the LoCUS algorithm.** When a node fails, a signal is sent to its parent and children to stop the swarm movement and inform the heir. The heir node then travels to the location of the failed node and the neighboring nodes update their local information.

Each node maintains a pointer to its *heir* in the tree. This is crucial to achieve fault tolerance in LoCUS. We define the heir of a node as its *successor*, if it exists, or its *predecessor*, otherwise. If neither a successor or predecessor exists, the node is a leaf node and the heir is null. To define a successor node, we first define an *inorder* traversal of the tree, denoted $\mathsf{in}(\mathcal{T})$. Let $\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(m)}$ be the children of the root node for $\mathcal{T}$. Then, the inorder traversal of $\mathcal{T}$ prints the IDs of these nodes in the following order (here, $\cdot$ represents the concatenation operator): $\mathsf{in}(\mathcal{T}^{(1)}) \cdots \mathsf{in}(\mathcal{T}^{\lfloor \frac{m}{2} \rfloor}) \cdot \mathrm{ID}(\mathcal{T}) \cdot \mathsf{in}(\mathcal{T}^{\lfloor \frac{m}{2} \rfloor + 1}) \cdots \mathsf{in}(\mathcal{T}^{(m)})$. Note that the inorder traversal is unique for a given tree. We can now define the successor and predecessor of a node.

Definition 3 *Node $\mathcal{T}_j$ is a successor of the node $\mathcal{T}_i$ in the tree $\mathcal{T}$ if $\mathsf{ID}(\mathcal{T}_j)$ immediately follows $\mathsf{ID}(\mathcal{T}_i)$ in the inorder traversal of $\mathcal{T}$. Similarly, we say* *that $\mathcal{T}_j$ is a predecessor of $\mathcal{T}_i$ if $\mathsf{ID}(\mathcal{T}_j)$ immediately comes before $\mathsf{ID}(\mathcal{T}_i)$ in the inorder traversal of $\mathcal{T}$. In all cases, a node is either a leaf, or either its successor or predecessor is a leaf node of tree $\mathcal{T}$.*

Formation Algorithm

The LoCUS algorithm swarm takes the shape of an $(\mathsf{R}_{\mathsf{min}}, \mathsf{R}_{\mathsf{max}})$-Balanced Range-Limited Tree. A balanced Range-Limited Tree layout obtains maximal spatial coverage while maintaining a minimum separation between drones to avoid collisions, and keeps drones within communication range.

Lemma 1 (Number of Nodes at a Given Level) *Let $\mathcal{T}$ be a maximal balanced (r, s)-Range-Limited Tree on N nodes with $\rho = s/r$. Then, the number of nodes at level zero is given as $n_0(\rho) = 1$, and for each $k > 0$, the number of nodes at level k is $n_k(\rho) = \left\lfloor \frac{2\pi}{\sin^{-1}\left(\frac{k}{\rho}\right)} \right\rfloor$. This is the calculation of the whole number of nodes that fit on a circle at radius $s \times k$ separated by distance r.*

Drones deterministically compute their location in the swarm with respect to tree layout. This computation is local to the drones and can be calculated purely by the drone IDs (see Figure 2.3). From Lemma 1, we know that the number of drones at level k is $n_k(\rho)$. Thus, the space of drone IDs can be partitioned based the levels in which the drones belong. For example, the drone with ID 1 is the root node and has level zero, whereas the drones with IDs from 2 to $n_1(\rho) + 1$ all belong to level one.

Each node (besides the root) in the LoCUS tree structure holds a parent reference and list of children. This facilitates bidirectional communication throughout the swarm, as required by the LoCUS algorithm. Parent nodes are calculated by the closest node in the previous layer.

Lemma 2 (Number of Levels) *The number of levels in a maximal balanced (r, s)-Range-Limited Tree on N nodes with $\rho = s/r$ is $O\left(\sqrt{N/\rho}\right)$.*

This is also a bound on the maximum height of the tree and hence, the maximum number of communication hops required for any node in the tree to transmit a message to any other node in the tree. In particular, when ρ is low (i.e. when the communication radius is not too large compared to the safety radius), then the diameter of the tree is $O(\sqrt{N})$, however, when the communication radius is large, say with $\rho = \Omega\left(\frac{N}{\log N}\right)$, then the diameter of this tree becomes $O(\log N)$, which is similar to that of a tree with constant arity. Communication is highly efficient in this case and the latency for transmitting messages is low.

Insertion of Nodes

Always insert at the first available leaf node so that insertion cost is $O(1)$. Insertions do not affect the balance of the tree, since no new levels are created unless the previous level is completely full.

Deletion of Nodes

Replace the deleted node by its heir. If the deleted node is a leaf node, then there is no heir replacement for this node and hence, there is no deletion cost. However, when a node at height $h \geq 1$ fails, then its heir is located at a communication hop distance of $O\left(\sqrt{N/\rho} - h\right)$ from this failed node. Hence, although only $O(1)$ link changes happen upon this replacement, the total number of messages sent is $O\left(\sqrt{N/\rho} - h\right)$.

2.5.2 Handling Drone Failures

The use of the Balanced Range Limited Trees data structure offers the swarm resilience against arbitrarily many crash failures, even when all but one drone remains in the system. We achieve this robustness as follows (see Figure 2.4) – When a node fails, a signal is sent to its parent and children to stop the swarm movement and inform the heir. This signal can be sent out when the node believes it is about to fail, for example when its battery is critically low, or by its parent and children when it fails to respond to a heartbeat. Upon receiving this signal, the heir node travels to the location of the failed node and replaces it in the swarm. Finally, because the tree structure changed, heirs are recalculated on ancestor nodes of the replacement heir node's original leaf location.

Since each drone stores its heir information, it can directly inform the heir drone when it believes a failure is inevitable. For the case when the

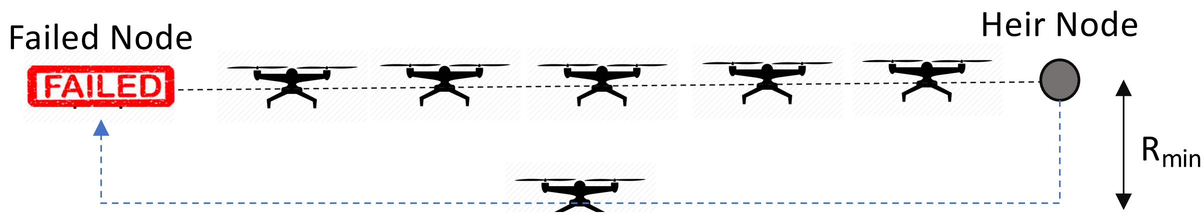


Figure 2.5: **The heir replaces the failed node by flying under the swarm at a safe distance to prevent collisions.**

failure happens without any signal being sent out, the child drones use their *parentHeir* field to contact the heir drone in the swarm for recovery. If there are no child drones, then the failed drone does not require any recovery mechanism since it is already in the last layer of the swarm.

To ensure that drones do not collide with other drones while the swarm is rearranged, the heir drone descends to a distance of $\mathsf{R}_{\mathsf{min}}$ and travels at this height to its destination (see Figure 2.5), at which point it climbs back up to the given elevation. The swarm must stop moving during this recovery phase to avoid complicating communications and movement when the swarm is disconnected.

This heir-based recovery scheme achieves a reformation cost of $O\left(\sqrt{N/\rho}\right)$ – the bound on $\mathsf{height}\,(\mathcal{T})$ – by only inducing local adjustment in the swarm, without disturbing other drones. Note that moving one drone to replace its heir would be approximately equal to drones between the heir and the failing drone shifting up in the tree.

2.5.3 Handling Simultaneous Drone Failures

If both a drone and its heir drone fail at the same time, and the swarm uses the algorithm above to simultaneously recover from both, then it will enter a deadlock scenario. We introduce the following algorithm for handling simultaneous failures with the caveat that it requires global knowledge of the swarm state to execute. A more advanced distributed version of this algorithm that executes without global knowledge is possible, but we leave this analysis and implementation for future work.

Outer-Level First (OLF): In this scheme, we use the fact the failures in outer levels of the swarm cost less to recover than failures on the inner levels. This is because the distance to the heir node is smaller in outer levels. For example, leaf nodes may be removed from the swarm outright without replacement, a node at $\mathsf{height}\left(\mathcal{T}\right)/2$ would require its heir to move $\mathsf{R}_{\mathsf{max}}\,\mathsf{height}\left(\mathcal{T}\right)/2$ distance, and the root node would require its heir to move $\mathsf{R}_{\mathsf{max}}\,\mathsf{height}\left(\mathcal{T}\right)$ distance to replace. Thus, whenever a node gets a failure signal, it first checks to see if there is any existing failure recovery that is active in any of its children. If yes, it waits for those to finish, then proceeds to process the signal from its parent. Concretely, this is implemented by gathering a set of failures across the swarm, and processing them in descending order by $\mathsf{height}\left(\mathcal{T}_i\right)$.

2.6 The MoBS Algorithm and its Implementation

The MoBS algorithm takes a different approach to the max flux problem by allowing each UAV to navigate independently. Each UAV starts at the center of the arena and picks a uniformly random angle between $0.00°$ and $360.00°$ and sets 100 waypoints in $1.00\,\mathrm{m}$ increments from the center in that direction to produce spokes to search the arena. At each waypoint the UAVs collects a gas plume sample and reacts accordingly. The UAV continues to follow the spoke waypoints if a reading of less than 0.005. Otherwise, the UAV changes strategies into the moth-pheromone chemotaxis algorithm inspired by [146]. Subsequent spokes are build by adding $2\pi/\phi$ rad to the previous spoke angle where ϕ is the golden ratio 1.618 that has been shown to search best given no communication amongst members of the swarm [7].

The moth-pheromone chemotaxis algorithm compares the gas reading at the current time step against the previous time step and determines if the signal has increased or decreased. If the signal increased then the drone continues moving in the same direction. If the signal stays the same or decreases then the drone moves in a new uniformly random direction. A zero signal detected for greater than 4 time steps reverts the drone back to continue the golden ratio driven spoke search algorithm. Because there is never any communication among UAV in MoBS, failed UAV stop collecting samples but have no impact on other UAV.

2.7 Experimental Methods

We measure performance of both algorithms for a range of swarm sizes and failure scenarios in simulation. Given the practical limitation of battery life on flight time, our primarily interest is minimizing the time to find the max CO_2 flux. We halt the simulation when the max flux is found (a drone samples within 1.00 m of the max flux location), if the entire swarm is in a failed state, or when 17.30 h of simulation time has passed (10^6 time steps).

We implement the LoCUS and MoBS algorithms in Autonomous Robots Go Swarming (ARGoS) [112].[1] Autonomous Robots Go Swarming (ARGoS) [113] is a C++ and Lua based physics multi-robot simulator and is suitable for proof-of-concept simulations, while preserving realistic physical dynamics with the DYN3D physics engine. We use ARGoS to simulate Spiri UAVs (Pleiades Robotics Inc) including 3 dimensional locality (GPS) inputs and go-to coordinate capabilities. Additionally, we are able to command N drones in the simulation. These capabilities make ARGoS a natural fit for experimental investigation of LoCUS and MoBS.

The gas plume is modeled in ARGoS as a simple two dimensional slice of a Gaussian plume [135] with a source max flux location (x and y),

[1] Implementation source code can be found at https://tinyurl.com/tne7tzu

stack height ($H = 10.00\,\mathrm{m}$), wind speed ($u = 50.00\,\mathrm{m/s}$), emission rate ($Q = 2.00\,\mathrm{kg/s}$), and diffusion rate ($K = 1.00\,\mathrm{kg/s}$):

$$\textsc{unperturbed}(x, y) = \frac{Q}{2\pi K x} \exp\left(-\frac{u(y^2 + H^2)}{4Kx}\right) \tag{2.1}$$

The source of the plume is located at a uniformly random location in the simulation within 100.00 m of the UAV take-off location. Each UAV may detect the gas concentration at its given coordinate as a floating point value between 0 (low) and 1 (high) gas concentration signals. To limit the experimental variance, we only vary the location of the plume and not the shape, intensity, or rotation of the plume. We test the algorithms against the smooth plume described in (6.18) and a perturbed version of the plume designed to make following the gradient more realistic and challenging:

$$\textsc{perturbed}(x, y) = (0.8 + 0.2\sin(4x))\ \textsc{unperturbed}(x, y) \tag{2.2}$$

Our two failure models in these experiments are motivated by flying a swarm of UAVs to gather volcano gas CO_2 emission data.

Generic Failures: To represent a UAV battery failure, crashes, or other miscellaneous failures that increase in likelihood as flight time increases, we use a uniform failure probability per drone per time step given by

$p_f > 0$, which depends on the number of drones existing in the system at time t.

In-Plume Failures: We use the gas plume emissions reading r at time t to drive the probability of failure on each drone given by $p_f r > 0$. This models the higher probability of failure as corrosive gases or temperatures associated with more concentrated volcano gas emissions are encountered.

Drone failure is represented by a boolean flag on the drone controller that, if enabled, stops the drone from moving or receiving further waypoints from its parent. Once a drone fails, it is never recovered.

2.7.1 Implementation of LoCUS

The LoCUS algorithm arranges members of the swarm by distributing each drone through space using specified $\mathsf{R}_{\mathsf{min}}$ and $\mathsf{R}_{\mathsf{max}}$. The unique ID of each member of the swarm allows a unique 2D location offset from the central root node to be calculated. The drones are distributed in a plane by each offset using a constant height of 10.00 m. Each drone's parent is assigned by finding the closest drone in the previous shell of the swarm. This parent/child relationship constructs the data structure pivotal to maintaining communication throughout the swarm.

We implement a recursive algorithm to distribute navigation waypoints by communicating them from the root drone down through its children, to its children's children, and so on. When waypoints are distributed, the swarm offset location for each drone is added to the waypoint to ensure that the swarm maintains $\mathsf{R}_{\mathsf{min}}$ and $\mathsf{R}_{\mathsf{max}}$.

At takeoff, the root LoCUS drone is given the initial starting position. Using the recursive waypoint distribution, this initial starting position waypoint directs the swarm to assemble the shell structure exhibited by Figure 2.3.

To make initial contact with the plume, the swarm is directed from the root to follow the Archimedes' spiral. For coordinated swarm search, the Archimedes' spiral has been shown to find targets faster than a spoke algorithm [5]. This search pattern is created by building waypoints along the spiral. Each waypoint is calculated to space the arms of the spiral by the radius of the largest full swarm shell and an incremented angle. Using the radius of the swarm ensures that we have full coverage of the simulation arena.

After a waypoint is reached, a plume gas reading is sampled from each drone and communicated via the tree structure up to the root drone where the readings (*val*) and associated gps coordinates (x, y) are aggregated into the *uav* array. The aggregated data is input into matrix and vector form and fit with a slope (b) using linear regression in the form $Ab+\epsilon = y$ by minimizing ϵ through least squares approximation provided by the Eigen C++ library [85]:

$$
\underbrace{\begin{bmatrix} 1 & uav[1].x & uav[1].y \\ 1 & uav[2].x & uav[2].y \\ \vdots & \vdots & \vdots \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} b[0] \\ b[1] \\ b[2] \end{bmatrix}}_{b} + \underbrace{\begin{bmatrix} \epsilon[0] \\ \epsilon[1] \\ \vdots \end{bmatrix}}_{\epsilon} = \underbrace{\begin{bmatrix} uav[1].val \\ uav[2].val \\ \vdots \end{bmatrix}}_{y} \tag{2.3}
$$

The slope of this linear fit $(b[1], b[2])$ is used to provide a normal vector to direct the swarm to perform a gradient descent in the direction of the highest plume signal. If a zero magnitude linear slope is found, then the swarm continues to follow the Archimedes spiral.

Failures are handled as follows. First, once a waypoint is reached, failures in the swarm are queried for from the root. This is a recursive call, similar to the waypoint distribution, to gather a set of the failed status of the entire connected swarm. For these experiments and for simplicity, we implement the Outer-Level First (OLF) scheme failure recovery model. This scheme requires global knowledge of the swarm as it uses failed drones to determine the status of their children. We then proceed to heal the swarm as outlined in 2.5.2 Handling Failures remove each of the failed drones and replace them by their heir in order, waiting for each heir to take the place of the failed drone before proceeding to the next failed drone. After the replacement, heirs of all ancestors up to the root are recalculated to take into account the change in the swarm. Of course, if a failed drone is found to have no heir (a leaf drone) then they are removed from the swarm without replacement.

Once all failures are processed a swarm re-balance is executed to ensure a consistent minimum radius to the swarm. This iteratively removes leaf drones from the deepest branches of the tree and inserts them into the shallowest branches of the tree. The root node executes this operation until $\text{height}_{max} - \text{height}_{min} \leq 1$. Once there are no more failures in the swarm and the swarm is re-balanced, then the next waypoint is calculated and the swarm movement continues.

We observed corner-case scenarios where the swarm oscillated between two points, never moving towards the max flux location. To resolve this, we added both a 0.10 m random offset and a uniformly random rotation to the swarm between 0.00° and 45.00° at each waypoint. This randomization strategy allowed to swarm to exit these oscillating corner cases.

2.7.2 Experimental Setup

The experimental factors we explore in simulation are swarm size, whether the plume gradient is smooth or perturbed, the failure rate, and whether the failure rate increases with gas concentration. For the LoCUS algorithm, we set $\mathsf{R}_{\mathsf{min}} = 3.00\,\text{m}$ and $\mathsf{R}_{\mathsf{max}} = 3.00\,\text{m}$. The response variables are whether the max flux was found, the elapsed time before encountering the plume, and the total time taken to find the max flux.

To compare LoCUS and MoBS we perform the following four experiments. In experiment 1, we compare the time to find max flux of the smooth plume for LoCUS and MoBS in 100 trials with both failure probabilities set to 0 by varying the swarm size from 2 to 20 UAVs. In experiment 2, we duplicate experiment 1 using the perturbed plume. These first two experiments are designed to compare the times to encounter the plume and navigate to the maximum flux of LoCUS and MoBS without failures. In experiment 3, we vary the generic failure probability from 10^{-1} to 10^{-6}, set the in-plume failure probability to 0 (so that the probability of failure is the same inside and outside of the plume), and use the smooth plume over 100 trials with a swarm size of 20. In experiment 4, we duplicate experiment 3 but vary the in-plume failure probability from 10^{-1} to 10^{-6} and set the generic failure probability to 0. The last two experiments measure the impact of failures on LoCUS and MoBS. In each of these experiments we compare the performance of LoCUS with healing enabled and disabled. This enables us to assess whether maintaining symmetry though healing is worth the time taken to repair the swarm.

2.8 Experimental Results

2.8.1 Experiment 1: Unperturbed Navigation

Experiment 1 compares the time to reach max flux of the unperturbed in swarms of 5 to 20 UAV for LoCUS and MoBS depicted in Figure 2.6

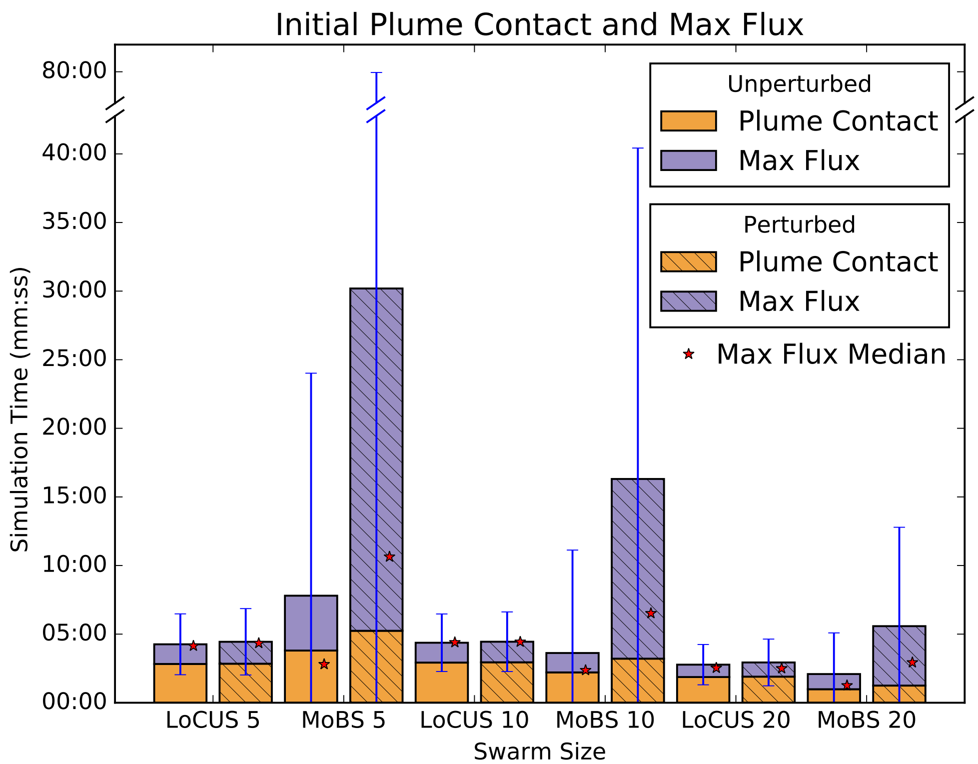


Figure 2.6: **Time to find the max flux (top, purple) and initial plume contact (bottom, orange) for LoCUS and MoBS** for swarm sizes 5, 10, and 20 in the unperturbed and perturbed plumes. Stars show the median time to max flux and error bars are one standard deviation centered at the mean.

with the unhashed bars. We find that, for smaller swarm sizes (up to the 5 UAV shown), the average time and standard deviation to plume contact and max flux for LoCUS is significantly smaller than MoBS. For larger swarm sizes (10, 20), MoBS reaches plume contact on average faster. LoCUS navigates from plume contact to max flux in about the same time as MoBS, but LoCUS has less variance in time to achieve both plume contact and max flux.

2.8.2 Experiment 2: Perturbed Navigation

Experiment 2 extends experiment 1 using the perturbed plume depicted in Figure 2.6 with the hashed bars. We find that the difference in plume dynamics significantly increases the average time to max flux for the MoBS algorithm, but the LoCUS time to max flux remains short. This increases the average time to max flux for MoBS so that is slower than LoCUS for all swarm sizes tested. Additionally, the standard deviation for MoBS is much larger than that of LoCUS which is partly driven by several outliers that lasted up to 344, 124, and 31 minutes for 5, 10 and 20 UAVs respectively. These times risk failure for the drones to return given the UAV battery capacity.

2.8.3 Experiment 3: Generic Failure Effects

Experiment 3 includes generic failure probabilities from 10^{-1} to 10^{-6} including a specialized version of LoCUS that does not heal from failures as depicted in Figure 2.7. We find that, with generic failures, LoCUS and MoBS both respond similarly to the failure probability by beginning to unsuccessfully complete the max flux location task between the probability of failures of 10^{-4} and 10^{-3}. This is contrasted against the LoCUS without healing that responds much earlier to the probability of failure at about 10^{-5}.

2.8.4 Experiment 4: In-Plume Failure Effects

Experiment 4 extends experiment 3 using the in-plume failure model. With in-plume failures, LoCUS and MoBS both begin to fail to complete the max flux location task at failure rates of 10^{-2}. This is contrasted against the LoCUS without healing that fails to complete the task with much lower UAV failure probabilities of about 10^{-4}.

2.9 Discussion

LoCUS provides a failure-tolerant structure for exploring and pinpointing the max flux location of a CO_2 plume. LoCUS guarantees that a group of drones can communicate to each other simultaneous spatially

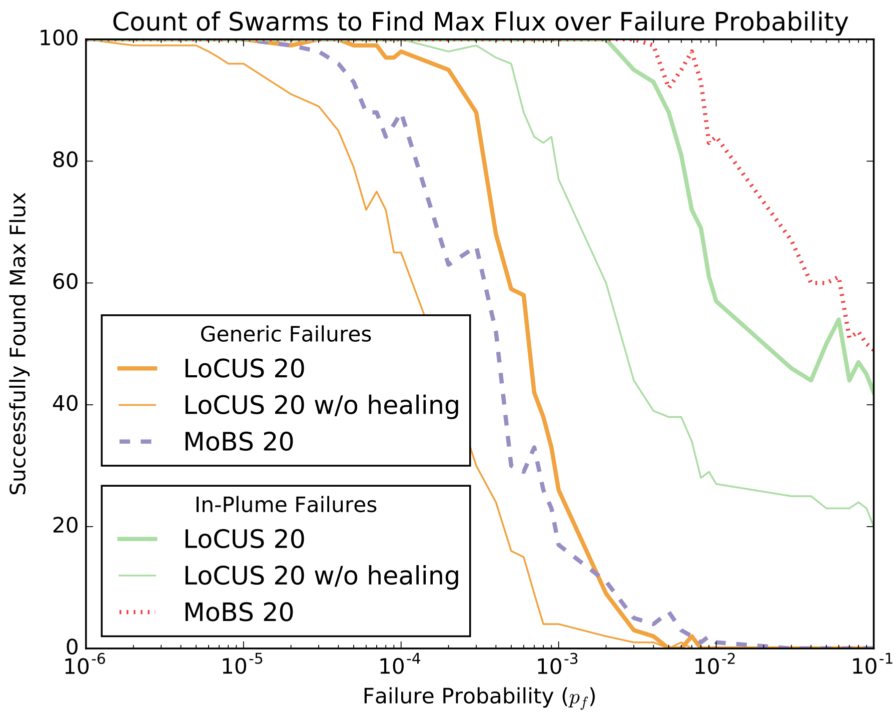


Figure 2.7: **Success rates in 100 trials with generic and in-plume failures with 20 drones.** The left thin solid lines are LoCUS with healing disabled and the right thick solid lines are LoCUS with healing enabled. The two left orange thin and thick solid lines are LoCUS with generic failures, while the green right solid lines are LoCUS with in-plume failures. MoBS success counts are graphed using dashed and dotted lines – the left dashed line is MoBS with generic failures and the right dotted line is MoBS with in-plume failures. This shows how significant healing is to LoCUS successfully completing the max flux location task.

dispersed measurements which can be used to calculate a gas gradient better than an individual drone. This is particularly useful for finding the location of maximum flux in a perturbed plume such as those produced by volcanos in dynamic environments. LoCUS provides a way re-form the swarm given the inevitable failure of drones in hazardous conditions present when monitoring gas efflux from volcanoes. We compare LoCUS with the fully dispersed MoBS algorithm and show in experiments 1 and 2 that the LoCUS algorithm is able to find the max flux of a plume, both smooth and perturbed, at least as fast as the MoBS algorithm in expectation, but with substantially smaller variation. The better worst-case performance of LoCUS is important given time limits imposed by battery life. Additionally, the LoCUS algorithm is able to find the max flux faster than MoBS after initial plume contact, particularly in a perturbed plume simulations.

For large swarms, the MoBS algorithm makes initial contact with the plume faster than LoCUS on average. The superior performance of MoBS at finding the plume, and LoCUS of finding the source once in a plume, suggests an approach that combines the best of both algorithms. For large swarms, we may perform the initial search for the plume using the more dispersed golden spoke algorithm used in MoBS. Then, when contact is made, a LoCUS structured formation can leverage nearby drones to perform gradient descent informed by communication among drones. Future work can explore the benefits of a fully dispersed set of spokes, with the first drone that contacts the plume calling nearby UAV to join together once the plume is found. Alternatively, sufficiently many UAV could be divided into multiple small LoCUS sub-swarms to use spoke search to contact the plume and LoCUS enabled gradient descent by each independent sub-swarm once it contacts the plume.

LoCUS relies heavily on its loss recovery model in order to maintain communication between spatially dispersed drones to perform a more robust gradient descent once a plume has been found (see the red lines within the plume in Figure 2). The loss recovery model allows the swarm to reorganise once a failure has been detected and continue to rely on receiving CO_2 measurements from multiple locations. In practice, we observed LoCUS successfully locating max flux with failures in a majority of the swarm, even down to a single remaining drone. We also observed that the loss recovery time is so fast that it is dominated by the time to encounter the plume and time to max flux. Thus, the time to recover the swarm formation is worth the superior gradient following performance provided by having spatially dispersed measurements. In experiments 3 and 4 we show that self-healing is critical to the success of LoCUS gradient following.

In comparison to MoBS, LoCUS is especially vulnerable to the in-plume failures. This is (ironically) because LoCUS brings the entire swarm into the plume and quickly closer to the source, putting swarm members in jeopardy due to more in-plume failures as the source is approached near the source of volcano efflux. In contrast, some of the UAV in MoBS spend more time out of the plume, making them less susceptible to in-plume failures. The results in experiment 4 show that even in the worst case for LoCUS, it can leverage the healing algorithm to mitigate this problem to complete the max flux task nearly as often as MoBS.

Being able to reliably and quickly determine the max CO_2 flux with drones that are limited to a maximum 1.00 h flight time, with practical swarm sizes for transportation to remote and hazardous regions, and are tolerant of drone loss is critical to the study of volcano behaviour. With LoCUS, we have demonstrated an algorithm that solves the CO_2 max flux task faster than, and approximately as reliably as a more dispersed approach.

2.10 Acknowledgements

We thank the UNM Vice President for Research, the Department of Energy's Kansas City National Security Campus, operated by Honeywell Federal Manufacturing & Technologies, LLC under contract number DE-NA0002839, a James S. McDonnell Foundation Complex Systems Scholar Award and DARPA award #FA8650-18-C-6898 for funding. We thank the UNM Center for Advanced Research Computing, supported in part by the National Science Foundation, for providing the high performance computing resources used in this work.

Chapter 3

Aerial Survey Robotics in Extreme Environments: Mapping Volcanic CO_2 Emissions with Flocking UAVs

3.1 Publication Notes

Citation: Ericksen, John, et al. "Aerial survey robotics in extreme environments: Mapping volcanic co2 emissions with flocking uavs." Frontiers in Control Engineering 3 (2022): 836720.

Publication date: 15 March 2022

Journal: Frontiers in Control Engineering

Publisher: Frontiers Media SA

Formatting: The original published text has been preserved as much as possible while still adhering to the formatting requirements of this dissertation.

Data and Software Availability: The code used in this paper is publicly available at

https://github.com/BCLab-UNM/dragonfly-dashboard

https://github.com/BCLab-UNM/dragonfly-controller

https://github.com/BCLab-UNM/dragonfly-sim

Funding: This work was supported by funding from the following: the Department of Energy's Kansas City National Security Campus, operated by Honeywell Federal Manufacturing & Technologies, LLC under contract number DE-NA0002839. GMF, SN, TF, RF, and MM support provided by the VolCAN project under National Science Foundation grant 2024520. Support was also provided by a Google CSR award. This study received funding from Google and Honeywell Federal Manufacturing & Technologies, LLC.

3.2 Abstract

We present methods for autonomous collaborative surveying of volcanic CO_2 emissions using aerial robots. CO_2 is a useful predictor of volcanic eruptions and an influential greenhouse gas. However, current CO_2 mapping methods are hazardous and inefficient, as a result, only a small fraction of CO_2 emitting volcanoes have been surveyed. We develop algorithms and a platform to measure volcanic CO_2 emissions. The Dragonfly

UAV platform is capable of long-duration CO_2 collection flights in harsh environments. We implement two survey algorithms on teams of Dragonfly robots and demonstrate that they effectively map gas emissions and locate the highest gas concentrations. Our experiments culminate in a successful field test of collaborative rasterization and gradient descent algorithms in a challenging real-world environment at the edge of the Valles Caldera supervolcano[1]. Both algorithms treat multiple flocking UAVs as a distributed flexible instrument. Simultaneous sensing in multiple UAVs gives scientists greater confidence in estimates of gas concentrations and the locations of sources of those emissions. These methods are also applicable to a range of other airborne concentration mapping tasks, such as pipeline leak detection and contaminant localization.

3.3 Introduction

Distributed mobile sensing has many application areas, such as monitoring of industrial gas leaks, hazardous material releases, and agricultural monitoring [122, 81, 116]. Often the materials we are interested in sensing can only be directly sampled, as the signal of CO_2 emissions relative to background is low. Remote sensing methods such as satellite imaging are capable of measuring total column integrated CO_2 on a global scale, but specific eruptions and volcanic plumes must be spatially and temporally

[1]Located in Jemez Springs, NM, USA

r

Figure 3.1: **Flocking Dragonfly UAVs** in formation. The flocking Dragonfly UAVs are used to survey volcanic CO_2 plumes.

targeted in order to capture events [89]. Atmospheric levels of CO_2 prevent accurate satellite imaging and remote laser methods require bulky equipment and have unrealistic line-of-site requirements. But, relatively small instruments exist that can make very accurate point-measurements of CO_2. This requires that the measurement instrument be moved through the area of interest. In the case of volcanic emissions, this has been done by hand-carrying the instruments into dangerous locations or by human-piloted aircraft flying through hazardous volcanic plumes. Ground surveys, in addition to the risk involved, are biased by surveyors' inability to survey areas of unstable rock, sheer cliffs, scalding mud-pots, or (without specialized breathing equipment) areas with poisonous gas.

Several groups recently used remote-piloted aircraft to measure volcanic CO_2 at Manam volcano [96]. That work provided new insights into volcanic CO_2 emissions but was hampered by the challenges of remote-piloting from great distances, with limited visibility, and under extremely hazardous conditions. Only a small subset of the drones deployed by the various teams involved was able to reach Manam volcano's plume, and only one drone survived the expedition.

Here we present the *first autonomous* surveys of volcanic CO_2. Autonomous UAVs are not restricted by line-of-sight or radio communication limitations, are not subject to hazardous ground conditions, and are immune to most poisonous gases. Autonomous UAVs can therefore survey volcanic CO_2 more effectively than human-piloted drones or ground-based surveys. Additionally, autonomous drones can coordinate their flight and sensor readings and make decisions based on those readings in real-time. Autonomy allows a team of UAVs, each equipped with a point-source measurement device, to become a much larger physically disconnected and therefore re-configurable sensor. In our case, three UAVs are required to map the CO_2 gradient fields required to localize CO_2 sources.

We designed, built, and field-tested a small swarm of UAVs called the Volcano Co-robot with Adaptive Natural algorithms (VolCAN) swarm. The VolCAN swarm executes a variety of surveillance algorithms to estimate the gas concentrations critical to volcanic eruption forecasting. The VolCAN swarm also implements a flocking algorithm for gradient descent to navigate to the locations where CO_2 is emitted from the ground. We test the swarm in simulation, in a hybrid field-simulation experiment in an open field, and ultimately perform multi-UAV atmospheric CO_2 emission surveys at the Valles Caldera supervolcano in New Mexico.

Volcanic Emissions

Worldwide, there are over 50 volcanic eruptions each year. More than 500.00 volcanoes are thought to be atmospheric CO_2 sources, yet less than 5.00% of those volcanoes have been surveyed [64]. Changes in the ratio of CO_2 to SO_2 from gas-emitting fumaroles have been observed to precede explosive volcanic eruptions [10, 133], highlighting the potential of real-time gas measurements for eruption forecasting. Better forecasting of these eruptions is one of the three Grand Challenges recently highlighted by the National Academies because forecasting eruptions can save lives and mitigate volcanic hazards [102]. Though dwarfed by anthropogenic emissions, volcanic CO_2 flux is also important to a complete understanding of global volatile budget. [63].

Volcano surveys are hampered by the difficulty and danger of sampling gases in and around active craters. Volcanic CO_2 emissions can only be measured remotely by satellite when a satellite orbit is capable of capturing a specific location during an event. The NASA Orbital Climate Observatory has a 16-day repeat cycle and a narrow sampling width [44], making targeting specific eruptions challenging. Ground-based remote sensing involves bulky instruments which are costly and difficult to deploy in remote areas [12]. Volcanologists, therefore, use hand-held detectors to gather point-source measurements by collecting and analyzing CO_2 concentration in-situ [52]. This is currently accomplished either by aircraft or by ground surveys [41], both of which are hazardous and inefficient. Our driving mission is to remove the human from these dangerous conditions while giving Volcanologists this critical data promptly.

Environmental sensing by UAVs

We designed and built the Dragonfly UAV as shown in Figure 3.1 as the VolCAN swarm hardware platform. The Dragonfly is designed to meet the requirements to survey active volcanoes in real-world conditions. These requirements are informed by our experiences surveying volcanoes with manually piloted UAVs [96]. Currently, the state-of-the-art for volcanic UAV surveys is manually piloted using a single, typically combustion-powered, UAVs [87]. Combustion engines introduce organic CO_2 into measurements which can be a source of significant error [66]. Piloted flights introduce the possibility of human error and does not leverage the collected data in real-time. Our objective is to autonomously fly multiple UAVs to map CO_2 gas concentrations automatically and act as a single collaborative instrument capable of measuring multiple CO_2 reading point-sources at a time. This technique provides redundancy in gas readings, and the ability to calculate a gas concentration gradient. This follows our previous work where we developed and analyzed the LoCUS algorithm.

Our work adds to the rapidly growing literature of environmental chemical sensing with small UAVs outlined by [32], spurred by decreasing costs of chemical sensors and commercially-available drones. For a motion planning approach using chemical-sensing drones see [28]. Nano drone chemical-sensing approaches have been demonstrated [31, 17]. However short flight times make them impractical for the larger-scale volcano surveys we target. Our Gradient following technique is similar to one demonstrated by [3] in aquatic environments for collaboratively mapping a lake boundary using 3 aquatic drones.

Autonomous robotic systems are becoming more resilient and capable of performing monitoring tasks in degraded and hazardous environments [42]. Applications include volcano monitoring [148], subterranean exploration [45], mapping mines [145], and nuclear facilities [141], surveying penguin colonies [131], and disaster relief operations [138], to name just a few. The underlying requirement across all these examples is one we share; to take the human out of harm's way, putting the risk on the expendable robotic hardware.

UAVs are an attractive solution for performing in-situ volcanic gas measurements. The Deep Carbon Observatory expedition to volcanoes in Papua New Guinea tested several remotely piloted, single-drone approaches to measuring gas plumes [96]. Several remote-piloted platforms were also tested at Masaya Volcano, Nicaragua [134]. [53] used UAVs equipped with miniature mass spectrometers to perform in-situ gas measurements the Turrialba Volcano in Costa Rica. The 2018 eruption at Kīlauea [105] and subsequent caldera collapse was extensively monitored with UAV-based imagery. These examples highlight hand-piloted UAVs and further underscore the need for automation in this space.

[142], developed and demonstrated a decentralized autonomous multi-drone flocking algorithm that avoids collisions between drones while maintaining a cohesive flock, a specialization of the canonical Boids flocking simulation in [118]. We apply the collision-avoidance and velocity-matching techniques described there and add our formation strategy.

The contributions of this work are as follows:

1. We develop the Dragonfly UAV platform as a versatile autonomous volcano survey tool.

2. We implement rasterization survey algorithms and extend our LoCUS algorithm to use a flocking strategy to follow gas gradients to their source.

3. We demonstrate that the hardware platform and algorithms successfully measure known simulated gas concentrations and sources in the

field using hybrid simulation/hardware experiments.

4. We validate that the VolCAN swarm can detect gas emissions and locate known gas sources in the challenging field conditions at the Valles Caldera active volcano. Our field tests demonstrate the utility of distributed sensing and communication among coordinated UAVs for surveys in challenging environments.

3.4 Methods

This section describes the development of the VolCAN swarm hardware and software as well as testing in simulation, hybrid simulation/field experiments, and natural field conditions. Simulation experiments were performed using the Gazebo real-physics simulator [92]. Field experiments were conducted at two New Mexico, USA sites: Balloon Fiesta Park in Albuquerque and near the Valles Caldera supervolcano.

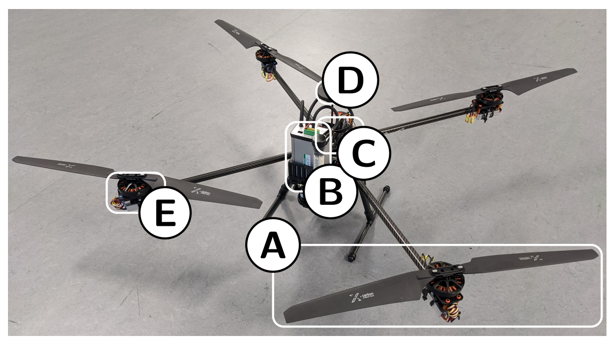


Figure 3.2: **Dragonfly UAV** designed for volcano monitoring with arms and landing gear unfolded in flight-ready configuration. (A) 56.80 cm diameter propellers (highlighted for scale reference). (B) PP-Systems SBA-5 CO_2 sensor with absorber column weighing 0.5 kg. (C) Onboard flight computers and electronics, including a Raspberry Pi. (D) Hex Here2 GPS. (E) T-Motor MN6007 320.00 kV motors and Flame 60.00 A ESCs.

3.4.1 Dragonfly Design and Mission Parameters

We designed and built the Dragonfly UAV (Figure 3.2) to fly with a 2.00 kg payload, including a CO_2 sensor, for 1.00 h duration. We chose larger motors and propellers to provide enough thrust to be able to fly in high winds. The highest wind speed under which we successfully tested the platform was $16.00\,\mathrm{m\,s^{-1}}$. During hover and under normal flight dynamics, the system uses between 15.00 A and 20.00 A. The Dragonfly folds to fit in a backpack case for transportation on foot to volcanic field sites. The foldable Tarot 650 frame allows for a variety of sensor configurations and payloads mounted to the payload rails. The UAVs communicate with each other and a base station through an ad-hoc wifi network. They can operate autonomously and under the guidance of a *scientist-in-the-loop* [19]. The Dragonfly design is centered around the MavROS programming interface to implement autonomous control. The Dragonflies were built using commodity hardware and 3D-printed parts to reduce costs and make it possible to do common repairs on-site.

The laboratory-grade PP-Systems SBA-5 CO_2 detection sensor was chosen to fill the requirement of CO_2 gas concentration sensing because its durability, accuracy of 1.00 ppm to 2.00 ppm, wide detection range, and mass of only 200.00 g [134]. The sensor is also capable of operating at high altitudes and in a wide range of operating temperatures [83, 93]. In our initial design experiments rotor wash was a confounding factor in

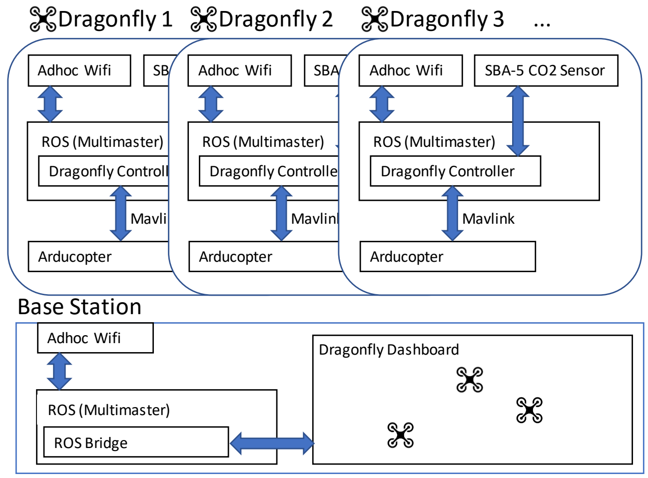


Figure 3.3: **Block level diagram of the VolCAN swarm.** Each Dragonfly in the swarm maintains flight using the on board Arducopter flight computer which is directed by the companion Raspberry Pi computer running the Dragonfly Controller. The Controller executes mission commands autonomously flying the Dragonfly as a virtual pilot while also communicating to other Dragonflies and the Base Station via an ad-hoc wifi network.

CO_2 measurement, for scales less than 3.00 m; however, 3.00 m is below the relevant resolution for volcanic plumes.

3.4.2 The Dragonfly Software Platform

Dragonfly software is comprised of two main components: the Dragonfly dashboard and the onboard controller. These components integrate the ecosystem of modules to control the VolCAN swarm as a whole as depicted in Figure 3.3.

The Dragonfly dashboard[2] is a human-friendly interface for planning missions. The dashboard provides a convenient Graphical User Interface (GUI) ground station for visualizing and managing the swarm of networked Dragonflies on a 3D map. The dashboard gives the user the ability

[2]Dragonfly Dashboard source code: https://github.com/BCLab-UNM/dragonfly-dashboard/tree/FRONTIERS2021

to control and provide expert feedback to the entire swarm, a foundation of the scientist-in-the-loop goal.

The Dragonfly Controller[3] acts as a virtual pilot. It is a collection of Robot Operating System (ROS) [115] melodic nodes running on the onboard companion computer. These nodes were run in a multimaster ROS environment, allowing ROS to broker communication between Dragonflies and the ground station.

The Dragonfly Controller contains the following ROS nodes.

1. **$\mathbf{CO_2}$ ROS Sensor Node** publishes data from the connected SBA-5 CO_2 sensor at 10.00 Hz. This allows any Dragonfly to stream any other Dragonfly's CO_2 readings.

2. **Data Logger** records CO_2 measurements and their global positioning system (GPS) coordinates.

3. **Command Service** provides high-level flight commands to operate the Dragonflies, including common actions like *takeoff*, *land*, *return to launch (RTL)*, *goto waypoint* along with the following actions:

   (a) Execute *DDSA* or *Lawnmower* (Section 3.4.3).

   (b) *Flock* and *Coordinated Gradient Descent*. These two commands direct the Dragonflies to organize into a flock and follow CO_2 gradients (Section 3.4.4).

[3] Dragonfly Controller source code: https://github.com/BCLab-UNM/dragonfly-controller/tree/FRONTIERS2021

(c) *Mission* executes a list of actions in series on the Dragonflies but in parallel across the swarm. Mission actions included all of the **command service** actions along with a *semaphore* that acts as an execution barrier to synchronize survey algorithm execution across multiple Dragonflies.

3.4.3 Preplanned Survey Algorithms

To map the CO_2 of a region of interest, we implement two rasterization survey algorithms: the lawnmower survey algorithm and the distributed deterministic spiral search algorithm (DDSA) [71] [71, 6] survey algorithm. Both of these algorithms create an exhaustive 2D rasterization map by visiting each area within a given radius. The lawnmower algorithm scans a polygon region by incrementally following longitudinal passes across a predefined region. We implement the lawnmower algorithm using a linear programming framework to perform boundary calculations against a user-defined polygon. This technique allows the mapping of irregularly shaped regions and avoids hazards such as trees, power lines, and sudden elevation changes which are commonplace in the target environments. The DDSA algorithm is a multi-agent spiral search algorithm that navigates multiple drones in interleaved square spiral paths. Unlike the lawnmower algorithm, the DDSA algorithm guarantees collision avoidance because the interleaved paths never cross. The output of these algorithms are

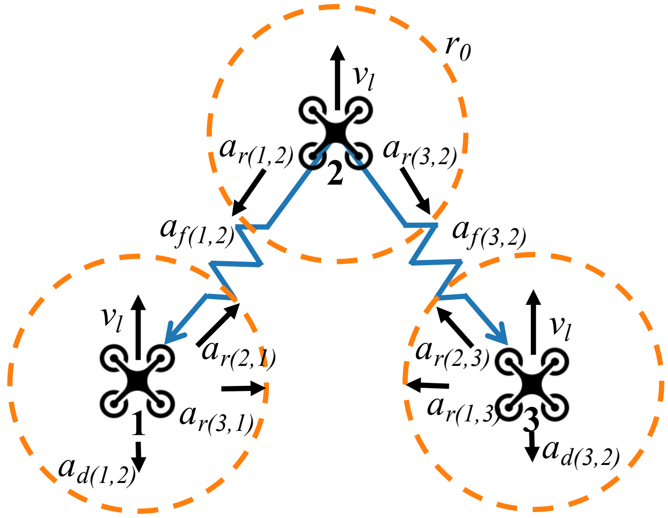


Figure 3.4: **Flocking Algorithm.** This diagram highlights the constituent forces acting on each Dragonfly, which are summed to produce the v_i control velocity on each of the three drones flocking formation. The dotted circles represent the minimum repulsion radius r_0 where $a_{r(i,j)}$ pushes two drones apart (see Equation (3.1)). The springs labeled with $a_{f(i,\ell)}$ are the flocking forces maintaing the drone formation in relation to Dragonfly 2 (see Equation (3.2)). $a_{d(i,\ell)}$ is the velocity dampening force (see Equation (3.3)), and v_ℓ is the leader velocity applied to the flock (see Equation (3.4)).

waypoints which the Dragonfly autonomously navigates during a mission. GPS stamped CO_2 data sets logged from these flights are ideal to create Kriging CO_2 concentration maps due to their uniform region coverage.

3.4.4 Flocking Algorithm for Gradient Descent

While the lawnmower survey algorithm is an autonomous pre-planned algorithm, the gradient descent flocking algorithm adapts the paths of the UAVs in response to the data they sense and communicate to each other. The gradient descent algorithm goal is for the Dragonflies to navigate to an unmapped location where the gas flux is highest which would identify the location from which the gas is emitted without spending the extra time to rasterize the surrounding area. This increase in efficiency follows work by

[6]. To spatially coordinate multiple Dragonflies, we used a leader-based flocking algorithm following [142] detailed in Figure 3.4.

Flocking drones avoid collisions by using a drone i, to drone j, repulsion force a_r. This force acts like a virtual spring between drones within a radius r_o of each other. r_1 acts as a maximum repulsion force:

$$a_r = -\sum_{\forall i,j:\, i \neq j, |x_{i,j}| \leq r_o} \min(r_1, r_0 - |x_{i,j}|) \frac{x_{i,j}}{|x_{i,j}|}. \tag{3.1}$$

To maintain the formation, a potential well applies force $a_{f(i,\ell)}$, aligns each dragonfly at their respective positions $x_{f(i)}$. $x_{f(i)}$ is a specified offset from the leader ℓ of $x_{\ell,i}$ given by,

$$a_{f(i,\ell)} = \frac{x_{\ell,i} - x_{f(i)}}{\left|x_{\ell,i} - x_{f(i)}\right|}. \tag{3.2}$$

A dampening term is used to prevent overshooting the leader ℓ when matching velocities:

$$a_{d(i,\ell)} = v_\ell - v_i. \tag{3.3}$$

To achieve formation flocking each force vector is scaled by a corresponding gain term c, c_f, and c_d and update time Δt to give a velocity vector v_i $\mathrm{m\,s^{-1}}$:

$$v_i = v_\ell + \Delta t(c_r a_r + c_f a_{f(i,\ell)} + c_d a_{d(i,\ell)}). \tag{3.4}$$

Formation flocking is implemented in Algorithm 1. FLOCK is called on an interval, once per Δt time-step which updates the velocity of the given drone in the flock by calling the SETVELOCITY function.

In previous work, we developed the LoCUS algorithm for formation flying which used a lock-step technique to move the formation in space [55]. We replaced this technique in LoCUS with the above flocking algorithm. This enabled the inherent dynamism of flocks to make collision avoidance more natural and, in general, made the group of UAVs more responsive to changing inputs.

Algorithm 1 Flocking Velocity Update Algorithm

function FLOCK($leaderIndex$, $selfIndex$, $positions$, $velocities$)
  $a_r \leftarrow \text{REPULSION}(selfIndex, positions)$
  $a_f \leftarrow \text{FORMATION}(leaderIndex, selfIndex, positions)$
  $a_d \leftarrow velocities[leaderIndex] - velocities[selfIndex]$
  $v \leftarrow velocities[leaderIndex] + \Delta t(c_r a_r + c_f a_f + c_d a_d)$
  SETVELOCITY(v)
end function

The dragonfly flock performs gradient descent by following the plume's atmospheric dispersion gradient towards the concentration maximum at the source of the plume. The gradient is calculated by the lead drone by aggregating CO_2 measurement point-sources φ and corresponding 2D spatial positions (x, y) of each agent α in the swarm at each time step. This data is fit with a linear slope b relating point-source concentration to position in the form $Ab + \epsilon = s$, and the slope parameter b is calculated by minimizing the magnitude of the error term ϵ through least-squares approximation:

$$\underbrace{\begin{bmatrix} 1 & \alpha[1].x & \alpha[1].y \\ 1 & \alpha[2].x & \alpha[2].y \\ \vdots & \vdots & \vdots \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} b[0] \\ b[1] \\ b[2] \end{bmatrix}}_{b} + \underbrace{\begin{bmatrix} \epsilon[0] \\ \epsilon[1] \\ \vdots \end{bmatrix}}_{\epsilon} = \underbrace{\begin{bmatrix} \alpha[1].\varphi \\ \alpha[2].\varphi \\ \vdots \end{bmatrix}}_{s} \tag{3.5}$$

The linear fit slope b components include the vector $(b[1], b[2])$ which points in the direction of maximum CO_2. To gradient descend in two dimensions, at least three drones are required for the linear fit. Additional drones above this minimum provide redundancy and an averaging effect across the swarm's CO_2 readings.

3.4.5 Simulation

To facilitate repeatable experiments under controlled conditions and to accelerate development, the Dragonfly hardware, Ardupilot flight control software, and the onboard Dragonfly Controller were implemented in the Gazebo real-physics simulation as shown in Figure 3.5.[4] We ran the same software in simulation (also known as Simulation-in-the-Loop) as in the physical hardware which expedited algorithm prototyping and debugging before flying in a physical UAVs.

We define a Virtual Plume for use in Gazebo simulations and hardware/simulation hybrid experiments (described in the next section). The

[4]Simulation source code: https://github.com/BCLab-UNM/dragonfly-sim/tree/FRONTIERS2021

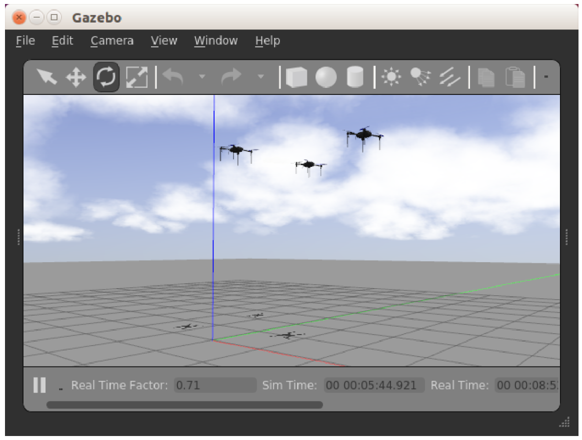


Figure 3.5: **Algorithm Testing in Simulation** with three simulated Dragonflies, each running their own instance of Arducopter and the Dragonfly Controller.

plume model is a horizontal 2D slice through a 3D Gaussian plume. The concentration is calculated from a x, y GPS coordinate offset with constants stack height H, wind speed u, emission rate Q, and diffusion rate K [136]:

$$\text{PLUME}(x, y) = \frac{Q}{2\pi K x} \exp\left(-\frac{u(y^2 + H^2)}{4Kx}\right). \tag{3.6}$$

3.5 Results

Experiments were conducted at two field sites. First, hardware Dragonflies were flown in an open field and tasked with the mission of mapping and finding the source of a virtual plume. This allowed us to evaluate how the algorithms would behave in real hardware in an outdoor environment. The virtual plume allowed us to evaluate how effectively the UAVs would map

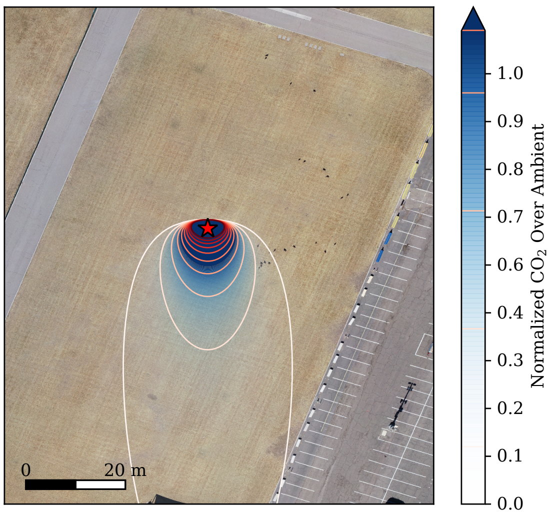


Figure 3.6: **Virtual Plume Plotted on Balloon Fiesta Park.** The Virtual Plume, with the source specified to be the middle of the field, is configured with a northerly wind producing the long tail. The isoconcentration lines are added to accentuate the plume's shape at lower concentration levels.

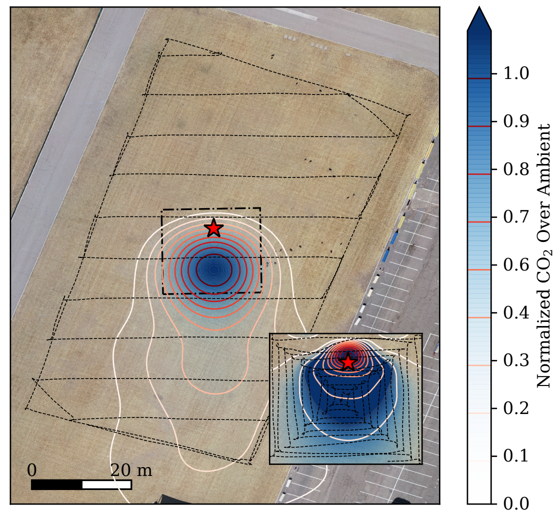


Figure 3.7: **Coarse-grained lawnmower flight and more thorough survey using the DDSA (inset) flight at Balloon Fiesta Park**. Each lawnmower pass across the field is separated by 10.00 m whereas each arm of the DDSA is separated by 1.00 m. Virtual plume data collected on the field is represented in the Kriging map. This map is compared with the Virtual Plume plot to see how the Kriging map represents the plume with limited information.

known gas concentrations and the known source location of the simulated CO_2 plume.

Second, experiments were performed at a natural volcanic site to test the plume-sensing capabilities of the Dragonfly platform under real-world conditions. Previous field studies were conducted by geoscientists at the site and sites of elevated CO_2 were identified in [80]. This provided us with a likely location for CO_2 emissions; however, CO_2 emissions change frequently, and measurements are affected by wind and temperature. Thus, it is difficult to acquire accurate ground truth. Therefore, we used the simulated plume at the first field site to show the VolCAN swarm could accurately map CO_2 concentrations and source location. The field study at Valles Caldera volcano demonstrated that the swarm could produce rasterized surveys and flock to a suspected source of CO_2 under difficult field conditions but where there is uncertainty in true sources and concentrations of CO_2. All field experiments were conducted according to U.S. Federal Aviation Administration (FAA) UAS regulations and with permission in relevant airspaces.

3.5.1 Open Field Experiments

Our open field site was the large flying field of Balloon Fiesta Park in Albuquerque, New Mexico. We used the Virtual Plume during field experiments to add a plume for the algorithms to map and follow. We centered the Virtual Plume at latitude 35.19465, longitude -106.59625 with parameters $H = 2.00\,\text{m}$, $u = 1.00\,\text{m/s}$, $Q = 5.00\,\text{kg/s}$, $K = 2.00\,\text{kg/s}$.

Rasterization Survey

To map the virtual plume a large-scale 10.00 m resolution rasterization survey of the field was performed using the lawnmower algorithm within a polygon outlining the designated field. The Dragonfly flew at 10.00 m altitude, with each longitudinal pass at 10.00 m spacing. We used the lawnmower survey to produce a Kriging heat map of the virtual plume seen in 3.6.

For a fine-grained 1.00 m resolution survey of the virtual plume, we executed a DDSA slightly to the north of the detected plume. We used a single Dragonfly in the DDSA with spiral arms separated by 1.00 m spacing, and we performed 10 loops at a single altitude of 10.00 m. The goal was to produce a detailed Kriging heat map of the Virtual Plume.

We generated two maps to test the ability of the two sample rasterization methods to recreate the simulated plume over the open field. The ground-truth virtual plume is depicted in Figure 3.6. The coarse-grained map of CO_2 generated from data using the lawnmower algorithm and the fine-grained map generated using the DDSA are displayed in Figure 3.7. The CO_2 plume is visible in the center of the field, even with the limited data available from the 10.00 m separated longitudinal passes of the lawnmower algorithm. The DDSA map, displayed in the lower right detail, has more structure due to the finer-grained arm separation of 1.00 m that closely resembles the simulated plume. The fine-grained DDSA algorithm estimated the highest CO_2 concentration very close to the location of the plume source (indicated by the red star). To compare the sampled maps S against the ground-truth maps G of shared dimensions (m, n), we find the mean absolute difference MD over the 2D space of these normalized data sets:

$$\mathrm{MD}(S, G) = \frac{\sum_{i,j} |S_{i,j} - G_{i,j}|}{m \times n}. \tag{3.7}$$

The coarse-grained lawnmower mean absolute difference is 0.1306 and the fine-grained DDSA mean absolute difference is 0.0380 indicating the higher resolution of the fine-grained DDSA was able to reproduce a more accurate representation of the ground-truth data. In addition, the course-grained lawnmower estimated the source at approximately 10.00 m away from its actual location whereas the fine-grained DDSA estimated the source at approximately 1.00 m from the ground-truth source, again representing that a finer-grained map produces a more accurate representation of the ground truth.

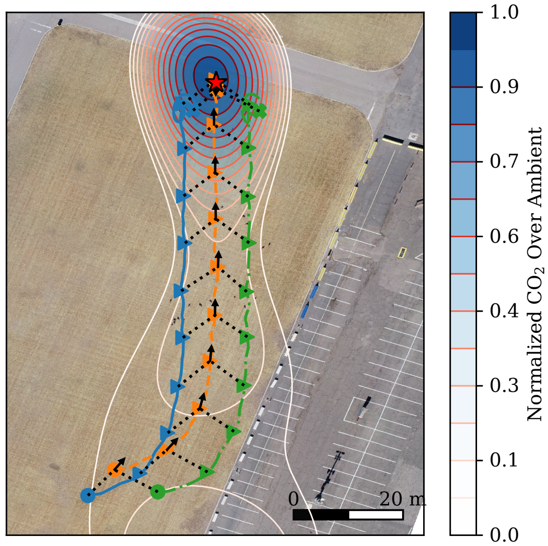


Figure 3.8: **Flocking Gradient Descent in the Field** mapping a virtual plume. Colored lines represent the flight paths of the Dragonfly flock in V-formation following the gradient. Dotted lines between key points in time, marked as directional triangles, indicate the network connections during flight. Arrows indicate the normalized gradient of the virtual plume, which is represented as a Kriging heat map. From the starting point, the flight path length was about 100.00 m and took the flock about 2.00 min to reach and identify the plume's maximum concentration, which is within 0.30 m of the source. The Virtual Plume was moved up field from the previous experiments to allow for more travel distance of the flock.

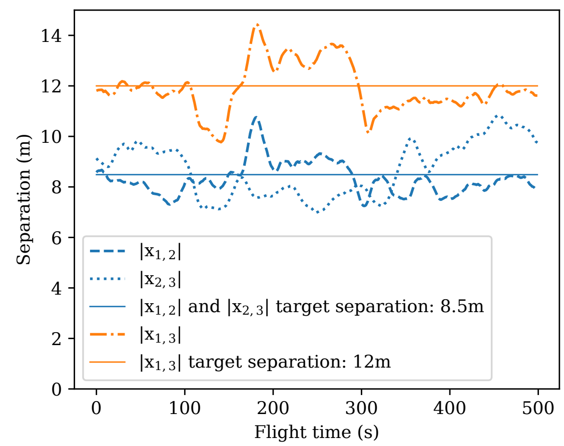


Figure 3.9: **Flocking Algorithm Achieves Target Separation**. The distance magnitude between Dragonfly a and b is signified $|x_{a,b}|$. All three Dragonflies stay well away from each other, only reaching a minimum distance of 6.98 m. Likewise, Dragonflies 1 and 2 stay within a maximum of 10.82 m of Dragonfly 2, keeping the flock in formation.

Flocking and Gradient Descent

We implemented LoCUS on 3 Dragonflies with Dragonfly 2 as the lead UAV and Dragonfly 1 and 3 oriented at (x, y) offsets ($-6.00\,\mathrm{m}$,$-6.00\,\mathrm{m}$) and ($6.00\,\mathrm{m}$,$-6.00\,\mathrm{m}$) respectively. This produced a V-formation orienting Dragonfly 1 and 3 orthogonally via Dragonfly 2, which is ideal for detecting a 2D gradient and separating the drones to avoid collisions due to GPS accuracy. For these experiments, we flew each Dragonfly at different altitudes, separated by 1.00 m, as an additional safety measure to avoid mid-air collisions. To test gradient descent, Dragonfly 2 was commanded to perform gradient descent autonomous flight using CO_2 readings correlated with location information from Dragonfly 1, 2, and 3. The Dragonfly flock was positioned in the plume's tail to start with an initial signal that was used to follow the virtual CO_2 plume towards the source. The goal of this experiment was for the flock to identify the source of the plume represented by the location with maximum CO_2 concentration.

The purpose of flocking is to identify the location with the highest CO_2 concentration using multiple UAVs close while preventing collisions. Figure 3.9 shows the Euclidean distances between drones during a manually piloted test flight. Dragonfly 1 and 3 reach a minimum and maximum distance between themselves and the lead Dragonfly 2 of 6.98 m and 10.82 m, respectively. These distances lie within 2.30 m of the ideal configured distance of 8.49 m. Similarly, Dragonfly 1 and 3 stay within 2.43 m of the

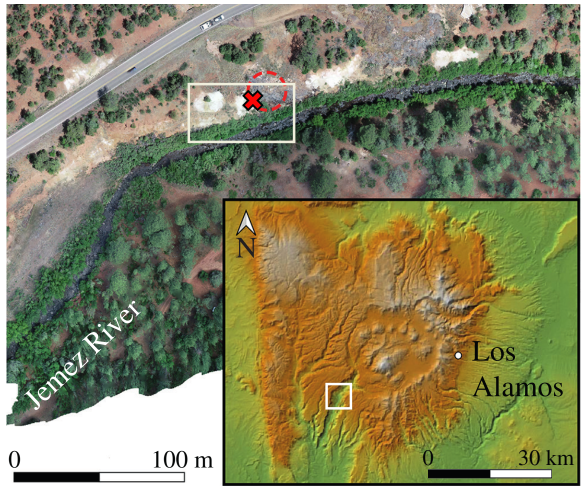


Figure 3.10: **Valles Caldera Supervolcano: Overview of Field Site.** A visible imagery mosaic of the field site, with an inset displaying the terrain of the the Valles Caldera volcano. In previous surveys, CO_2 emissions were associated with the white calcite surface deposits [117]. The calcite associate with the primary CO_2 source, circled in red, has been obscured by brush. The white rectangles indicates the survey site. The red $\times$ indicates the position of a fixed ground sensor.

ideal configured distance of 12.00 m. All three Dragonflies fly more than 6.00 m away from each other, greater than the acceptable GPS error radius of 6.00 m. This effectively kept each drone in an orthogonal orientation which is essential for collecting a 2D sample gradient vector while avoiding collisions between members of the flock.

Figure 3.8 shows the path of the three flocking Dragonflies as they follow the gradient vector indicated by the black arrow. The arrow is a normalized representation of the $(b[1], b[2])$ vector described in Section 3.4.4. Dragonfly 2 successfully navigates into the virtual CO_2 plume by following the b vector, finding the maximum CO_2 value of 708.52 ppm. This result matches our results from [55], where the flock can quickly and directly

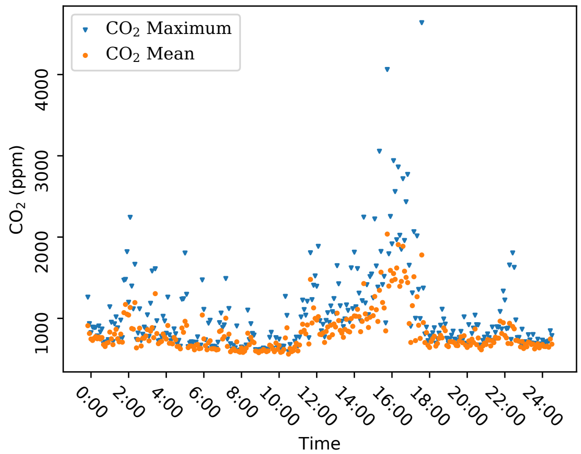


Figure 3.11: **CO_2 Concentrations Measured by Ground Sensor** from a permanently installed multiGAS system developed at the University of New Mexico [9].

follow the plume's gradient back to the source. Just as with the flocking result, we produce an accurate Kriging map from the data collected across all three drones, which produces a map similar to the one produced in Figure 3.7 and Figure 3.8.

3.5.2 Volcano Field Tests

Valles Caldera is a supervolcano in northern New Mexico (Figure 3.10). The caldera is more than 22.00 km in diameter with CO_2 emissions at several sites [79]. The field site chosen for VolCAN swarm surveys and flocking is a small canyon formed by the Jemez River. CO_2 degases diffusely out of the ground on the northern side of the canyon [80]. The location offers a challenging flight environment because of the forest and steep canyon hills on either side. This bracketed the available flight space and required extremely accurate mission planning and flight control. Our challenge was to balance flying low enough to detect the degassing CO_2

emitted from the ground, high enough to clear the tallest trees, and within a boundary to not collide with the canyon hills. There was quite a bit of effort put into mission plans including sighting the treetops, creating and analyzing a topology map, reviewing CO_2 data collected during the missions, and manually adjusting mission boundaries. Additionally, the design and tuning of the Dragonflies played a large component in flying accurately even in windy conditions. Despite our efforts, We experienced occasional catastrophic collisions with trees while tuning flight parameters. Unfortunately, damage to the platform significantly affects the flight characteristics of the UAVs which underscores the importance of planning flights clear of obstacles. While Valles Caldera is safe for researchers because there is no current danger of eruption, it is an ideal real-world test site for the VolCAN swarm due to its active release of gasses and its topology typical of challenges in volcanic regions.

The site was previously surveyed in [117] for CO_2 emissions. The evidence for historical emissions is the white areas of hydrothermal alteration. A permanent ground sensor is located at the location of the highest CO_2 emissions indicated by the red star in Figure 3.10. An example of the CO_2 emissions detected from the ground sensor is shown in Figure 3.11 and highlights the variability of CO_2 over time.

Rasterization Survey

We executed a large-scale rasterization of the area at 50.00 m above the canyon floor takeoff location. Flying at this height cleared the tallest trees in the area but still was able to detect a difference in CO_2. The large-scale rasterization lawnmower was executed with longitudinal passes at 5.00 m spacing. Additionally, we flew the lawnmower with three dragonflies in flocking v-formation with Dragonfly 1 and 3 oriented at (x, y) offsets (-6.00 m, -6.00 m) and (6.00 m, -6.00 m) respectively. Flocking allowed us to gather redundant data during the rasterization survey. By simultaneously collecting CO_2 data from multiple nearby UAVs, we increased the volume of data collected during the mission to create more detailed maps. Additionally, this approach allowed us to compare CO_2 concentrations from three different sensors to understand variation in both the environment and the sensors over the same period. CO_2 concentration readings from this survey were gathered and used to generate a Kriging map of the concentration gradient against aerial photography of the region.[5]

Figure 3.12 shows the Kriging map generated from CO_2 data combined from all dragonflies in the flock. The Kriging map shows a complex distribution of CO_2 with the highest elevated CO_2 emanating from a previously unknown source to the east, and an elevated plume above and slightly to

[5] Video of the survey: https://youtu.be/VVz68ZqhD8k

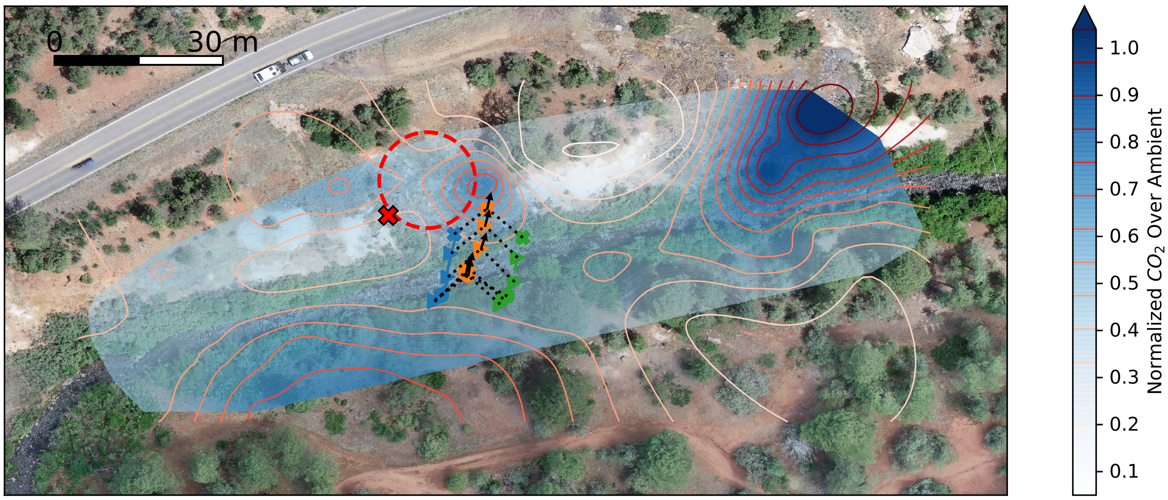


Figure 3.12: **Kriging isoconcentration map produced from CO_2 data collected during the flight.** This map is the result of data collected using 3 Dragonfly drones flying the pattern in formation. Regions of note are the elevated concentration above and slightly to the east of the known source outlined in the dashed-red circle and the region in the east of the plot indicating another CO_2 source plume. The path of the flock of Dragonflies is drawn as they follow the detected gradient which corresponds to the known source and previously mapped gradient.

the east of the known source indicated by the dashed red circle. Detected CO_2 across the swarm and overtime is normalized to produce the percentage difference in CO_2 shown in the figure. Normalization is performed by linearly scaling the data to the minimum and maximum CO_2 reading over the data set. Detected CO_2 readings were between 410.00 ppm and 434.00 ppm across Dragonfly 1, 2, and 3. The total flight time to rasterize the area was 13.00 min 42.00 s.

Flocking and Gradient Descent

We followed a CO_2 gradient to its source by performing gradient descent with the following mission. First, we flocked Dragonflies 1, 2, and 3 in

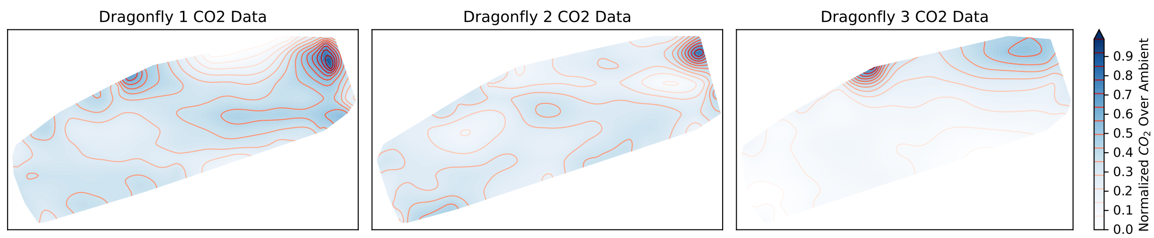


Figure 3.13: **Kriging maps of the individual component Dragonfly CO_2 readings combined in Figure 3.12.** Of note are the common elevated regions in the upper right and mid-left of each survey. These common readings confirm that there is elevated CO_2 in that region of the sky.

V-formation with Dragonfly 2 as the leader. We then navigated the flock to a waypoint position south of the known source along the Jemez River. Finally, we triggered Dragonfly 2 to follow the gradient calculated from data collected from the swarm. The goal of this experiment is to validate that gradient descent can navigate the swarm of drones using a previously identified natural source of CO_2.

In addition to the Kriging map survey, Figure 3.12 displays the path of the Dragonfly flock following the CO_2 concentration gradient using gradient descent in real-time. The path starts to the south of the known source and proceeds to move north until a maximum CO_2 of 428.00 ppm was reached due east of the known source indicated by the dashed red circle. This path follows the gradient previously detected by the rasterized survey and shows the flock effectively using the input CO_2 data to fly towards the known source plume's highest concentration. The total flight time for the gradient descent portion of the mission was 25.00 s.

3.6 Discussion

Our results show that the Dragonfly UAV platform effectively maps aerial CO_2 emissions in the challenging, real-world conditions of volcanic environments. We developed the Dragonfly UAVs platform with the navigational capabilities, flight duration, and payload capacity foundations to be able to accurately collect CO_2 gasses, analyze them in-situ, and respond to the detected concentrations in real-time across multiple UAVs. This culminated in the rasterization survey of a known hot-spot in the Valles Caldera where we autonomously flew the VolCAN swarm in formation and mapped the CO_2 at 50.00 m. This highlighted the elevated CO_2 around the known source but also indicated an additional previously unknown source to the northeast. Additionally, we executed an autonomous gradient descent that successfully navigated into the elevated CO_2 near the known source in the area. These field tests demonstrate the utility of the VolCAN swarm in mapping and navigating CO_2 gradients in real-world extreme environments.

The development of the Dragonfly UAV platform took considerable efforst to meet the mission requirements. The biggest driving requirement was the flight time which dictated the overall weight of the aircraft, specifically the large battery requirements. This drove motor and propeller sizes to be able to produce enough lift with agility. To control the

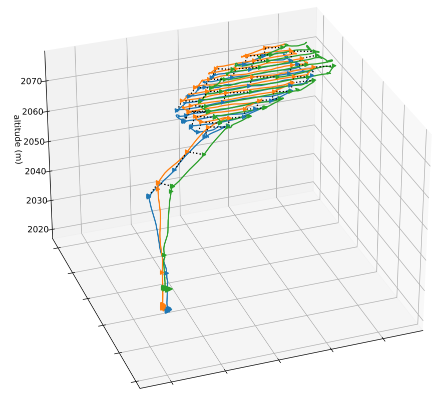


Figure 3.14: **3D flight map of the Dragonflies executing the lawnmower mission to produce the Kriging map displayed in Figure 3.12.** The VolCAN swarm takes off and assembles in V-formation then proceeds to execute the lawnmower pattern in the predefined polygon region. The flocking flight paths overlap over time producing redundant confirmation data of the CO_2 concentrations.

aircraft, we found that the flight controller PID tune was critical and only found success after hand-tuning these parameters. Tuning the PID loop to be somewhat aggressive resulted in an aircraft that would behave well in gusty high wind conditions, perform as expected around other aircraft, and mirror the aircraft's simulation behavior.

Trials of the Dragonfly UAV in the open field helped solidify the algorithms developed in simulation in a real-world environment. Flying in the semi-controlled environment with a virtual plume ensured that we could map a CO_2 plume and represent it with a level of confidence. Our results show that the coarse-grained rasterization produced a map with a mean absolute difference of 13% to ground truth, with a smaller mean absolute difference of 3% for the fine-grained rasterization. Flocking the Dragonflies and following the virtual plume gradient demonstrated that we could identify the source of the plume, represented by the highest CO_2 concentration in the gradient.

Mapping the known sources around the Jemez river in the Valles Caldera offered a chance to test the VolCAN swarm against real CO_2 sources. To map the overall region we had to fly at a high elevation above ground level to avoid the tree canopy obstructions. This resulted in a relatively low CO_2 signal compared with ground measurements. However, due to the accuracy of the SBA-5, we were still able to detect differences on the order of 1.00 ppm making the collection missions still effective. This further highlights the efficacy of our technique in mapping and responding to the aerial CO_2 signal.

The missions at the Valles Caldera highlighted the survey speeds. During the same day as the aerial surveys, a ground survey team was gathering CO_2 groud flux and concentration readings using portable CO_2 fluxmeters. Their survey of a region similar to the lawnmower survey took 8.00 h to complete. In comparison, the lawnmower survey took less than 15.00 min and the gradient descent took less than 30.00 s, each an order of magnitude faster than the previous.

Our survey methods are also applicable to a variety of other environmental monitoring tasks that require in-situ measurements and efficient localization, such as detecting gas line leaks and environmental monitoring.

3.6.1 Caveats and Future Work

With any data collection task, especially one of such a dynamic process, the challenge is to tie the data back to known ground truth. In our case, the data and Kriging maps demonstrate a high correlation to the virtual plume in the open field environment. This ensures that data collection missions in the field represent a facsimile of what is truly occurring. That being said, the data represented by flying high above the ground and known sources in Jemez Springs should be interpreted with caution. The low CO_2 readings did correlate to the known source on the ground, and indicate another source to the northeast, but a slight breeze can easily shift the location of these gasses. We propose that mapping larger regions, on the order of 100.00 m will help in localizing gas sources or taking into account wind direction and speed to produce an offset.

Upon inspection of the data used for gradient descent, it is feasible that the low CO_2 concentrations and proximity of each drone to each other that differences in sensor temperatures and calibration played a factor in the detected gradient. The bottom line is that the drones did respond to the gradient and future work will be dedicated to determining the spacing of the flock to best gather and follow gradients.

With the development of the VolCAN swarm including the Dragonfly platform and the algorithms to map and follow gradients, the next step is to further automate the swarm's behavior. This includes handing over decisions when and where to perform a survey and at what resolution, where to perform gradient descent to find the source of emissions, and utilizing battery life to maximize the investigation of a region. Also, automating the initial setup and mission parameters like the elevation to clear obstacles, and the overall flight boundaries would help speed the overall mission.

Acknowledgments

JE support provided by Department of Energy's Kansas City National Security Campus, operated by Honeywell Federal Manufacturing & Technologies, LLC under contract number DE-NA0002839. GMF, SN, TF, RF, and MM support provided by the VolCAN project under National Science Foundation grant 2024520. Support was also provided by a Google CSR award. This study received funding from Google and Honeywell Federal Manufacturing & Technologies, LLC. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication. All authors declare no other competing interests. We thank the Hummingbird Music Camp and the City of Albuquerque Parks and Recreation for their support. We thank Jarett Jones and Brian Verkaart for technical advice supporting the development of the Dragonflies.

Chapter 4

Drone CO_2 Measurements During the Tajogaite Volcanic Eruption

4.1 Publication Notes

Citation: Ericksen, John, et al. ”Drone CO 2 measurements during the Tajogaite volcanic eruption.” Atmospheric Measurement Techniques 17.15 (2024): 4725-4736.

Publication date: 15 August 2024

Journal: Atmospheric Measurement Techniques

Publisher: Copernicus Publications

Formatting: The original published text has been preserved as much as possible while still adhering to the formatting requirements of this dissertation.

Data and Software Availability: Data and plot generation code used in this paper is publicly available at https://github.com/BCLab-UNM/

lapalma-expedition.

UAS code available at:

https://github.com/BCLab-UNM/dragonfly-dashboard

https://github.com/BCLab-UNM/dragonfly-controller.

Funding: This work was supported by funding from the following: JE support provided by the Department of Energy's Kansas City National Security Campus, operated by Honeywell Federal Manufacturing & Technologies, LLC under contract number DE-NA0002839. GMF, SN, TF, RF, and MM support provided by the VolCAN project under National Science Foundation grant 2024520. Support was also provided by a Google CSR award. This study received funding from Google and Honeywell Federal Manufacturing & Technologies, LLC. VOLRISKMAC II (MAC2/3.5b/328) financed by the Program INTERREG V A Spain-Portugal MAC 2014-2020 of the European Commission. Thanks to Samantha Wolf for help calculating the flux.

4.2 Abstract

We report in-plume carbon dioxide (CO_2) concentrations and carbon isotope ratios during the 2021 eruption of Tajogaite Volcano, La Palma Island, Spain. CO_2 measurements inform our understanding of volcanic contributions to the global climate carbon cycle and the role of CO_2 in eruptions. Traditional ground-based methods of CO_2 collection are difficult and dangerous and as a result only about 5% of volcanoes have been directly surveyed. We demonstrate that UAS surveys allow for fast and relatively safe measurements. Using CO_2 concentration profiles we estimate the total flux during several measurements in November 2021 to be $1.76 \pm 0.20 \times 10^3$ to $2.23 \pm 0.26 \times 10^4$ t day^{-1}. Carbon isotope ratios of plume CO_2 indicate a deep magmatic source, consistent with the intensity of the eruption. Our work demonstrates the feasibility of UAS for CO_2 surveys during active volcanic eruptions, particularly for deriving rapid emission estimates.

4.3 Introduction

Measurements of volcanic CO_2 emissions during eruptions are critical for understanding magma and eruption dynamics. CO_2 is a significant greenhouse gas [20] and making measurement of CO_2 emissions is important for climate science. CO_2 gas is second only to water vapor in abundance in volcanic emissions [78]. Despite the significance and abundance of CO_2 in the Earth System in general and in magmatic systems in particular, measuring the emission rates of this gas from volcanic craters, diffuse sources, and low-level hydrothermal sites has remained a major challenge [63]. As a result, detailed CO_2 surveys have been conducted at just 5% of volcanoes [64].

The main contributions of this work are that, for the first time, we estimate CO_2 flux using direct in-plume CO_2 measurements rather than using in-plume CO_2 to SO_2 ratios combined with separately measured SO_2 emissions. The second major contribution is that we perform in-situ gas sample-return during a major volcanic eruption for carbon isotope measurements. We use the Dragonfly Unpiloted Aerial System (UAS) [58] to gather samples directly from the eruption plume (Figure 4.1). The UAS transects the plume and employs an onboard infrared (IR) sensor to continuously obtain concentration readings. These readings are then used to estimate a 2D isotropic Gaussian concentration model. In-plume wind velocity measurements in combination with the plume model allow us to estimate CO_2 flux. While our technique has similarities to the 'ladder traverse' technique utilizing large in-situ sensing equipment mounted on a piloted fixed-wing aircraft [147], it has the obvious advantages of being much less costly, logistically less challenging, and less hazardous. Since our approach extrapolates the shape of the plume it requires far fewer plume transects. Crucially, the Dragonfly UAS does not use a combustion engine, which previous work has shown to contaminate CO_2 measurements and samples with jet-fuel derived organic carbon [66]. The resulting plume CO_2 concentration profile is used to guide the UAS to a productive sample return location of maximum concentration. Carbon isotope analyses of the samples reveal information, such as CO_2 source, which is relevant to

Figure 4.1: **A Dragonfly UAS returning from a CO_2 sample mission during the November 2021 eruption of Tajogaite volcano.** The large volcanic ash plume is visible in the background and contains an invisible CO_2 plume, which was the mapping target of this drone.

predicting the course of the eruption. We tested this technique during the 2021 Tajogaite volcanic eruption on La Palma Island, Spain, and compared the resulting flux estimates to the traditional ground-based CO_2 to SO_2 ratio method. As we demonstrate, UASs provide a method for obtaining in-plume gas samples, concentrations, and wind velocity measurements. Together these data allow isotope ratios to be determined and estimation of CO_2 flux, furthering our understanding of volcano dynamics during an eruption and allowing predictions of eruption intensity and duration. Our technique can be widely used at passively degassing and erupting volcanoes to obtain near-real-time CO_2 flux measurements to better constrain the global volcanic CO_2 budget, and assess volcanic activity.

4.3.1 Related Work

While global initiatives to directly determine CO_2 flux from biogenic sources, i.e. FLUXNET [108] have advanced our understanding of the surface carbon cycle, estimates of volcanic flux are to a large extent obtained by combining SO_2 flux measurements with observed CO_2 to SO_2 ratios [63]. This approach relies on two separate sets of measurements utilizing a ground-based or space-based remote sensing technique to determine the SO_2 concentration of the volcanic plume and a direct sampling or sensing technique to determine the CO_2 to SO_2 ratio. In almost all cases, these two separate sets of measurements are not made simultaneously and result in intrinsic uncertainties in CO_2 flux estimates [33]. CO_2 surveys have been performed using satellite-based approaches, for example, [89] performed CO_2 flux estimates of the 2018 Kilauea Volcano. Their work utilized the Orbiting Carbon Observatory -2 (OCO-2) to measure the CO_2 emissions from the 2018 Klīlauea eruption. A measurement of 77.1 ± 41.6 kt/day was obtained during the one day of observations where conditions enabled the collection of consistent high-quality data. Cloud coverage and aerosol are the major inhibitors for obtaining consistent CO_2 data using OCO-2. In addition, the wind direction must be near perpendicular to the satellite's orbit path and the measurements must be made down-wind from the plume. The OCO-2 16-day repeat cycle currently makes this method impractical for frequent, high-rate CO_2 flux measurements from erupting volcanoes and the only other successful volcanic CO_2 emission study was by [130] of Yasur in Vanuatu. Therefore, space-based CO_2 instruments require favorable atmospheric conditions and satellite positioning and are not yet feasible for volcano monitoring [130].

The value of UAS surveys of volcanic emissions was recognized by [149] who surveyed passive degassing SO_2 at Turrialba volcano, Costa Rica and estimated SO_2 flux. Other investigators have used UAS to measure plume SO_2 and collect plume trace gases [123] or use miniDOAS systems mounted on UAV to obtain SO_2 fluxes [134]. Recently UAS have been used to collect gas samples and measure gas compositions volcanic plumes from passively degassing volcanoes in remote regions [96, 73] and during the 2023 eruption of Litli Hrútur, Iceland to obtain information on CO_2 degassing and related carbon-isotope fractionation [69]

[75] and [147] estimate plume CO_2 flux using the parsimonious assumption that plumes are uniform. They use the mean value to estimate the flux whereas we use our observations in the field that support the hypothesis that plumes can be well modeled by Gaussian distributions. Our work relies on the assumption that a Gaussian model of the plume cross-section results in more accurate estimates of total flux.

[34] surveyed emissions of the Tajogaite eruption in early October 2021. Their survey included SO_2 measurements by UAV that were used to infer CO_2 concentrations. Our work in late November complements the Burton

et. al. survey by providing additional information on the evolution of the eruption and by using a different CO_2 flux estimation method that employs direct CO_2 measurements rather than CO_2/SO_2 ratios. Our estimates of CO_2 flux taken a month later were lower than those of Burton et. al.

4.3.2 Background

La Palma Island is in Spain's Canary archipelago [129]. The northern sector of the island hosts the oldest subaerial (on land) volcanism, characterized by repeated large lateral edifice collapses [48, 2]. Volcanism resulted in the formation of Garafía and Taburiente and then moved southward to form Cumbre Vieja volcano, at the southern part of the island. This southern system represents the last stage in the geological evolution of La Palma island, as volcanic activity has taken place exclusively on that part of the island for the last 123 ka [38]. The most recent volcanic eruption of Cumbre Vieja is Tajogaite (2021) [37, 144], preceded by that of Teneguía in 1971 [62] and San Juan in 1940 [62, 14]. At 14:10 UTC on September 19, 2021 Tajogaite volcano erupted from a vent on the western side of La Palma Island, in the vicinity of the Llano del Banco eruptive center of the San Juan eruption of 1949 [84]. The eruption was forecast using seismic, geodetic and geochemical techniques by Spanish researchers who alerted the civil protection officials several days before the start of the eruption [49]. The monitoring network of diffuse CO_2 emissions on La Palma detected magmatic CO_2 several months before the eruption [94, 119]. This monitoring activity took advantage of extensive previous work characterizing diffuse CO_2 emissions on La Palma. This work provided key insights into the dynamics of magmatic CO_2 degassing on the island [109]. The eruption itself began with an explosive phase that ejected ash to an altitude of 5 km, then transitioned to fire fountains, violent strombolian activity, and the production of highly fluid lava flows. Within 24 hours of the initial eruption a 3 km long lava flow was evident [84]. The eruption lasted for more than 85 days and built a pyroclastic cone of about 225 m in height. Over the period of the eruption, the volcano showed dynamic and changing activity with new vents frequently opening on the active cone. These vents produced explosive and effusive eruptions of varying intensity [39]. Bulk tephra, matrix glass and glass inclusions have a basanitic-tephritic composition of 43 to 46 wt%.

Since the onset of the 2021 Tajogaite eruption on September 19, frequent measurements of SO_2 emission rates using miniDOAS traverses by car, ship, and helicopter were performed. Using this data a flux of over 5×10^4 t day^{-1} of SO_2 was estimated [111]. Daily monitoring of SO_2 gas emissions occurred before and throughout the eruption using TROPOMI data from the Sentinel 5P satellite (Copernicus SO_2 satellite monitoring, Smithsonian Institution's Global Volcanism Program 2021). The range of measured emissions rates depended upon wind direction and velocity, as well as eruptive style and activity. The measured SO_2 flux ranged from 3×10^4 to 5×10^4 t day^{-1} at the beginning of the eruption and a mean of 10^4 t day^{-1} over the duration of the active eruption [15]. These SO_2 emission rates are likely different from CO_2, but provide the best available proxy for CO_2 emissions and are a useful point of comparison for our UAS-based flux estimates in addition to the measurements made by Burton et al. 2023 in October 2021 which range from 3.36×10^4 to 4.19×10^4 t day^{-1}.

Additional gas monitoring techniques deployed during the eruption included stationary Multi-GAS and FTIR-based plume gas composition measurements as well as carbon isotope analyses of plume CO_2 in collaboration with the international volcanic gas community [111].

4.4 Methods

Our aim was to measure plume CO_2 concentrations, calculate the resulting flux, and obtain isotope data from samples taken within the plume. To achieve these goals we utilized the Dragonfly UAS, with an approximate battery life of 50 min. This extended flight time enables long-distance transects to capture large plumes. CO_2 concentrations were measured by PP Systems SBA-5 IR sensor mounted on the Dragonfly with data transmitted to the pilot in real-time [58]. Wind velocity and direction were derived from the ERA5 model of the European Centre for Medium-Range Weather Forecasts 10 m height wind velocities corresponding to the time of each flight [96]. These measurements were independently validated using a hand-held anemometer and the UAS drift method [96, 73]. For the drift method, a Dragonfly was programmed to maintain its altitude but not its lateral position and allowed to drift with the plume. We used this estimate of wind velocity within the plume with the highest CO_2 concentration (Plume B) to parameterize the flux estimation (Figure 4.2).

At the location with the highest measured CO_2 concentration, a timed trigger activated a small pump, and a plume gas sample was collected into a Tedlar bag (Figures 4.2 and 4.3). We also collected gas samples of the plume from the ground when the wind direction was favorable and volcanic activity permitted. Ground-based plume samples were analyzed by Infrared Isotope Spectroscopy with a Delta Ray located at the INVOLCAN Volcano Observatory, La Palma, following the procedure described previously [66, 83]. The error bounds on the $\delta^{13}C$ measurements are less than 0.1‰ for all analyses.

We also placed a Multi-GAS instrument at an accessible and safe location about 1 km to the north of the crater. Data from this instrument recorded CO_2 and SO_2 concentrations in the gas plume. The ratios were calculated using the Ratiocalc software and we report averages for each day of the experiment.

Crosswind transects were flown downwind of the eruption to encounter the plume. CO_2 was measured at 10 hz during flights across the plume at specified altitudes relative to launch. Each measurement was correlated to the latitude, longitude, altitude, and time of the UAS during flight, giving a CO_2 concentration cross-section of the plume.

We set the ambient background CO_2 to the value observed outside the plume for each flight. The actual measurements of ambient CO_2 were made well outside of the plume (up to 400 m away from the edge of the plume) and only vary from 415 to 430 ppm.

To estimate the total flux of the plume, we perform the following procedure.

1. Convert GPS coordinates into a linear distance in meters from the launch point. Each distance is normalised to the wind direction perpendicular by multiplying it by $\cos(\text{heading}_{\text{uas}} - \text{heading}_{\text{wind}})$
2. Isolate the plume by setting an ambient CO_2 threshold and removing data points less than that threshold.
3. Fit a Gaussian curve to the data set as follows.
   (a) Calculate the mean, μ, and standard deviation, σ, of the CO_2 across the transect.
   (b) Scale the two-dimensional Gaussian curve to fit the data by choosing a constant amplitude, a, using gradient descent to minimize the squared difference between the model and plume sample data. We assume that the Gaussian shape is uniform in both x

and y dimensions.

$$\begin{aligned}
\text{GaussianModel2D}() &= a\frac{e^{-\frac{1}{2}(\frac{x-\mu_x}{\sigma_x})^2}}{\sigma_x\sqrt{2\pi}}\,\frac{e^{-\frac{1}{2}(\frac{y-\mu_y}{\sigma_y})^2}}{\sigma_y\sqrt{2\pi}} \\
&= a\frac{e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}}{\sigma\sqrt{2\pi}}\,\frac{e^{-\frac{1}{2}(\frac{0}{\sigma})^2}}{\sigma\sqrt{2\pi}} \\
& y = 0, \mu_y = 0, \sigma = \sigma_x, \sigma_y = \sigma, \mu = \mu_x \\
&= a\frac{e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}}{\sigma^2 2\pi}
\end{aligned}$$

4. Integrate the two-dimensional Gaussian and multiply by the measured wind velocity, v, to obtain plume flux in $\mathrm{mg\,S^{-1}m^{-2}}$. Multiplying this again by the number of seconds in a day, and the number of mg in a ton gives the flux in $t\,day^{-1}$.

$$\int \text{GaussianModel2D}() = a\int \frac{e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}}{\sigma^2 2\pi} = a$$

$$\text{flux}(a, v) = v\,a$$

Uncertainty in the flux calculation is given by the following root sum of squares method which combines the uncertainties in wind velocity ϵ_v, wind direction ϵ_d sensor error ϵ_s, and background CO_2 ϵ_b. The total uncertainty, ϵ, is calculated in accordance with the uncertainty estimation techniques described in [104, 95, 103, 89]:

$$\epsilon = \sqrt{\epsilon_v^2 + \epsilon_d^2 + \epsilon_s^2 + \epsilon_b^2}$$

4.5 Results

Flux estimates are derived from the 3 UAS transects that crossed plume A. These transects were collected on November 26th and 27th, 2021. Other transects shown in Figure 4.2 either did not intersect any plume or did not cross the entire plume. In the latter case this resulted in a poor fit to the Gaussian distribution, violating our assumption of normality. We also report carbon isotopes of plume CO_2, and flux estimates based on the Multi-GAS CO_2/SO_2 ratios.

Table 4.1: CO_2 data collected by UAS across plumes A and B during the Tajogaite eruption. * Indicates transect with samples collected into Tedlar bags and analyzed by Infrared Isotope Ratio Spectroscopy. † Indicates transects that encountered plume B, but the gas distribution did not meet our Gaussian fit assumptions, as indicated by the low R^2 value in comparison to the Gaussian amplitude. Thus we did not include plume B in our flux calculations.

Date	Transect	Altitude	Wind [$m\,s^{-1}$@°]	Max Con. [ppm]	Gaussian Fit Amplitude	R^2	Flux [$t\,day^{-1}$]
2021-11-26	2 Plume A	200 m	11.8 @ 68°	501	8.95×10^5	0.93	$1.76 \pm 0.20 \times 10^3$
2021-11-27	6 Plume A	100 m	12.2 @ 38°	616	1.10×10^7	0.71	$2.23 \pm 0.26 \times 10^4$
2021-11-27	7 Plume B †	100 to 250 m	12.2 @ 38°	613	3.02×10^6	0.01	$6.15 \pm 0.71 \times 10^3$
2021-11-27	8 Plume A	300 m	12.2 @ 38°	577	2.81×10^6	0.75	$5.71 \pm 0.66 \times 10^3$
2021-11-28	9 Plume B* †	300 m	11.3 @ 44°	963	3.85×10^7	0.36	$7.25 \pm 0.84 \times 10^4$

4.5.1 Plume Transect Wind Measurements

The calculated CO_2 flux for the 5 relevant transects with the corresponding wind velocities and directions are shown in Table 4.1 for transects across plume A and B. The wind velocity measured by UAS drift method was 10.7 ms^{-1}. ERA5 modeled wind velocities yielded results ranging from 10.0 to 12.2 ms^{-1} with an average of 11.1 ms^{-1}. The wind direction

Table 4.2: Measured CO_2 concentrations and $\delta^{13}C$ from ground and UAS.

Date	**CO_2 [ppm]**	**$\delta^{13}C$ VPDB ‰**	**Collection method/site**
2021-11-21	435	-7.46	Ground
2021-11-21	472	-8.34	Ground
2021-11-21	437	-7.65	Ground
2021-11-21	416	-8.00	Ground
2021-11-28	671	-4.44	UAS
2021-11-30	1030	-3.65	Ground
2021-11-30	2998	-2.12	Ground
2021-11-30	2863	-2.15	Ground
2021-12-01	4459	-2.03	Ground
2021-12-01	2722	-1.47	Ground
2021-12-01	1326	-2.40	Ground

given by the ERA5 model yielded results ranging from 38° to 68° with an average of 53°. These ranges contribute to the overall uncertainty ϵ_d

4.5.2 Carbon isotopes of plume CO_2

The CO_2 concentrations and $\delta^{13}C$ values of plume gas samples are given in Table 4.2. Samples collected from the ground at the UNM Multi-GAS site show background CO_2 concentrations 416 to 471 ppm CO_2 with $\delta^{13}C$ values of -8‰ (relative to Peedee belemnite) which is close to that of air. The sample collected by UAS has a CO_2 concentration distinctly elevated from air of 671 ppm and a heavier $\delta^{13}C$ value of -4.44 ‰. Samples collected from the ground closer to the vent have even higher CO_2 concentrations from 1030 to 4459 ppm with $\delta^{13}C$ values from -2.40 to -1.47 ‰.

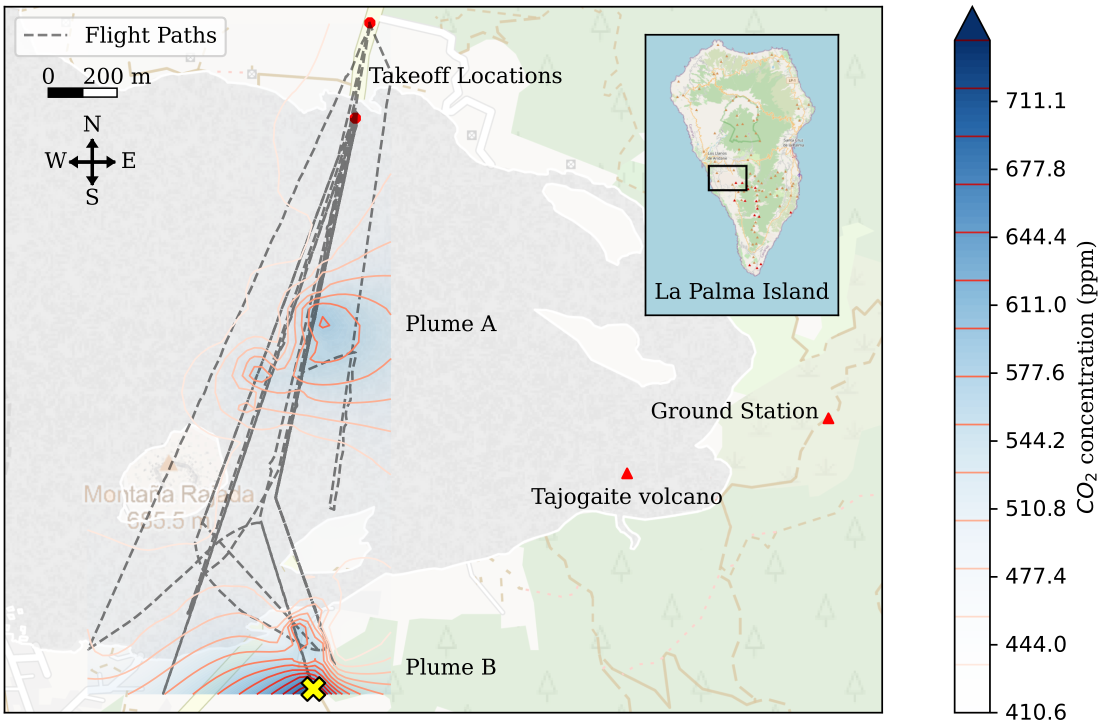


Figure 4.2: **Top-down perspective map of all transect flight paths.** Flights occurred over a four-day period during the 2021 eruption. This map includes a horizontal cross-section Kriging plot of the CO_2 concentration highlighted as the distinct Plumes A and B. The sample collection location is indicated by the yellow $\times$. Insert shows the location of Tajogaite Volcano on La Palma Island. Map images © OpenStreetMap contributors 2024. Distributed under the Open Data Commons Open Database License (ODbL) v1.0.

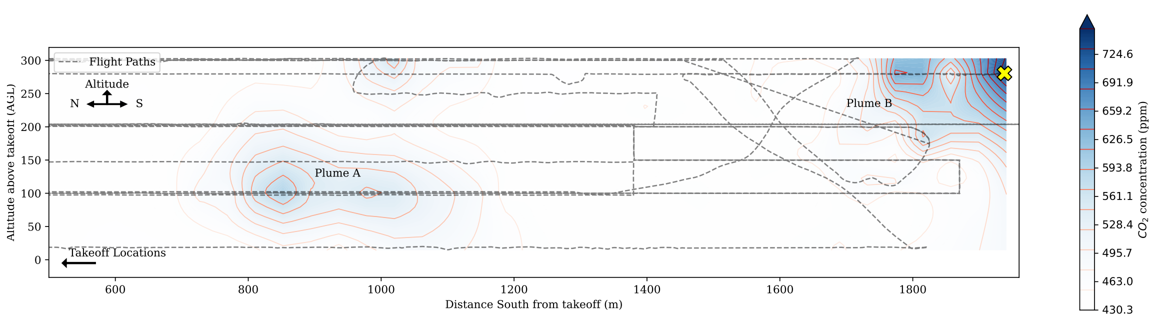


Figure 4.3: **Lateral perspective kriging map of all transects plotted in Figure 4.2.** The plot indicates two separate plumes in the vertical cross-section labeled Plume A and Plume B. The sample collection location is indicated by the yellow $\times$.

4.5.3 Multi-GAS measurements of plume

The Multi-GAS CO_2/SO_2 ratios during the period from November 21 to November 25, 2021 range from 5 to 26 and are shown in Table 2. These values are consistent with those reported by [15] and [34]. We use the range of reported SO_2 fluxes (mean of 10^4 t day^{-1} over the duration of the active eruption [15]) in combination with the range of our Multi-GAS CO_2/SO_2 ratios to obtain CO_2 fluxes ranging from 7.3×10^4 to 3.6×10^5 t CO_2 day^{-1} for this period (Table 4.3).

Table 4.3: Multi-GAS measurements, SO_2 flux and computed CO_2 flux .

Date	**Average CO_2/SO_2 (molar)**	**SO_2 flux (t/day)**	**CO_2 t/day**
2021-11-21	26 ± 15	$2 \pm 1 \times 10^4$	$3.6 \pm 1.8 \times 10^5$
2021-11-22	10 ± 2	$2 \pm 1 \times 10^4$	$1.4 \pm 0.7 \times 10^5$
2021-11-23	5 ± 2	$2 \pm 1 \times 10^4$	$7.3 \pm 3.7 \times 10^4$
2021-11-24	7 ± 2	$2 \pm 1 \times 10^4$	$9.5 \pm 4.8 \times 10^4$
2021-11-25	16 ± 2	$2 \pm 1 \times 10^4$	$2.3 \pm 1.1 \times 10^5$

4.6 Discussion

This work highlights our efforts collecting and analysing CO_2 gasses during the Tajogaite volcanic eruption. Through this work, we demonstrated the efficacy of using a UAS to study the CO_2 plumes associated with an in-process eruption.

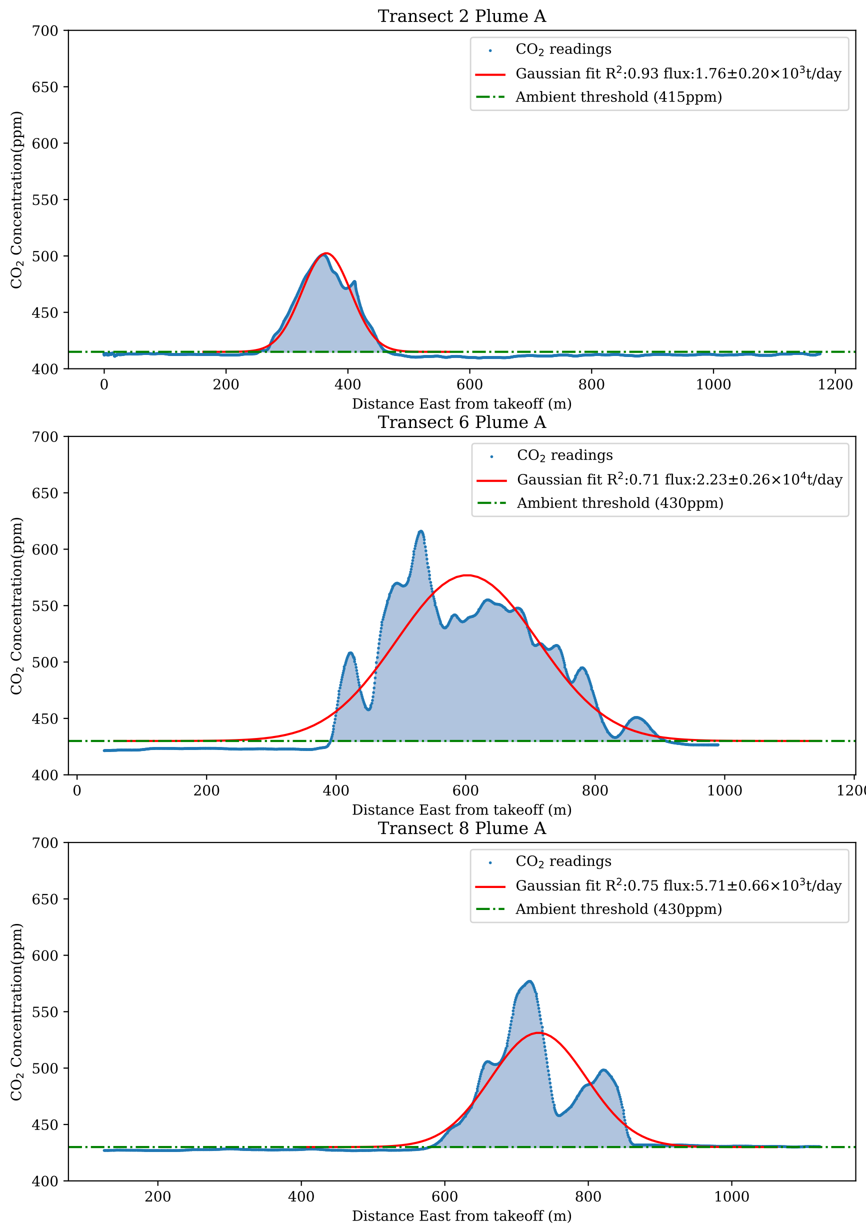


Figure 4.4: **Three plots of encounters with plume A with the closest Gaussian model fit.** CO_2 concentration (blue) over the encountered plume as a function of distance from takeoff location.

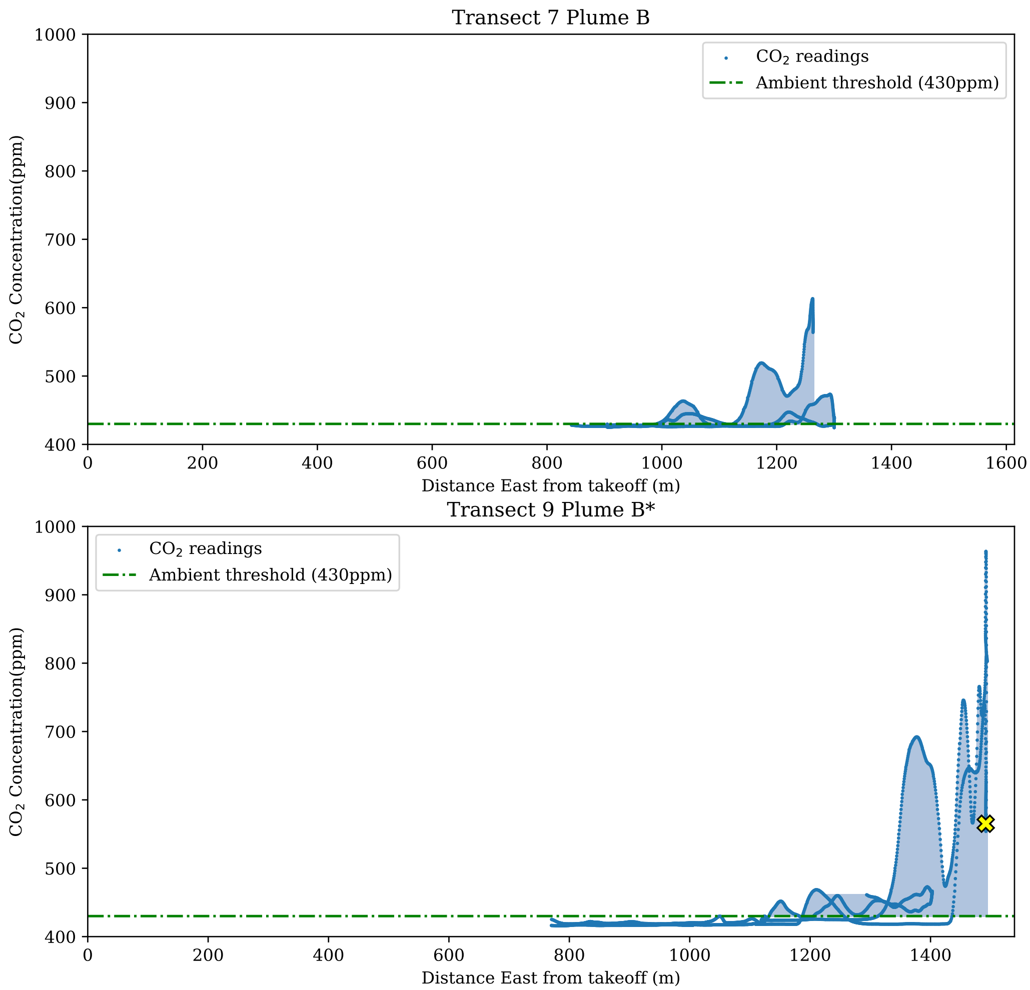


Figure 4.5: **Encounters with plume B were not as well-fit as plume A encounters.** These plots show the CO_2 readings collected during the two highest plume model fit. As with Figure 4.4, CO_2 concentration (blue) over the encountered plume as a function of distance from takeoff location. The sample collection location is indicated by the yellow $\times$.

4.6.1 CO_2 Emissions

Our UAS-based CO_2 emission estimation technique yields CO_2 fluxes using direct measurement with a single type of instrument. This simplifies the estimation of CO_2 flux. However, in-situ measurement during an active eruption is challenging. The most serious difficulty we encountered was obtaining complete transects across the plume or plumes. In several of our transects, especially for the more distant Plume B, we were not successful in flying the UAS far enough to get to background CO_2 on the far side of the plume. Gas plumes change shape and direction on relatively short-time scales as the wind shifts. While ideally, we would like to perform several flights at various altitudes through a plume in order to obtain a complete CO_2 concentration map of the plume, this is challenging for wide or distant plumes because of limited UAS flight times and the need to know the plume's location and extent a priori. To address this challenge we assume a Gaussian plume and fit a Gaussian curve to our data. We then rotate the Gaussian fit to obtain a 2D concentration slice which is multiplied with estimated wind velocity to yield the flux. This approach produces the most accurate results if we transect the plume through its widest part. However, identifying the widest part and then transecting the plume before the plume changes will require teams of collaborating UASs. A good fit of the data by the Gaussian model is given by a high R^2 value. For instance, transect 2 was fit with a R^2 value of 0.93 accounts for 93% of the variance in the observed data. The model fit represented by this high R^2 value is depicted in Figure 4.4.

Uncertainty is introduced by the assumptions made by the model. With just one horizontal transect, we assume the vertical Gaussian standard deviation is identical to the horizontal standard deviation of the plume. Both dimension standard deviations are linearly correlated to the flux calculation, meaning that a 20% error in the vertical standard deviation will affect the flux estimate by 20%. We estimate the vertical standard deviation is likely close to the horizontal standard deviation, but the difference is impossible to determine. Additionally, we assume that the horizontal transect samples the plume at the altitude where the plume is widest. If the transect is not through the largest cross-section, the flux calculation may be a lower bound. Wind velocity was measured during one of the transects, but weather is notoriously unpredictable. This represents another source of uncertainty in the model which has a linear effect on the flux measurement. We used our wind estimates during the time of each flux calculation. This variation in wind velocity ϵ_v is $\pm 11\%$ which is calculated from the wind velocity range measured over the experiments (Table 4.1). The range of wind directions is $\pm 15°$ from Table 4.1, which gives an error in the flux estimate based on $\epsilon_d = 1 - \cos(\text{angle})$, thus $\pm 3.40\%$. The SBA-5 documentation reports sensor error ϵ_s is 1% in the range of CO_2 we measured. Finally, background ambient CO_2 ϵ_b adds 1% to the uncertainty model which we calculated from the uncertainty in ambient CO_2 readings. Therefore, our estimated flux uncertainty given by the root sum of squares method is $\epsilon = \pm 11.61\%$.

Our data show that for Plume A, transect 6 (Figure 4.3) represents the widest plume and results in the highest CO_2 flux value of $2.23 \pm 0.26 \times 10^4$ t day^{-1}, an order of magnitude higher than the other two Plume A transects. This transect was flown at the lowest altitude (100 m) of the three, implying that the other two transects only captured the upper parts of the plume. Comparison with CO_2 fluxes obtained by combining SO_2 fluxes with CO_2 to SO_2 ratios measured 1 km from the vent gives fluxes ranging from 7.3×10^4 to 3.6×10^5 t CO_2 day^{-1} (Table 3). Therefore our highest flux measurement is consistent with the lowest estimate using the combined method. While comparing these two approaches is helpful, our experiment was not designed to make a direct comparison. The discrepancy could be due to a significantly varying CO_2 emission rate during eruptions, an overestimate of the SO_2 flux, or the lack of validity of the 2D Gaussian extrapolation approach. Our estimates are consistent with the October 2021 high emissions presented by Bruton et al., 2023 who report fluxes of 3.36×10^4 to 4.19×10^4 t CO_2 day^{-1} (389 to 486 kg/s) for the smaller, non-ashy plume that we measured. More work needs to be performed in the future to better assess sources of discrepancies with new and coordinated measurements at passively degassing and erupting volcanoes. However, even with such discrepancies, it is clear that the Tajogaite eruption in November 2021 produced a CO_2 flux up to 2×10^4 t day^{-1} or even 5×10^5 t day^{-1}. Even the 5×10^5 t day^{-1} would be only 0.4% of the daily CO_2 emitted by the burning of fossil fuels [43].

4.6.2 Carbon Isotopes

The carbon isotope data obtained from the UAS-captured samples and the samples collected from the ground are generally consistent and show mixing of air-derived CO_2 with a deep magmatic source. Figure 4.6 shows that all plume samples collected from the ground define a set of mixing lines in δ^{13}C versus CO_2^{-1} space, i.e. in a Keeling plot [91] that allows for the extrapolation of the δ^{13}C value of the pure CO_2 being emitted from the volcanic vent. The sample collected by UAV lies slightly above this set of mixing lines and extrapolates to somewhat heavier δ^{13}C. The resulting volcanic δ^{13}C values taking into account all samples lies between -1.5 and +1.5 ‰. Despite these uncertainties, these values overlap with δ^{13}C data obtained from mantle xenoliths erupted at the nearby El Hierro Volcano [126]. Extrapolation of all these data results in a δ^{13}C value of 0.1 ± 1.5‰. Notably the carbon isotope values are significantly heavier than those measured in cold CO_2-rich gas discharges from springs on La Palma [109] and within the range of values measured in olivines and pyroxenes of xenoliths from El Hierro Island [126]. These authors suggested that the

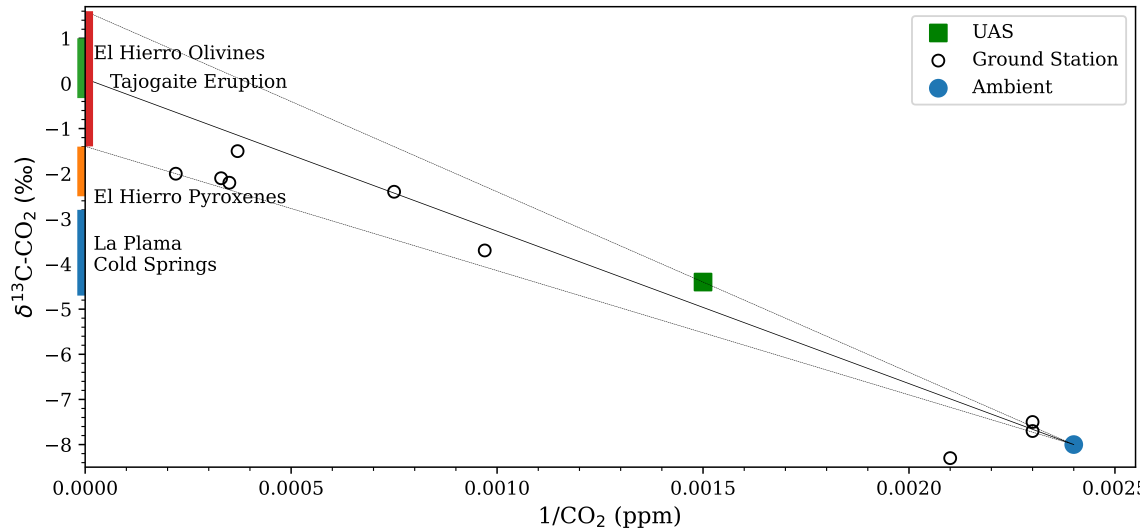


Figure 4.6: **Keeling plot showing standard air, samples collected on the ground, and with the UAS.** Linear extrapolation indicates a volcanic $\delta^{13}C - CO_2$ value of -1.40 to 1.60 ‰. Also shown are data from olivines and pyroxenes collected at the El Hierro Volcano [126] and the composition of cold CO_2-rich gas discharges on La Palma Island [109].

heavy values of the xenoliths are related to recycling of crustal carbon, likely derived from carbonates into the mantle source of the Canary Islands hot spot. Our data suggests that the magmatic system that is driving the Tajogaite eruption taps into this deep CO_2, rather than remobilizing CO_2 that feeds the cold degassing springs on the island. [125] report $\delta^{13}C$ values measured in olivines, clinopyroxenes and orthopyroxenes from lava flows erupted in 2021. Their data is consistent with our extrapolated heavy $\delta^{13}C$ values. For olivines, representing the earliest crystallization phase, their values range from 0 to 1‰. Values are somewhat lighter for orthopyroxenes and clinopyroxenes. Using all data, their estimated mantle endmember is -1.5‰. Our data extrapolate to -1.4 to +1.6‰.

Given the difference in sample medium, i.e. phenocrysts versus gas plume, the results are remarkably consistent. More work at erupting volcanoes is needed to better constrain the sources of magmatic CO_2 emitted during heightened activity of volcanic systems.

4.7 Conclusion

The use of UAS is revolutionizing volcano science by enabling the collection of data that previously required extensive, costly, and hazardous aerial surveys using piloted fixed-wing aircraft or helicopters. Especially in the field of volcanic gases, recent UAS-based campaigns showed the value of utilizing UAS to make gas flux and gas composition measurements and also collect plume samples for subsequent chemical and isotopic analyses [96, 73]. Our work during the explosive and hazardous eruption of the Tajogaite Volcano shows that CO_2 emission measurements and plume gas samples can be collected even during these heightened periods of volcanic activity. We demonstrate that a UAS capable of automated sampling can be guided by the expert knowledge of scientists in the field to collect valuable data that would be impossible with robots or scientists alone. The collected data provide key insights into the volcano's state and the course of an eruption. Future work is needed to increase UAS autonomy in choosing flight paths to more completely capture data from dynamic plumes, but, as we have demonstrated, the present approach works for volcano monitoring during eruptions and can provide much-needed information about eruptive gas emissions.

Code availability Additional data and plot code is available at:
https://github.com/BCLab-UNM/lapalma-expedition
UAS code available at:
https://github.com/BCLab-UNM/dragonfly-dashboard
https://github.com/BCLab-UNM/dragonfly-controller

Author contribution Author contributions: JE, GMF, SN, and TF (UNM VolCAN team) performed UAS fieldwork for this paper. JE, NP, PHP, EPG (INVOLCAN team) and TF conducted the ground fieldwork. JE developed UAS software and hardware supervised by GMF and MEM. SN designed the sample collection device. JE, NP, PHP, EPG performed data analysis. TF performed isotope and gas analysis. JE, TF, GMF, SN, and MEM wrote the manuscript.

Competing interests The authors declare that they have no competing interests. The authors give consent for publication. All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials.

4.8 Acknowledgements

JE support provided by the Department of Energy's Kansas City National Security Campus, operated by Honeywell Federal Manufacturing & Technologies, LLC under contract number DE-NA0002839. GMF, SN, TF, RF, and MM support provided by the VolCAN project under National Science Foundation grant 2024520. Support was also provided by a Google CSR award. This study received funding from Google and Honeywell Federal Manufacturing & Technologies, LLC. VOLRISKMAC II (MAC2/3.5b/328) financed by the Program INTERREG V A Spain-Portugal MAC 2014-2020 of the European Commission. Thanks to Samantha Wolf for help calculating the flux.

Chapter 5

Navigating the Edge: UAS Boundary Tracing for Efficient Volcanic Plume Monitoring

5.1 Publication Notes

Citation: Ericksen, John, et al. ”Navigating the Edge: UAS Boundary Tracing for Efficient Volcanic Plume Monitoring.” 2024 IEEE International Symposium on Safety Security Rescue Robotics (SSRR). IEEE, 2024.

Publication date: 12 November 2024

Conference: 2024 IEEE International Symposium on Safety Security Rescue Robotics (SSRR)

Publisher: IEEE

Formatting: The original published text has been preserved as much as possible while still adhering to the formatting requirements of this dissertation.

Funding: This work was supported by funding from the following: JE support provided by the Department of Energy's Kansas City National Security Campus, operated by Honeywell Federal Manufacturing & Technologies, LLC under contract number DE-NA0002839. GMF, SN, TF, RF, AI, JS, KA and MM support provided by the VolCAN project under National Science Foundation grant 2024520.

5.2 Abstract

We present the implementation and validation of SKETCH, an algorithm that uses two UASs to trace the boundary of volcanic plumes. SKETCH guarantees asymptotically optimal flight distance and turning by maintaining a *sandwich invariant* where one UAS stays inside the plume boundary (defined by a CO_2 concentration threshold) and the other UAS stays outside. The UASs adjust their flight paths based on real-time CO_2 measurements to maintain this invariant. This paper details the implementation of SKETCH on a real-world UAS platform, the Dragonfly drone. We evaluate the efficacy of SKETCH through extensive testing in physics-based simulations and real-world outdoor environments using virtual plumes. The algorithm is compared to a single-UAS baseline algorithm called

ZigZag. Results show that sketch meets the expectations set by theory, and it is more efficient than ZigZag, achieving shorter flight paths, less turning, and faster mapping times. While ZigZag exhibits slightly higher accuracy in estimating plume area and boundary, sketch offers a more efficient real-time volcanic plume monitoring approach, especially in time-sensitive situations. These results demonstrate the feasibility and efficacy of sketch for real-time volcanic plume monitoring, paving the way for accurate CO_2 emission estimation in hazardous and challenging environments.

5.3 Introduction

Volcanic eruptions cause widespread devastation and loss of life. Monitoring volcanic gas emissions, especially carbon dioxide (CO_2), is a crucial tool for predicting these eruptions [69]. Measuring volcanic CO_2 emissions also contributes to models of climate change [57]. We tackle this challenge by developing UASs and associated algorithms capable of efficiently mapping volcanic plumes.

Volcano gas monitoring requires safer and more efficient methods. The CO_2 plume is invisible, not co-located with visible ash plumes, and usually difficult or dangerous to access from the ground. Satellite and ground-based remote sensing of volcanic CO_2 is extremely limited [137] or relies on in situ SO_2 proxies [36, 89]. Even the most recent NASA satellite hyperspectral cameras have relatively low resolution (10.00 ppm of CO_2) compared in situ sensors that can be integrated into a UAS.

The VolCAN project is an interdisciplinary effort among computer scientists, geologists, and computer engineers that aims to revolutionize the study of volcanic gases using UASs. In previous field studies we have characterized volcanic CO_2 emissions in Tavurvur in Papua New Guinea [73], Tajogaite in La Palma [57], multiple eruptions in Reykjanes, Iceland [69], and CO_2 the Valles Caldera supervolcano in New Mexico, USA [58].

We designed and field-tested the SKETCH algorithm to identify the boundary of volcanic gas plumes. We define a *plume* as the set of points with a CO_2 concentration threshold above some predetermined concentration. The *boundary* (or edge) of the plume is the polycurve containing this set of points. Finding the plume boundary is crucial since it gives both the location and cross-sectional area of the volcanic plume. SKETCH (first described in [46] and further detailed in [47]) is an efficient boundary tracing algorithm using two UASs flown in tandem. SKETCH guarantees asymptotically optimal flight distance and turning. In addition, it does not assume an unrealistically maneuverable UAS (instantaneous response time to sensor readings) , and so it is adaptable to drones with a wide range of specifications.

SKETCH operates by navigating along the boundary, maintaining a *sandwich invariant*: one UAS maintains a location with a CO_2 concentra-

Figure 5.1: **A UNM VolCAN Dragonfly drone flying into the plume of the actively erupting Litli-Hrútur volcano in Iceland.** This expedition involved measuring CO_2 concentrations across multiple transects of the plume to build a plume model and estimate the flux of the eruption.

tion greater than the threshold, and the other UAS maintains a location with a concentration lower than the threshold. Both move perpendicular to the concentration gradient. If a UAS crosses the boundary and the invariant is invalidated, then the UASs collaboratively turn towards the boundary to reestablish the invariant. This last step is carefully designed so that the total turning of both drones over the course of the algorithm asymptotically equals the total curvature and length of the boundary.

The main contribution of this work is to demonstrate an implementation of the SKETCH algorithm in the DragonFly UAS platform; a platform that we have successfully used at multiple active eruptions. This bridges the well known and often challenging *reality gap* between theory and implementation. We validate the efficacy of SKETCH through comprehensive testing in physics-based simulations and demonstrate feasibility in UASs in real-world outdoor environments using virtual plumes, showcasing its ability to accurately trace complex plume boundaries under varying conditions. These contributions advance the state-of-the-art in collaborative UAS environmental monitoring and provide a foundation for future research and development.

5.4 Related Work

Recent advances in UAS technology have led to significant improvements in environmental monitoring, boundary detection, and volcanic plume mapping. UASs can cover large areas, collect data from hard-to-reach places, and perform tasks with high efficiency and accuracy, and thus find applications in many domains [88]. In this section we focus on related work on environmental monitoring with UASs.

Determining the boundary of a region has many practical applications, for example: mapping pollution sources such as chemical spills and emissions [121], radiation hazards [81], agriculture [116] and volcanic plumes. Sung et al. [139] provide a survey of decision-theoretic approaches for robotic environmental monitoring.

Facinelli et al. [61] describe challenges in gas plume detection in industrial areas using coordinated UASs . Ghamry et al. [77] investigate strategies for forest monitoring and fire detection leveraging the combined

Algorithm 2 Initially, UASs D_1, D_2 are $\sqrt{\lambda}$ apart; one inside and one outside

procedure SKETCH(λ) ▷
$\nabla \leftarrow$ boundary gradient; SketchTerminate $\leftarrow$ **false**;
  while SketchTerminate $=$ **false do**
    if inside (D_1) XOR inside (D_2) **then**
      Move λ distance in the direction of ∇
    end if
    if not inside (D_1) and not inside (D_2) **then**
      CROSS-BOUNDARY($D_1, D_2, -\sqrt{\lambda}$)
    elseif inside (D_1) and inside (D_2)
      CROSS-BOUNDARY ($D_2, D_1, \sqrt{\lambda}$)
    end if
  end while
end procedure

capabilities of UASs and UGVs for enhanced efficiency. Additionally, Euler et al. [60] describe an adaptive sampling strategy for efficient spatial mapping in large-scale environments, through cooperative UASs. Assenine et al. [22] developed a cooperative deep reinforcement learning approach that focuses on real-time monitoring of pollution plumes using a fleet of drones equipped with advanced sensing technology. Rossi and Brunelli [120] describe a team of UASs equipped with electronic noses for effective gas detection and mapping. Karbach et al. [90] use UAS to observe volcanic plume chemistry with ultralight sensor systems. Asadzadeh et al. review [21] the state-of-the-art in UAS-based remote sensing for the petroleum industry and environmental monitoring. Saldaña et al. [124] approximate boundaries of a 2D surface oil slick using aquatic robots taking pointwise measurements.

In comparison to these earlier results which make strong assumptions

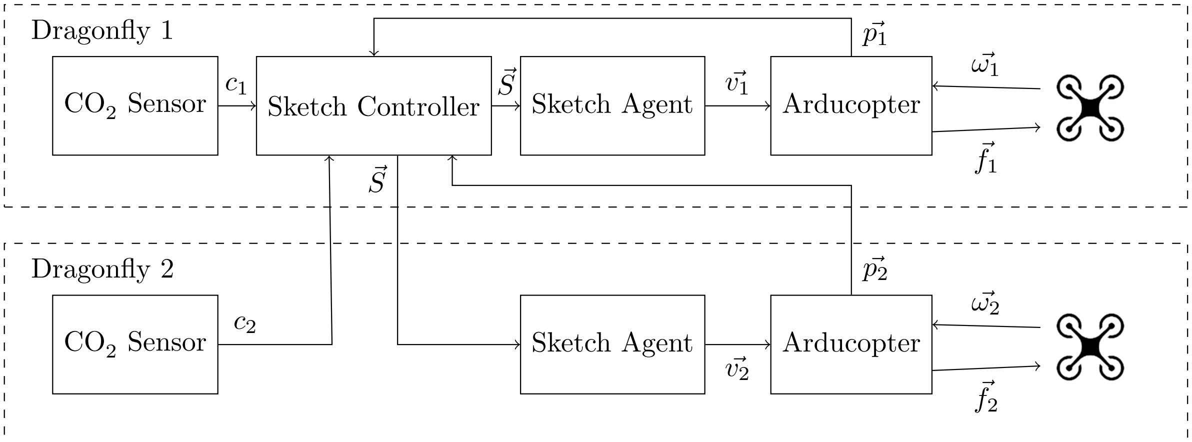


Figure 5.2: **System level diagram of the Dragonfly sketch implementation.** The SKETCH algorithm is executed on the lead drone (Dragonfly 1) within the SKETCH Controller, which directs both itself and the follower drone's (Dragonfly 2) Sketch Agent to direct the flight path of each drone. The communication between the SKETCH Controller and the SKETCH Agents leverages the multi-agent-oriented ROS2 DDS infrastructure. Vector flight directives are issued from the SKETCH Agent to the Arducopter flight controller through the LOCAL_POSITION/VELOCITY ROS topic.

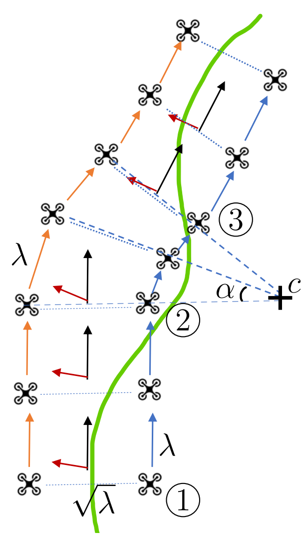


Figure 5.3: **Illustration of Algorithm 3.** At ①, the UAS pair is sandwiching the edge of the plume in green. At ② a UAS has crossed the boundary, and so the UASs perform a series of turning steps to reestablish the sandwich invariant (via the CROSS-BOUNDARY subroutine). At ③ the UASs sandwich the boundary again, and so move in a straight line perpendicular to the last measured boundary gradient.

on the boundary shape (e.g. convexity or star-convexity [82, 86]), our problem is both easier and harder. It's easier because we assume a largely static boundary; it's harder because we make fewer assumptions on the boundary shape. Furthermore, in contrast to most of the earlier results which use a single agent, we deploy two UASs in order to handle arbitrary boundary shapes. Rather surprisingly, the use of a UAS pair does not impact algorihmic efficiency. In fact, our theoretical results guarantee optimal distance traversed and angle turned by the UASs, while also ensuring precise estimation of the boundary. Our prior lab experiments [56] and the field experiments and simulations we present here support these theoretical guarantees.

5.5 Methods

5.5.1 sketch Algorithm and Implementation

We implement SKETCH (Algorithm 3; illustrated in Figure 5.3; see also [47] for details) with a two-level architecture depicted in Figure 6.3. The top level is the Sketch Controller, which runs on the leader drone (DragonFly 1 in the figure) and processes the positions and CO_2 readings from both UASs. The Sketch Controller decides whether to fly straight or turn towards the plume boundary. These decisions are sent as commands to the lower-level Sketch Agents which run on both the leader and follower drones. The Sketch Agents execute these commands, controlling the flight dynamics using velocity vector commands through the flight control computer.

The Sketch Agent flight dynamics are implemented using a flocking-style algorithm, which maintains alignment, cohesion, and separation between the two UASs through a linear combination of velocity vectors. The linear vector sum is given as:

$$\vec{v_i} = c_s\vec{v_s} + c_t\vec{v_t} + \Delta t(c_{to}\vec{a_{to}} + c_{ta}\vec{a_{ta}} + c_e\vec{a_e}) \tag{5.1}$$

In this equation:

- $\vec{v_s}$ is the straight-line velocity vector.
- $\vec{v_t}$ is the turning velocity vector.
- $\vec{a_{to}}$ is the tandem-offset acceleration vector.
- $\vec{a_{ta}}$ is the tandem-alignment acceleration vector.
- $\vec{a_e}$ is the error-correction acceleration vector.
- $\vec{\delta t}$ is the update interval in seconds for the $\vec{v_i}$ update. In our experiments $\vec{\delta t}$ is set to a constant 10.00 Hz.

Straight-line and turning vectors are applied based on the Sketch Controller's command vector $\vec{S}$. The constants c_s, c_t, c_{to}, c_{ta}, and c_e are gain scalars used to tune the dynamics of the algorithm.

To trace the boundary of a volcanic plume, SKETCH relies on the concept of a concentration gradient. While SKETCH assumes the gradient is provided by an oracle, in practice the gradient is calculated from readings collected along the flight paths. We evaluated several techniques for calculating the gradient, including fitting a line from three points near the plume crossing, using the previous 100 sample data points, and limiting data points to those within a distance of λ from the current position. Our results indicated that the approach of limiting data points within λ distance produced the most accurate gradient estimation compared to the oracle. This method balanced the smoothing effect by averaging multiple points while maintaining proximity to the current gradient state.

5.5.2 ZigZag Algorithm

For comparison, we implement a single UAS boundary-following algorithm called ZIGZAG that incorporates λ that defines the same turning behavior as SKETCH. ZIGZAG is intended to illustrate the advantages of using two UAS that maintain the sandwich invariant; it is not intended to be an optimal algorithm. ZIGZAG is conceptually similar to other line-following robot algorithms (e.g. [128]) that use sensors to determine whether a line is to the left or right of the robot. ZIGZAG turns clockwise when the robot crosses from a concentration below the boundary to one above the

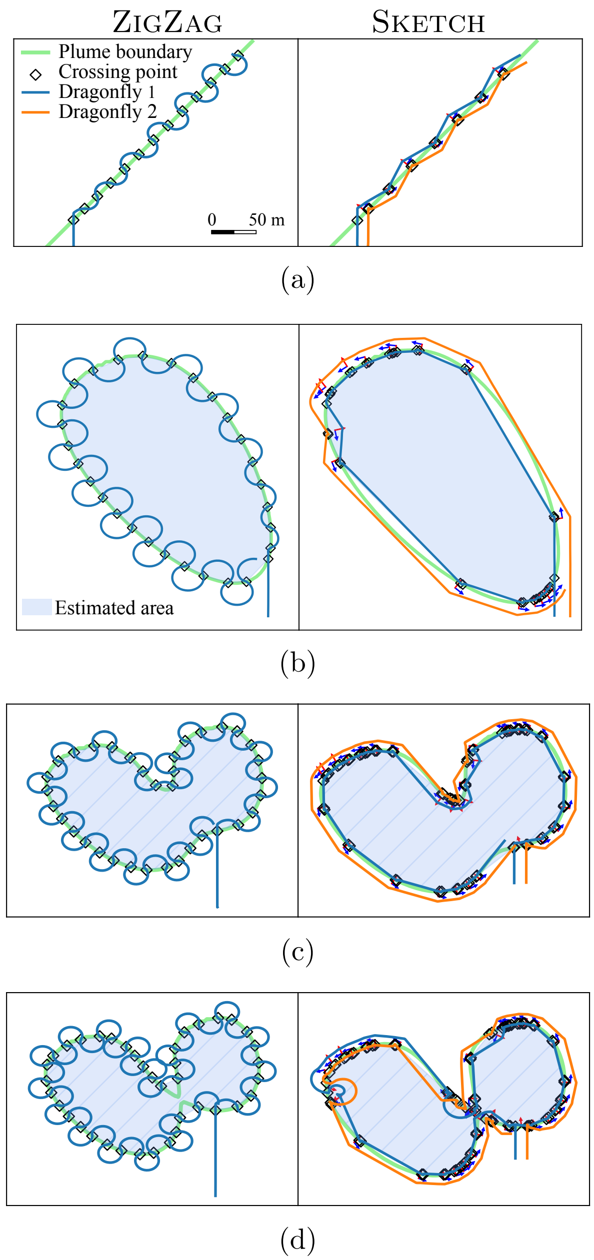


Figure 5.4: **Flight path simulations of UASs executing the ZigZag algorithm (left) and the sketch algorithm (right).** Scenarios include: (a) Straight line, (b) Single plume, (c) Double plume, and (d) Dumbbell configuration. Diamond markers indicate crossing points, which are connected to estimate the plume area, shown with a hashed blue fill. Both algorithms accurately estimate plume areas, but SKETCH completes the circumnavigation 1.5 to 2 times faster.

boundary and anticlockwise otherwise. This results in a looping path that intersects the boundary. The distance the UAS can travel from the plume is determined by the radius of the turning angle.

$$\textsc{ZigZag}(v, c, \lambda) = \begin{cases} \textsc{TurnRight}(v, \lambda) & \text{if c > threshold} \\ \textsc{TurnLeft}(v, \lambda) & \text{otherwise} \end{cases} \tag{5.2}$$

5.6 Experiments

To validate SKETCH, we conducted a series of experiments in both Gazebo[1] and a real-world field test. Because it is impossible to generate a large CO_2 plume at our field site, we tested SKETCH on a virtual plume simulated at the field site. This virtual plume is generated using a location-based equation based on a smeared Gaussian distribution. We conducted experiments with plumes that vary in size, shape, and CO_2 concentration gradients, providing a comprehensive assessment of the boundary-tracking capabilities of SKETCH.

To evaluated the algorithms, we focus on the following metrics: 1) Distance maintained from the plume boundary using two parameters: the Fréchet distance and the average Hausdorff distance, 2) Percent error between the ground truth plume area and the estimated area of the polygon formed by the UASs' plume-crossing points, 3) Amount of turning by the

[1] Gazebo simulation: https://www.youtube.com/watch?v=-FzHBOoKdXE

UASs, 4) UAS path length, and 5) UAS flight time. The Fréchet distance [16] measures the maximum divergence between the UAS paths and the plume boundary, assessing the worst-case separation, which is then compared to the theoretical $8\sqrt{\lambda}$ guarantee. In contrast, the average Hausdorff distance [140] measures the difference between the ground truth and the estimated plume boundary.

We simulated SKETCH and ZIGZAG in Gazebo across four cases: a straight-line threshold, a simple single-plume model, a double-plume model, and a challenging double-plume scenario with a narrow neck (dumbbell case). Finally, we conducted field tests of SKETCH using physical Dragonfly UASs capable of tracing volcanic plumes in real-world environments (Figure 5.5). These tests focused on validating the algorithm's performance on physical UASs. We collected data on the UASs' flight paths, CO_2 readings, and their ability to track the plume under real-world conditions.

5.7 Results

Each scenario in Gazebo was simulated once for both SKETCH and ZIGZAG, except the double-plume scenarios, which were tested over 30 trials per algorithm. Furthermore, single and double-plume scenarios for SKETCH were flown using the physical Dragonfly UASs. Figure 6.4 displays the ground truth plume, flight paths, crossing points, and the estimated plume

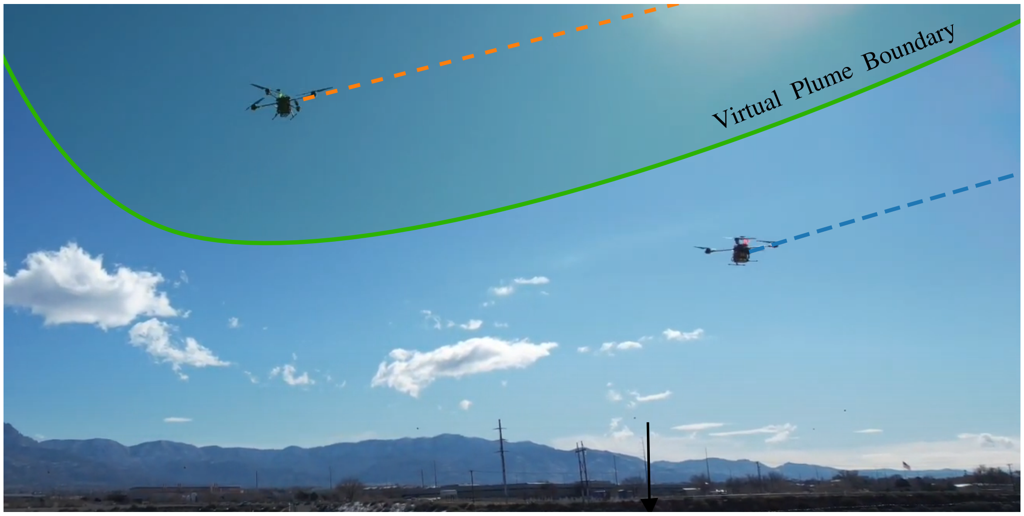


Figure 5.5: **sketch executed using two DragonFly UAS to map a virtual plume boundary.** The pair of drones adapt their flight path to maintain the sandwich invariant.

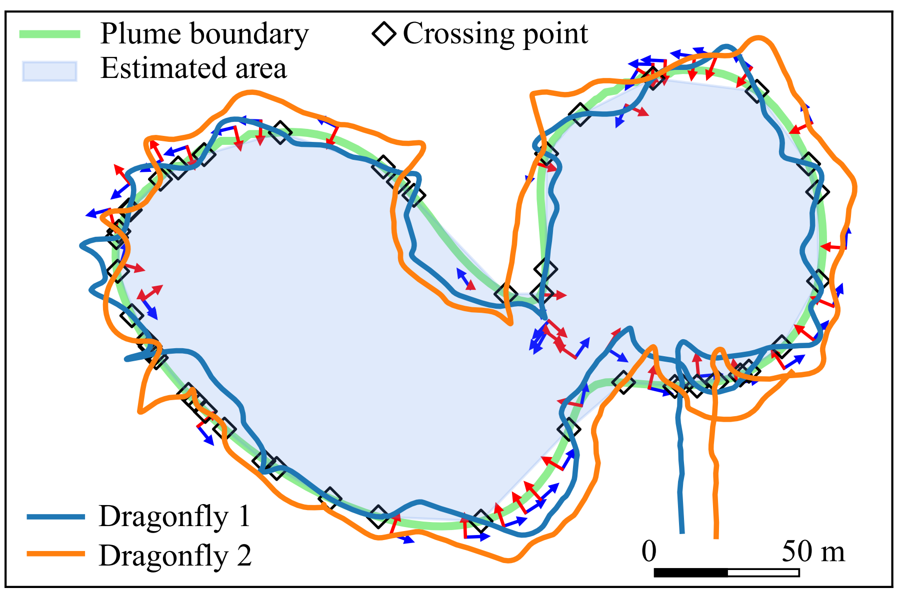


Figure 5.6: **Flight paths of physical UASs executing sketch.** Flight paths of the Dragonfly UASs at Balloon Fiesta Park in Albuquerque, NM, showing vectors $\vec{S}$ and the total plume area. $\vec{S}$ includes the blue arrow indicating the forward or turning movement, and the red arrow indicating the calculated gradient.

Table 5.1: Comparison of Sketch and ZigZag.

Plume	Algorithm	Platform	Fréchet Distance (m)		Average Hausdorff Distance (m)	Plume Area Estimation Error (%)	Amount of Turning (°)		Path Length (m)		Flight Time (s)
			UAS1	UAS2			UAS1	UAS2	UAS1	UAS2	
Straight line	ZigZag	Gazebo	15.20	-	2.71	-	2351.07	-	400.12	-	262.00
	Sketch	Gazebo	9.97	9.94	3.71	-	1382.83	1346.79	257.26	251.58	172.99
Single plume	ZigZag	Gazebo	18.69	-	2.45	1.15	6659.43	-	1141.88	-	758.58
	Sketch	Gazebo	9.77	10.77	9.46	8.66	3228.91	1764.25	525.22	573.36	377.97
	Sketch	Hardware	11.31	15.05	5.59	3.67	2028.38	1844.16	535.33	565.82	200.53
Double plume	ZigZag	Gazebo	18.47	-	2.55	1.99	8271.46	-	1428.57	-	947.55
	Sketch	Gazebo	14.31	13.25	3.79	3.87	5503.37	4557.33	708.69	737.39	518.09
	Sketch	Hardware	18.88	15.90	3.14	4.03	5697.20	5456.32	822.53	861.64	341.15
Dumbbell plume	ZigZag	Gazebo	19.38	-	2.59	3.67	10 357.14	-	1800.72	-	1196.33
	Sketch	Gazebo	14.57	24.61	3.69	4.65	6976.93	5909.99	800.95	800.08	602.00

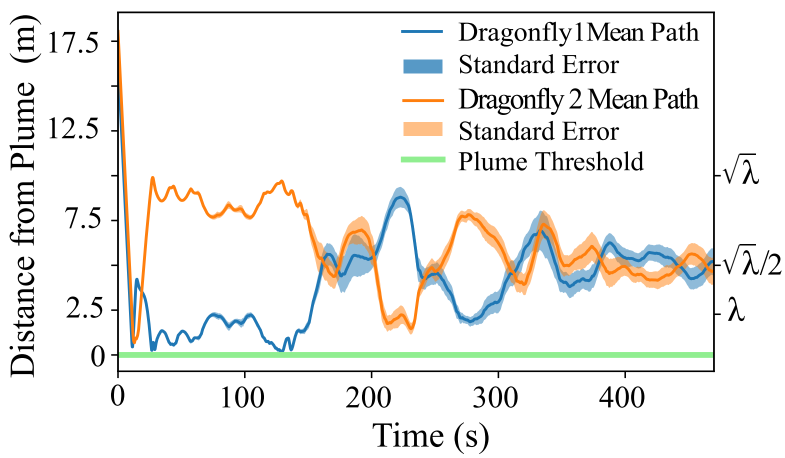


Figure 5.7: **Graphical representation of the two Dragonflys' distance from the double plume threshold across 30 trials.** The distance measurements highlight the UASs' ability to consistently maintain proximity to the plume edge, with values consistently within the expected theoretical limit of $8\sqrt{\lambda}$.

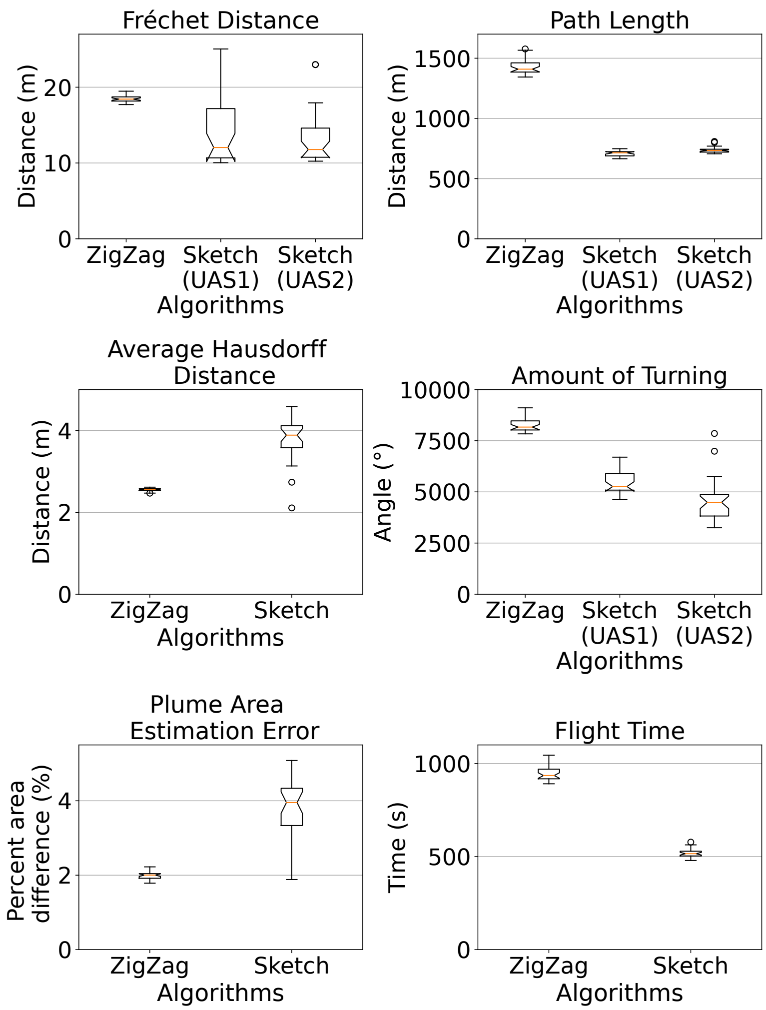


Figure 5.8: **Statistical comparison between sketch and ZigZag** over 30 simulation trials on Gazebo for the double plume case. Sketch circumnavigates the plume faster due to the shorter path length with less turning. The Error introduced by the lack of interaction with the plume is minimal while the Fréchet distance from the edge of the plume is minimal and within theoretical bounds.

boundary for the Gazebo simulations of both SKETCH and ZIGZAG across the four scenarios. Meanwhile, Figure 5.6 presents these components for SKETCH conducted with physical Dragonfly UASs for the double plume case.

Figure 5.7 displays the distance from the plume boundary across 30 trials of the double plume Gazebo simulation. The UASs maintained an average distance from the boundary of 5.58 m with a standard deviation of 4.56 m and a maximum distance of 25.04 m, aligning with the theoretical prediction of being within $8\sqrt{\lambda}$ (Theorem 1 in [47]), or 80.00 m, where λ is a parameter that is scaled with respect to the plume having a unit diameter (see Section 1.1 in [47]). Minimizing this distance ensures that the UASs accurately traces the boundary.

Table 5.1 summarizes the performance metrics for each scenario. All values in this paper are displayed to two-significant digits. For the double plume cases, the mean values are shown in the table while a statistical comparison is presented in Figure 5.8 within the 30 simulation trials. In all cases, SKETCH is at least 1.5 times faster than ZIGZAG. Additionally, SKETCH demonstrates a shorter path length, lower amount of turning, and closer proximity of the UASs to the plume boundary (lower Fréchet distance), whereas ZIGZAG achieves higher accuracy in plume area and boundary estimation. The benefit of completing the map in less time is obvious in the case of mapping an active volcano. Aditionally, quadcopter turns demand more energy than straight-line flight (e.g. [107]), making reduced turning beneficial.

A class of plumes we hypothesized would be challenging for SKETCH are ones with very narrow necks [e.g. Figure 6.4 (d)]. These plumes have two different sections of the boundary that are within $4\sqrt{\lambda}$ of each other, which if allowed to appear arbitrarily in general shapes would make the boundary topologically degenerate and hence was ruled out as an assumption in our theoretical analysis. However, for this specific and challenging case, in practice we found that UASs executing SKETCH were able to robustly navigate the boundary.

5.8 Conclusion

Our experiments demonstrate that SKETCH effectively navigates plume boundaries in both simulated and real-world environments. Compared to the single-UAS ZigZag algorithm, SKETCH demonstrates superior efficiency, achieving shorter flight paths, less turning, and faster completion times. This efficiency is crucial for real-time monitoring, especially in dynamic environments such as active volcanoes. The Dragonflies performed comparably to their simulated counterparts in Gazebo, maintaining stable flight while accurately tracking the plume boundary. This confirms the viability of SKETCH for field deployment to estimate the area of elevated CO_2 plumes. Combined with our earlier work demonstrating flights through erupting volcanic emissions, these findings will allow scientists to efficiently estimate CO_2 plume sizes. This advancement in UAS-based plume monitoring, facilitated by SKETCH, has the potential to significantly improve the accuracy and efficiency of CO_2 emission estimation, leading to a better understanding of volcanic activity and its environmental impact.

Acknowledgment

JE support provided by the Department of Energy's Kansas City National Security Campus, operated by Honeywell Federal Manufacturing & Technologies, LLC under contract number DE-NA0002839. GMF, SN, TF, RF, AI, JS, KA and MM support provided by the VolCAN project under National Science Foundation grant 2024520.

Chapter 6

The Dynamic Duo: Sketch Boundary Mapping Executed by Drones

6.1 Publication Notes

Publication date: Submitted

Journal: IEEE Robotics and Automation Letters 2025

Formatting: The original published text has been preserved as much as possible while still adhering to the formatting requirements of this dissertation.

Funding: This research was supported by the VolCAN project under National Science Foundation grant 2024520. Additional support to JE was provided by the Department of Energy's Kansas City National Security Campus, operated by Honeywell Federal Manufacturing & Technologies, LLC under contract number DE-NA0002839.

6.2 Abstract

Mapping the boundary of a region is a robotics challenge with broad application. Here we apply the SKETCH algorithm to this problem using aerial robots to map a simulated gas plume. SKETCH has several advantages over previous work in that it: (1) guarantees optimality for flight time, energy consumption, and accuracy; (2) provably handles any boundary that is a curvilinear polygon with a finite number of discontinuities and measurable gradient; this includes practically all physically realizable boundaries; and (3) provides a single controlling parameter that is directly related to robot dynamic limitations; this makes SKETCH simple to deploy on a real robot platform. We demonstrate that SKETCH can accurately map boundaries through indoor experiments using the Crazyflie robot platform.

6.3 INTRODUCTION

Determining the boundary of a region has many practical applications. For example, mapping pollution sources such as chemical spills and emissions, radiation hazards, and agriculture [121, 81, 116]. The application we focus on is mapping the boundary of volcanic gas plumes. This application is motivated by the scientific need to estimate the flux of carbon dioxide (CO_2) to predict eruption dynamics and as input to climate models [65].

We and others have found that UASs are useful tools for measuring CO_2 emissions by active volcanoes in the field [67, 65, 98, 72]. In our prior work [**?**, 68] we found that determining the cross sectional area of volcanic CO_2 plumes is the key measurement needed to complete flux estimates. We developed the SKETCH algorithm to trace plume boundaries to estimate plume cross-sectional area and, ultimately, volcanic CO_2 flux.

The boundary mapping problem we address involves approximating a closed boundary (defined by a threshold value). The SKETCH algorithm uses two coordinated drones that maintain a "sandwich invariant" by keeping the boundary between them. When this invariant breaks, a recovery procedure reestablishes it by turning the pair of drones back towards the boundary. SKETCH requires only binary inside/outside information driven by in-situ sensor readings, maintains provable error bound, and guarantees optimal travel distance and rotation.

The SKETCH algorithm guarantees accurate boundary tracing of the plume with optimal amount of turns and distance traveled by the drones [46, 47]. Here, we use ROS2 [101] to implement the SKETCH algorithm and empirically test its performance using a Crazyflie-based UAS hardware platform [26].

SKETCH depends critically on a parameter λ, which describes both the

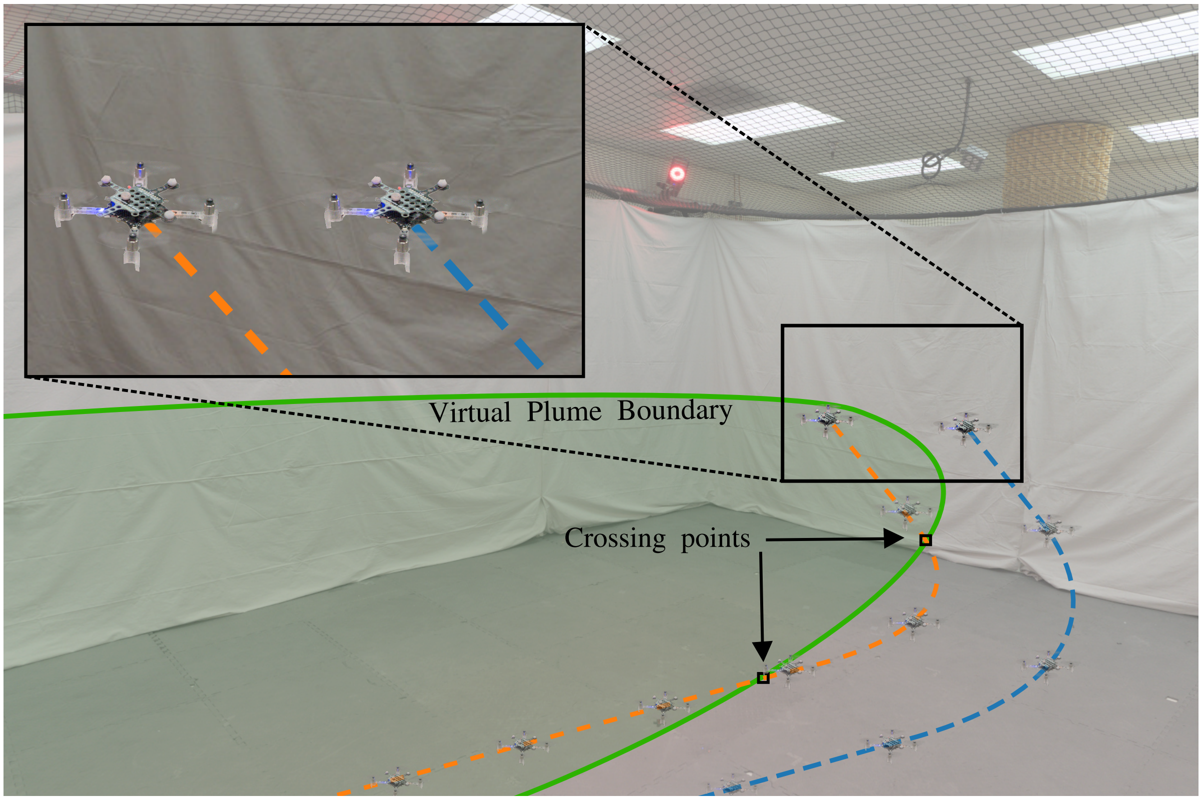


Figure 6.1: Schematic of SKETCH algorithm trials on Crazyflie UASs in the VICON lab environment. The Crazyflie UASs detect if they are inside or outside the virtual plume boundary and, following the SKETCH algorithm, fly straight while sandwiching the boundary or turn into the boundary if both UASs are inside or outside the plume.

smoothness of the plume boundary and the reaction time of the robots i.e. their dynamic limitations. Additionally, the accuracy of SKETCH is guaranteed in terms of λ, thus providing a natural alignment between its output and the robot hardware required to produce that output. By setting λ appropriately for the robot capabilities, the boundaries can then be mapped without having to resort to ad-hoc heuristics or secondary mechanisms. This means that SKETCH's theoretical guarantees survive the reality gap and the proofs are maintained, preserving the relationship between λ and accuracy in the real world.

6.3.1 Related Work

Recent advancements in UAS technology have led to significant improvements in environmental monitoring, boundary detection, and volcanic plume mapping. These areas benefit greatly from the use of cooperative UASs due to their ability to cover large areas, collect data from hard-to-reach places, and perform tasks with higher efficiency and accuracy [88]. In this subsection we discuss some related work on environmental monitoring with UASs.

Some challenges of detecting and localizing gas plumes in industrial areas using a coordinated approach among multiple UAS are presented in [61]. Ghamry, Kamel, and Zhang [77] investigate strategies for forest monitoring and fire detection leveraging the combined capabilities of UAS and UGVs for enhanced efficiency. Also, an adaptive sampling strategy for efficient spatial mapping in large-scale environments through cooperative UAS is introduced in [60]. Assenine et al. [22] describe a cooperative deep reinforcement learning approach that focuses on real-time monitoring of pollution plumes using a fleet of drones. A team of UAS equipped with electronic noses for effective gas detection and mapping is described in [120].

Saldana et al. [124] approximated boundaries of a 2D surface oil slick using aquatic robots taking pointwise measurements. Saldana et. al approximated a boundary that changed over time, where we consider bound-

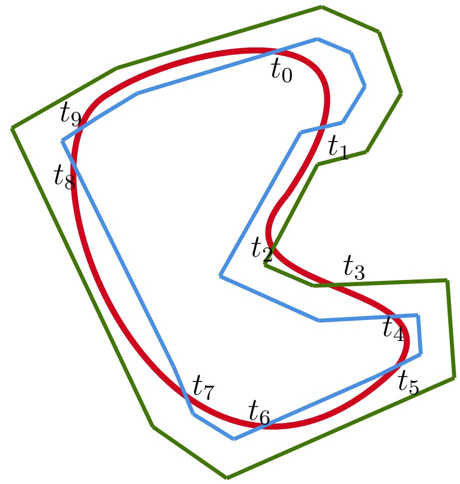


Figure 6.2: A high level execution of Algorithm 3, BOUNDARY-SKETCH, where the robots cross the boundary over timestamps t_0 through t_9. Here CROSS-BOUNDARY is executed whenever the sandwich invariant fails, in particular around timestamps t_0, t_2, t_4, t_6, t_8. The shape boundary is in red and the path of the robots are in blue and green.

aries that are stable over the short term. Their work was experimental and not grounded in a theoretical framework such as SKETCH.

Sung et al. [139] provide a survey of decision-theoretic approaches for robotic environmental monitoring. Robotic monitoring of volcanoes includesstudies of volcanic plume chemistry with ultralight sensor systems [90]. UAS-based remote sensing for the petroleum industry are discussed in [21].

A framework for multi-robot planning for persistent environmental monitoring is described in [100]. It appears that all such results: (1) assume instantaneous and continuous tracking of quantities such as boundary gradient or robot distance to boundary; (2) assume infinitesimally accurate control of the robots; and (3) do not give asymptotic bounds on robot travel time or energy expenditure.

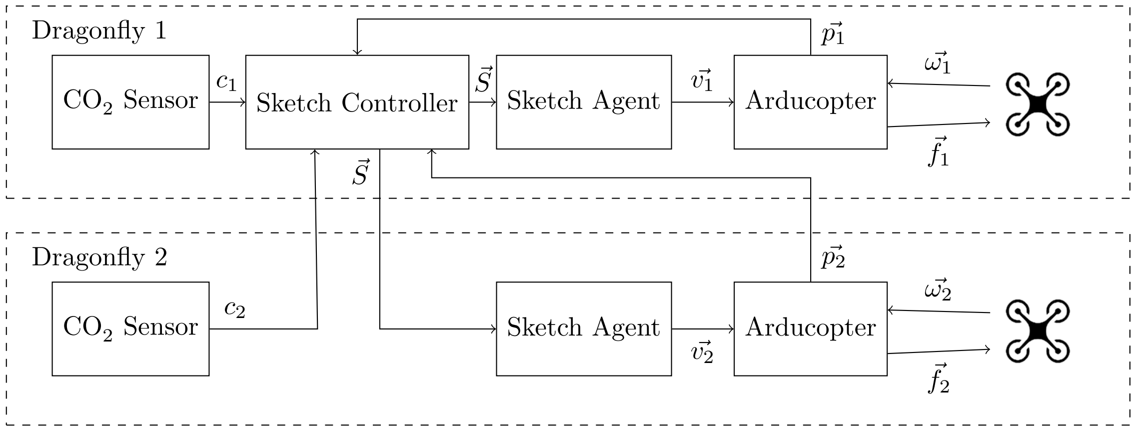


Figure 6.3: System level diagram of the Crazyflie Sketch implementation. The SKETCH algorithm is executed on the lead drone (Crazyflie 1) by the Sketch Controller. Flight paths on both drones are governed by their respective Sketch Agents under the direction of the Sketch Controller. Communication between the Sketch Controller and the Sketch Agents leverages the multi-agent-oriented ROS2 DDS infrastructure. Vector flight directives are issued from the Sketch Agent to the Crazyflie flight controller through the CFLib API.

Our algorithm, SKETCH, removes these assumptions and uses 2 robots to achieve asymptotically optimal distance traversed and angle turned by the robots. This has broad implications for minimizing a large class of efficiency measures. In particular, SKETCH is asymptotically optimal for any efficiency function that is polynomial in distance traveled and/or amount turned. Nguyen et al. [107], for example, discuss the importance of minimizing turns in UAS energy conservation.

The contributions of this paper are: i) crossing reality gap by demonstrating an implementation with low-cost robots in a lab environment, and ii) demonstrating that theoretical predictions, where the λ parameter guides the physical implementation, agree with results achieved in a real world implementation.

6.4 METHODS

We assume the boundary is a static curvilinear polygon: a closed shape with at most a finite number of discontinuities [114] which does not change over time. In practice, a boundary can be specified by picking a particular threshold for isoconcentrations, which defines the edge of a plume. Also, we acknowledge that a plume would likely change shape over time, so we leave more dynamic plume mapping for future work.

Every time a robot moves, it commits to traveling a path of distance at least λ, *or* turning at least λ radians; this is similar to the OBLOT model [70]. The unit λ is relative to the plume boundary diameter, which is normalized to be 1 unit. Additional topological assumptions governing the shape are described in terms of λ (see Section 1.1 in [47]).

6.4.1 Algorithm Overview

SKETCH seeks to maintain a *sandwich invariant*: the line segment connecting the two robots intersecting the boundary. If the invariant fails after the boundary is crossed by a robot, then both robots will be on the same side of the boundary. Then, the subroutine CROSS-BOUNDARY reestablishes the sandwich invariant, while ensuring that the distance traveled and rotation of both robots is optimal with what is required due to the curvature and perimeter of the plume boundary. Figure 6.2 illustrates the SKETCH algorithm.

The main idea in CROSS-BOUNDARY is to use the boundary gradient vector learned at the crossing point to guide the robot back to the other side of the boundary. The crossing robot successively takes small steps at a gradually increasing offset from the gradient at the last crossing. The angular offset is in the direction (clockwise or counterclockwise) of the shape boundary. Essentially the robot travels a regular polygon that approximates a small circle, until it crosses the boundary again. After the crossing, the robots synchronize to reorient so that both robots are found in parallel lines that sandwich the boundary. See Figure 6.2 that illustrates this overview.

By using CROSS-BOUNDARY to maintain the sandwich invariant, SKETCH efficiently computes an ϵ-sketch: a polygon with the property that every point on the boundary is within distance ϵ of the boundary.

6.4.2 Main Result

The properties of SKETCH are summarized in the following theorem from [46],

Theorem 1 *For any positive $\lambda < 1/2^6$, there exists an algorithm that uses two robots to compute an ϵ-SKETCH of the plume boundary, for $\epsilon = 8\sqrt{\lambda}$. Moreover the algorithm requires the robots to travel a total distance and rotate a total amount that are both asymptotically optimal.*

As a corollary we can use this ϵ-SKETCH to estimate the area of the plume.

Corollary 1 SKETCH *can estimate the area of the plume up to an additive error of* $O(\ell\sqrt{\lambda})$, *where* ℓ *is the perimeter of the shape.*

6.4.3 Implementation

The SKETCH implementation is divided into two parts, the Sketch Controller and the Sketch Agent. The Sketch Controller is executed on the leader drone, while the Sketch Agent is executed on both the leader and the follower drones. The Sketch Controller's role is to aggregate correlated position and CO_2 readings from the drone pair and make decisions on whether to fly in a straight line or turn towards the plume threshold. The Sketch Controller's output $\vec{S}$ is streamed to both Sketch Agents where the flight dynamics are built and sent on to the Crazyflies through CFLib. Figure 6.3 outlines the architecture of the SKETCH implmentation.

Algorithm 3 Initially, UASs D_1, D_2 are $\sqrt{\lambda}$ apart; one inside and one outside

procedure BOUNDARY-SKETCH(λ) ▷
$\nabla \leftarrow$ boundary gradient; SketchTerminate $\leftarrow$ **false**;
    while SketchTerminate $=$ **false do**
        if inside (D_1) XOR inside (D_2) **then**
            Move λ distance in the direction of ∇
        end if
        if not inside (D_1) and not inside (D_2) **then**
            CROSS-BOUNDARY($D_1, D_2, -\sqrt{\lambda}$)
        elseif inside (D_1) and inside (D_2)
            CROSS-BOUNDARY ($D_2, D_1, \sqrt{\lambda}$)
        end if
    end while
end procedure

Algorithm 3 determines if the pair of UASs should move forward by λ or turn by λ. This decision is based on the CO_2 readings at each of the UASs positions given by c and $\vec{p}$ at 10.00 Hz from each of the UAS. If the pair is sandwiching the plume threshold, the controller commands the pair to move forward, otherwise the controller commands the pair to turn towards the plume threshold. Both a forward and a turn command are given with a starting position $\vec{S}.position$ and vector direction $\vec{S}.direction$. This indicates how to start the given movement. A forward movement is given by the mode $\vec{S}.movement$= FORWARD, likewise a turn movement is given by the mode $\vec{S}.movement$= TURN.

Table 6.1: SKETCH Parameters

Name	Symbol
CO_2 threshold	c
Controller Command	$\vec{S}$
UAS Position	$\vec{p}$
UAS Velocity	$\vec{v}$
Motor Forces	$\vec{f}$
Flight Dynamics	$\vec{\omega}$

For forward movements, if the previous command was FORWARD then the direction is simply the previous direction vector as the sandwich invariant still holds. If the previous command was TURN (i.e. the sandwich invariant was just re-established) then the direction vector is calculated as the closest perpendicular vector (both 90.00 deg and −90.00 deg) to the calculated CO_2 normal to the previous direction vector. A turn command is given by the mode $\vec{S}.movement$= TURN, the starting position $\vec{S}.position$ and the direction $\vec{S}.direction$.

For turn movements, an additional term $\vec{S}.\alpha$ is given as the turning angle. The algorithm defines this either as positive or negative λ, depending whether the pair are to the left or to the right of the plume threshold. $\vec{S}.\alpha$ is used to calculate the turn center point $\vec{c}$, the pivot point for the turn.

The SKETCH dynamics are implemented using a flocking algorithm, where alignment, cohesion, and separation is maintained between two drones using a linear combination of velocity vectors. The linear vector sum is given as:

$$\vec{v_i} = c_s\vec{v_s} + c_t\vec{v_t} + \Delta t(c_{to}\vec{a_{to}} + c_{ta}\vec{a_{ta}} + c_e\vec{a_e}) \tag{6.1}$$

In this linear equation, $\vec{v_s}$ is the straight-line velocity vector, $\vec{v_t}$ is the turning velocity vector, $\vec{a_{to}}$ is the tandem-offset acceleration vector, $\vec{a_{ta}}$ is the tandem-alignment acceleration vector, and $\vec{a_e}$ is the error-correction acceleration vector. $\vec{\delta t}$ is the update interval in seconds for the $\vec{v_i}$ update. In our experiments $\vec{\delta t}$ is set to a constant 10.00 Hz. Straight-line or turning vectors are applied depending on the Sketch Controller's command vector $\vec{S}$. c constants are gain scalars used to tune the importance of the algorithm's dynamics.

$$\hat{v} = \frac{\vec{v}}{|\vec{v}|} \tag{6.2}$$

$$\text{MAX\_MAG}(\vec{v}, m) = \frac{\vec{v}}{\max(|\vec{v}|, m)} \tag{6.3}$$

$$\text{ROTATE}(\vec{v}, \theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \vec{v} \tag{6.4}$$

To ensure separation and cohesion between the two drones, we use the force vector $\vec{a_{to}}$ to maintain a tandem offset of $\sqrt{\lambda}$. This force pushes the drone away if the partner drone is closer than the specified offset, or pulls the drone towards the partner if the partner drone is farther away than the specified offset:

$$\delta\vec{p} = \vec{p}_{self} - \vec{p}_{partner} \tag{6.5}$$

$$p_{offset} = \delta\vec{p} - \vec{S}.offset \times \hat{\delta\vec{p}} \tag{6.6}$$

$$\vec{a_{to}} = \text{MAX\_MAG}(p_{offset}, \text{MAX\_FORCE}) \tag{6.7}$$

While flying in SKETCH, the drones are oriented perpendicular to both the straight-line velocity vector $\vec{v_s}$ or the turning velocity vector $\vec{v_t}$. This dynamic is maintained by adding either a forward or reverse force to the drone depending upon the partner's position:

$$a_{ta} = (\hat{\delta\vec{p}} \cdot \vec{S}.direction) \times \vec{S}.direction \tag{6.8}$$

Straight-line velocity is given by the Sketch Controller with a unit magnitude:

$$v_s = \vec{S}.\widehat{direction} \tag{6.9}$$

Turning velocity is produced by combining a force forward $\vec{c}_{opp}$ and a centripetal force $\vec{c}_{hyp}$ towards the center point c. Flying as a tandem pair, the outer and inner drones fly faster and slower respectively to maintain their orientation while flying the circular path.

$$\vec{c}_{opp} = \frac{\lambda}{2}\,\text{ROTATE}(\vec{S}.\hat{d}, \vec{S}.\alpha) \tag{6.10}$$

$$\vec{c}_{hyp} = \frac{\lambda}{2}\,\text{ROTATE}(\vec{S}.\hat{d}, \pi/2)\ \csc(\vec{S}.\alpha/2) \tag{6.11}$$

$$\vec{c} = \vec{p}_{self} + \vec{c}_{opp} + \vec{c}_{hyp} \tag{6.12}$$

$$v_{turn} = \begin{cases} 1 & \text{if } |\vec{p}_{self} - \vec{c}| > |\vec{p}_{partner} - \vec{c}| \\ \frac{|\vec{p}_{self} - \vec{c}|}{|\vec{p}_{partner} - \vec{c}|} & \text{else} \end{cases} \tag{6.13}$$

$$\vec{v}_t = v_{turn}\,\widehat{\vec{S}.direction}\,(\vec{c}_{opp} + \vec{c}_{hyp}) \tag{6.14}$$

Finally, we correct for deviations off of the straight-line path by using the following error correction vector. This mitigates real-world forces like wind acting upon the drones.

$$\vec{p}_c = \vec{p}_{self} - \text{AVE}(\vec{p}_{self}, \vec{p}_{partner}) \tag{6.15}$$

$$\vec{v}_{ec} = ((\vec{S}.direction \cdot \vec{p}_c) \odot \vec{S}.direction) - \vec{p}_c \tag{6.16}$$

$$a_e = \text{MAX\_MAG}(\vec{v}_{ec}, \text{MAX\_ERROR\_CORRECTION}) \tag{6.17}$$

The resulting paths flown by the pair of UASs are a constant distance $\sqrt{\lambda}$ apart. They maintain a perpendicular orientation to the $\vec{S}.direction$ vector, fly straight when sandwiching the plume threshold, and turn into the plume threshold when the sandwich invariant is broken. Following the theory behind SKETCH, decision points are made at λ intervals, meaning that there are discrete steps where the controller decides to fly straight or turn based on the main BOUNDARY-SKETCH algorithm.

6.4.4 Experimental Design

For the implementation, we utilize physical Crazyflie 2.1 UASs within a VICON Motion Capture System environment to track the positions of markers attached to them. We employed the Crazyflie library (CFLib) [25] for issuing commands to the Crazyflie UASs from a central station and CrazySwarm2 [1] to provide a ROS2 interface to manage Crazyflie UASs fleets with a motion capture library incorporated.

We implement a virtual plume that produces a CO_2 signal c to the Sketch Controller based on a given latitude and longitude position $\vec{p}$ using the PLUME equation given in 6.18. The virtual plume can be configured using wind speed u, emission rate Q, diffusion rate K, and angle θ [135].

$$\text{PLUME}(x, y) = \frac{Q}{2\pi K x} \exp\left(-\frac{u(y^2 + H^2)}{4Kx}\right) \tag{6.18}$$

We set up two experimental plumes. The first was a single convex curvilinear polygon. We simulate this with a single plume source with u=2.00 m/s, Q=12 000.00 g/s, $\theta = -30°$, and the boundary threshold is set at 680.00 ppm. The second experimental plume is a more complex, non-convex curvilinear polygon. In this case, two plume sources are virtually placed with a separation of x=1.61 m and y=0.11 m, and the boundary threshold is set at 550.00 ppm. The first plume in this scenario is configured with u=1.30 m/s, Q=3170.00 g/s, and $\theta = -30°$. The second plume is configured with u=1.20 m s^{-1}, Q=1200.00 g/s, and $\theta = -35°$.

Both scenarios begin with the pair of UASs starting at waypoints South of the plume separated by 0.20 m, with $\lambda = 0.005$ [1] The pair of UASs start searching for the plume by traveling forward due North until the plume threshold is encountered. As the pair of UASs execute SKETCH and walk the plume boundary, each position and virtual CO_2 reading is logged for post-flight analysis, enabling the calculation of the following key metrics: 1) Distance from the plume boundary (calculated two ways as described below), and 2) Percent error between the actual plume area and that estimated from the SKETCH boundary. The plume area is estimated by computing the area of a polygon formed using interpolated crossing locations with the plume.

Distance is calculated by comparing the location of the Crazyflie UASs during the flight to the nearest plume boundary point and then taking the maximum; this is the ϵ in the theoretical guarantee. We calculate the Fréchet distance [16] between the UAS flight paths and the ground truth plume. Fréchet distance captures the worst-case divergence between the UAS paths and the plume boundary. We compare this to the theoretical

[1] Since the robots start off being $\sqrt{\lambda}$ unit distance apart, $\sqrt{\lambda}$ scales linearly with the plume diameter and for the turn angle in CROSS-BOUNDARY, we scale this $\sqrt{\lambda}$ accordingly by dividing by the plume diameter.

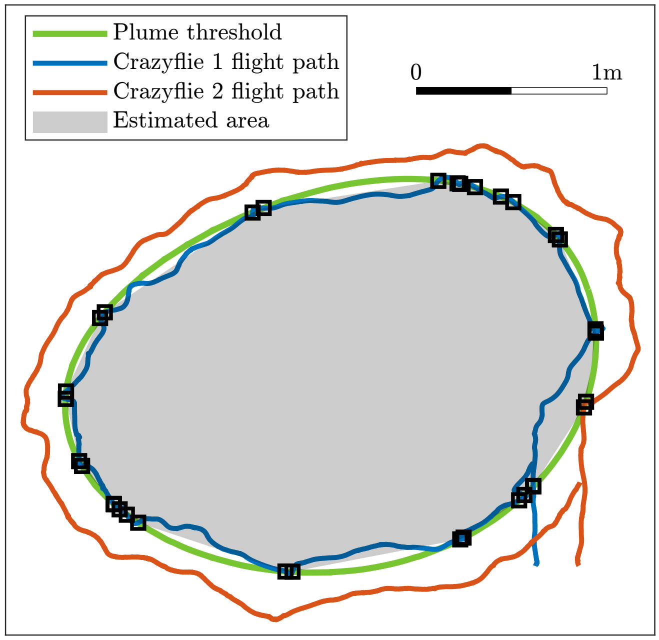
$8\sqrt{\lambda}$ guarantee.

We also calculate the average Hausdorff distance [54, 140] between the estimated and actual plume boundaries. Average Hausdorff distance provides an indication of how closely the drones followed the boundary on average. This practical estimate is a useful indication of how well SKETCH approximates the enclosed area.

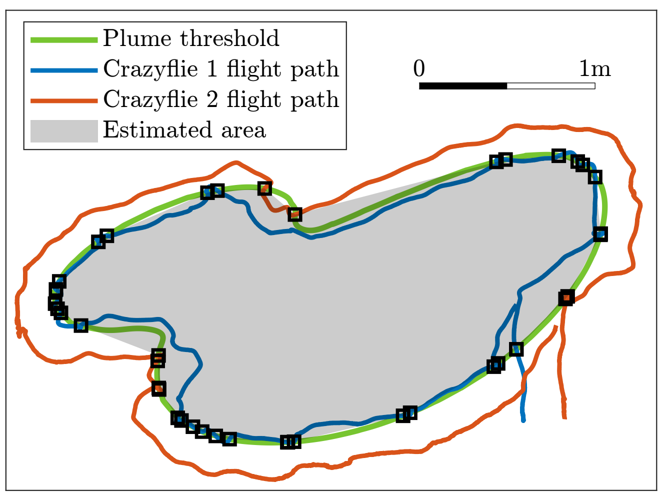
6.5 RESULTS

We mapped the single and double plume scenarios over 30 trials with Crazyflies in the laboratory. Figure 6.4 illustrates the ground plume threshold, flight paths of the Crazyflie UASs, and the estimated area for one trial of the SKETCH flight algorithm in different scenarios. Figure 6.5 shows the Crazyflie's distance from the plume boundary over time. The highest average distance is 0.30 m for both single and double plume cases, which is far below the theoretical maximum of $8\sqrt{\lambda}$ distance of 1.60 m. Figure 6.6 displays the Fréchet distance statistics. The maximum Fréchet distances are 0.34 m for the single plume and 0.40 m for the double plume, both of which are also below the theoretical maximum for SKETCH of 1.60 m. All distance comparisons here are in the actual units and not scaled against the unit diameter plume.

The algorithm effectively traces the plume area and boundary with

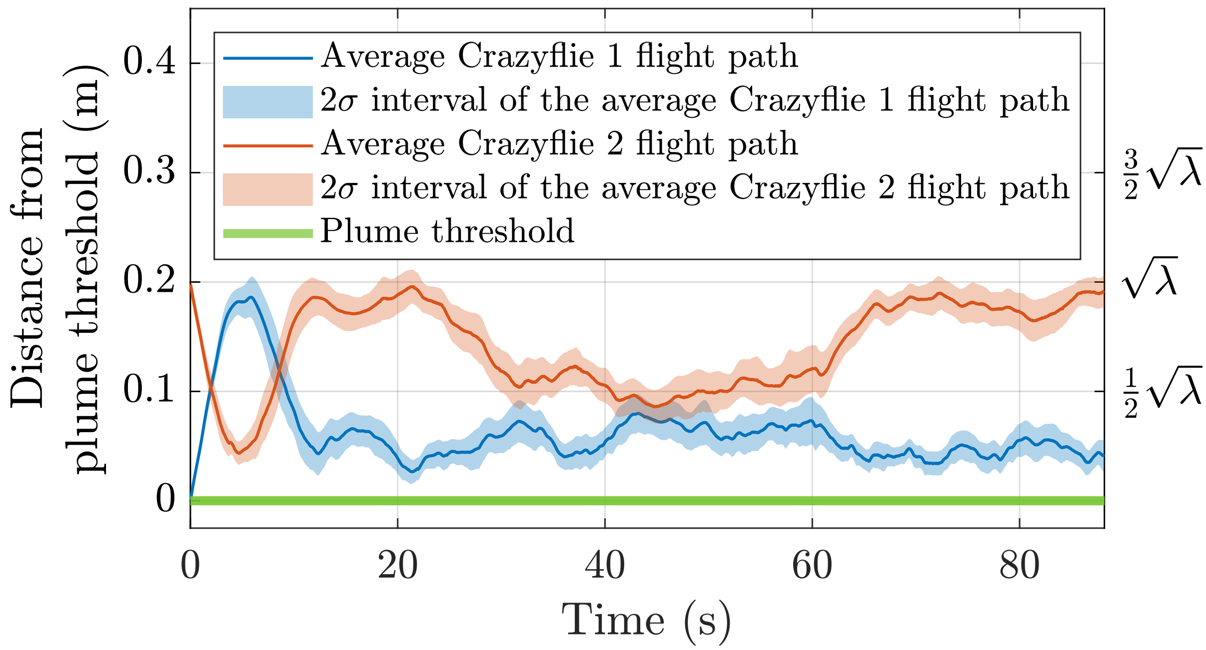


(a)

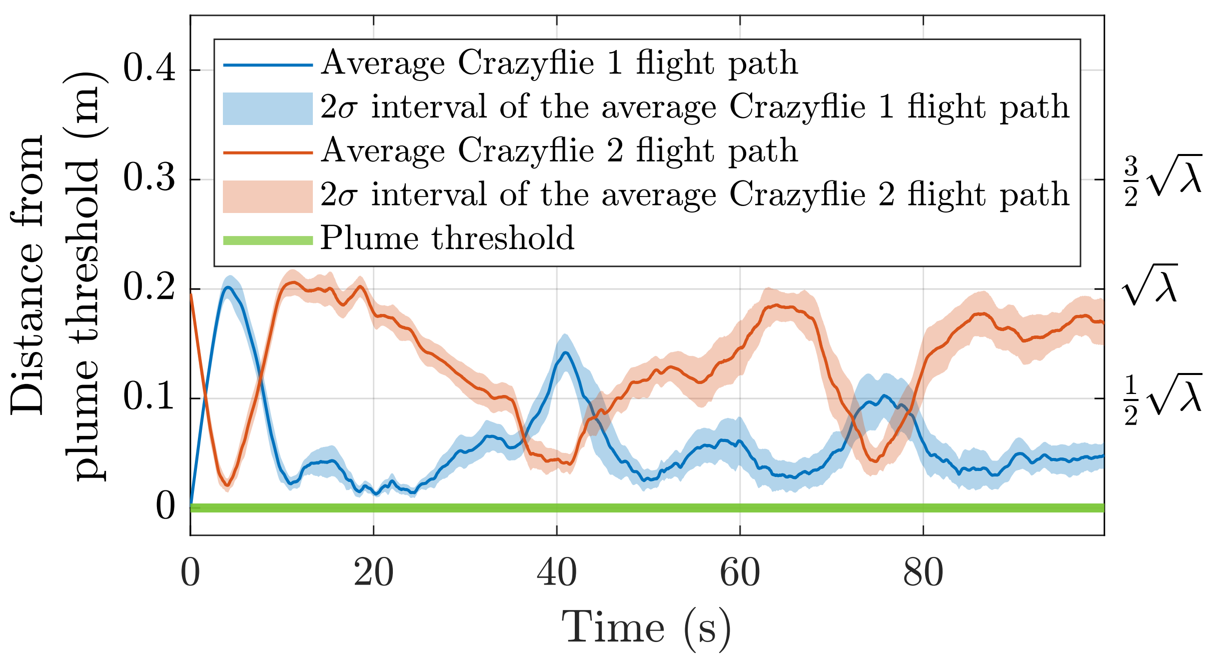


(b)

Figure 6.4: Example flights. (a) Single plume mapped by Crazyflies in the laboratory. (b) Double plume mapped by Crazyflies in the laboratory. The estimated area pictured in grey is formed by the polygon created by connecting the crossing points along the plume threshold.



(a)



(b)

Figure 6.5: Drone distance from the plume boundary over time. (a) Single plume mapping by Crazyflie UAS. (b) Double plume mapping by Crazyflie UAS.

minimal errors. Figure 6.6 presents the average Hausdorff distance statistics for different experiments. The average Hausdorf distance between the actual and estimated boundary is less than 0.11 m for the single plume and less than 0.07 m for the double plume in the Crazyflie implementation. Figure 6.7 displays the percentage error in area estimation. The algorithm computes the plume area with less than 10.73% of error in the single plume case and less than 7.63% in the double plume case.

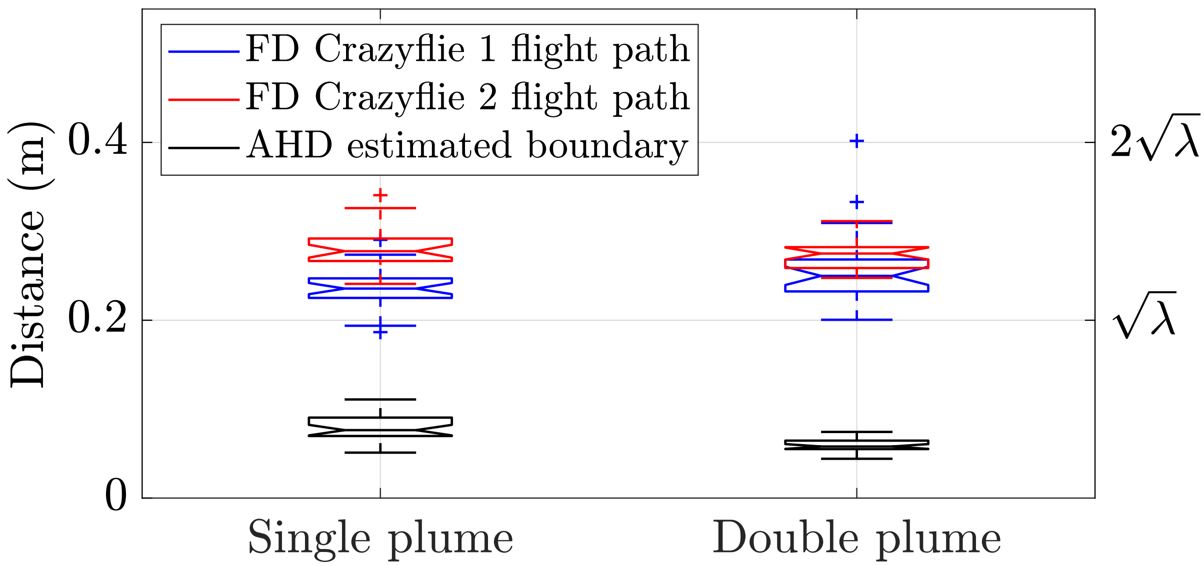


Figure 6.6: Distance between Crazyflie paths, the estimated plume boundary, and actual plume boundary. Fréchet distance (FD) captures the worst case divergence between the UAS paths and the plume boundary (blue and red). Average Hausdorff distance (AHD) indicates the average boundary following performance (black). 30 trials per experiment.

6.6 CONCLUSIONS

We empirically demonstrate in hardware (Figure 6.1) that the SKETCH algorithm survives the reality-gap and is a viable boundary mapping and area estimation algorithm. Our experiments show that the maximum

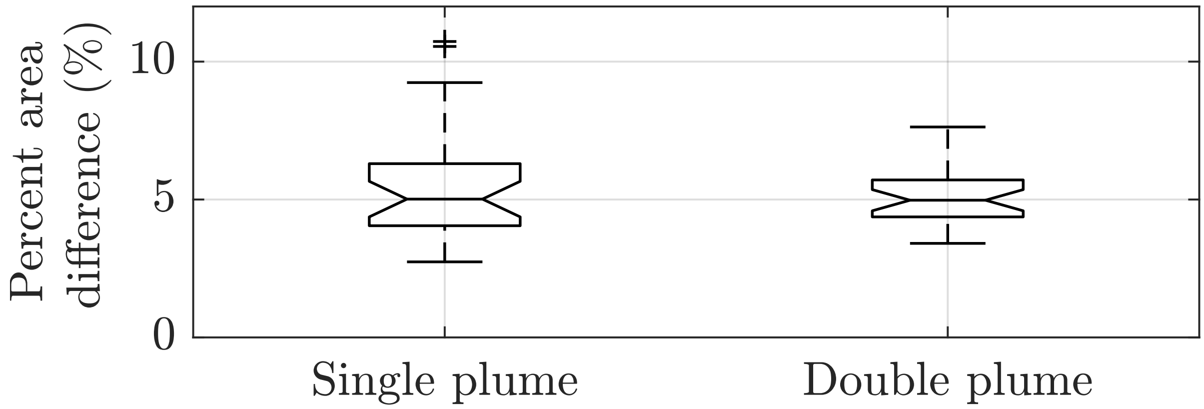


Figure 6.7: Accuracy of plume area estimation. Single plume error (mean=5.49 %, max=10.73 %) and double plume error (mean=5.16 %, max=7.63 %). 30 trials per experiment.

distance of each Crazyflie to the plume boundary is much less than the theoretical optimal bound for SKETCH of $8\sqrt{\lambda}$ (Figure 6.6). In practice this leads to plume area estimates in commodity hardware that in the worst case are within 10.73% of the actual area (Figure 6.7).

While our implementation focuses on volcanic plumes, the SKETCH algorithm's theoretical guarantees and practical performance make it applicable to a wide range of boundary mapping problems, including pollution tracking, agricultural monitoring, and environmental conservation. The algorithm's key strength lies in its ability to maintain accuracy guarantees while accounting for real-world robot dynamics through the λ parameter, eliminating the need for ad-hoc adjustments typically required when bridging theory and practice.

Our work demonstrates that optimality guarantees for distance traveled and rotation can be preserved in physical implementations, even with the constraints of low-cost robotic platforms like the Crazyflies. This has implications for energy-efficient operation in field deployments where battery life and changing conditions are critical constraints.

Looking ahead, we plan to integrate SKETCH with field-ready UASs equipped with actual gas sensors for deployment at active volcanic sites. Future work will focus on adapting the algorithm to handle dynamic boundary changes and varying plume conditions. Additionally, we intend to explore multi-robot configurations beyond pairs to further optimize coverage efficiency in larger-scale environments.

Encouraged by these results, in-field tests [59], and our previous volcano drone surveys [**?**, 68], we are confident that SKETCH provides a robust foundation for practical boundary mapping applications in challenging real-world environments.

ACKNOWLEDGMENT

This research was supported by the VolCAN project under National Science Foundation grant 2024520. Additional support to JE was provided by the Department of Energy's Kansas City National Security Campus, operated by Honeywell Federal Manufacturing & Technologies, LLC under contract number DE-NA0002839.

Chapter 7

Conclusion

This dissertation advances the field of distributed robotics by developing novel algorithms for coordinated sensing and mapping with autonomous UAS swarms. Moving from theoretical computer science foundations to practical field validation, this research demonstrates how distributed spatial algorithms can transform environmental monitoring, specifically addressing the urgent need for safer and more efficient methods in volcanology.

The main contributions advance both distributed robotics and spatial computing. The LoCUS algorithm provides a resilient method for multi-robot coordination that ensures reliable data collection even with drone failures—demonstrated here in volcanic plume surveying. Building on LoCUS, we successfully implemented flocking algorithms for coordinated gradient descent, allowing UASs to effectively track CO_2 plumes to their source, improving plume mapping efficiency and enabling precise location of high-concentration areas. We developed new methods for analyzing plume transect sensor data to calculate plume flux using direct CO_2 measurements that provide a more principled alternative to traditional monitoring methods that rely on CO_2/SO_2 ratios. Finally, we implemented and performed an empirical analysis of the Sketch algorithm, which provides an efficient method for boundary tracing using two UASs. This algorithm's ability to provide asymptotically optimal flight paths and turning behavior is crucial for accurate plume area estimation and rapid data collection.

The practical implementation was validated through simulations and field experiments, providing insights into the performance of the algorithms and hardware. The Dragonfly UAS platform, designed for long-duration flights and real-time CO_2 measurement in harsh environments, served as a testing framework. Its autonomous capabilities and network integration enabled effective field validation. Simulations of LoCUS demonstrated the algorithm's speed and reliability in locating the maximum CO_2 flux and responding to drone failure, when compared with a more dispersed approach. Field tests at the Valles Caldera demonstrated the effectiveness of the Dragonfly platform in rasterizing and mapping a known CO_2 source using lawnmower and DDSA algorithms, and successfully using a flocking algorithm to perform gradient descent to locate the known CO_2 source. Measurements during the Tajogaite volcanic eruption provided geo-referenced data on CO_2 concentrations, highlighting the potential of UASs for rapid assessment of eruptive gas emissions. Validation of the Sketch algorithm in both Gazebo and real-world field tests confirmed its ability to accurately trace plume boundaries, offering advantages over single-UAS approaches.

This research advances both distributed robotics and volcanology by providing new algorithms and approaches for coordinated sensing and mapping. These algorithms enable rapid and efficient plume mapping, accurate identification of CO_2 sources, precise estimation of plume boundaries and area, new methods to calculate CO_2 flux, and robust data collection in hazardous and unpredictable environments. By utilizing autonomous UASs, researchers can gather critical data more safely, efficiently, and frequently, enhancing the understanding of volcanic behavior and improving the prediction of volcanic hazards. The methods developed have broader applications in fields requiring distributed sensing, boundary tracing, and environmental monitoring.

7.1 Lessons learned

Deploying autonomous UAS swarms for volcanic CO_2 monitoring revealed critical challenges that bridge theory and practical field implementation. Our experiences highlighted key areas for improvement in drone platform design, sensing methodologies, communication infrastructure, and software frameworks.

The Dragonfly drone platform presented significant logistical challenges during field expeditions. Its substantial power system—six 6S LiPo batteries totaling approximately 3kg—proved cumbersome for transport and operation. Future deployments could benefit from a more compact and travel-friendly platform, potentially achieving similar flight endurance with just two 6S batteries. A lighter, more agile drone design would significantly reduce transportation burdens while mitigating safety risks associated with large drone operations.

Our CO_2 sensing approach utilized the PP-Systems SBA-5 sensor. While effective, we did not explore alternative, lower-cost sensors during this study. A future comparative analysis of budget-friendly CO_2 sensors could help assess accuracy trade-offs and reduce both sensor costs and overall drone weight. This would enhance scalability and accessibility for widespread environmental monitoring applications.

The software and networking architecture also presented important considerations. While ROS provided a robust framework for drone autonomy, its necessity, particularly for drone-to-drone communication, should be re-evaluated. A lightweight RPC and queuing system may offer more efficient and granular control over swarm coordination, simplifying data transmission and reducing network overhead.

Our communication hardware infrastructure highlighted opportunities for improvement. Implementing a true mesh network, such as XBee for drone-to-drone communication complemented by long-range connectivity via RFD 900x or similar technologies, could enhance swarm coordination and improve data transmission reliability while flying multiple UAVs.

The Raspberry Pi 4B proved to be an excellent companion computer, offering robust community support, strong computational performance, and multiple serial I/O options. The ability to program natively in Python allowed us to focus on algorithm development rather than low-level platform details. Additionally, its serial communication capabilities facilitated seamless integration with the Cube flight controller, network hardware, and SBA-5 CO_2 sensor, ensuring efficient data handling across subsystems.

Finally, vector-based navigation proved critical to our research, enabling flight capabilities beyond traditional waypoint-driven paths. This approach opened innovative possibilities for flocking and formation flying, fundamentally expanding our autonomous navigation strategies.

These lessons extend beyond volcanic monitoring, offering insights into autonomous systems development. They demonstrate that breakthrough technologies emerge from rigorous, iterative engagement with real-world complexities.

7.2 Future work

The success of this research opens several promising directions for future investigation. A key insight from the comparison of LoCUS and MoBS algorithms suggests a new approach I term "swarmlets" - multiple smaller subgroups of UASs, each operating as a cohesive unit but independent from other groups in the swarm. This hybrid approach could leverage MoBS's fast initial search capabilities while maintaining LoCUS's ability to perform gradient descent using simultaneous readings from members of a swarmlet.

Autonomous transect planning presents another important direction for development. By optimizing flight paths for plume mapping, we could extend beyond single transect analysis and eliminate assumptions about plume symmetry. This development would minimize flight time while maximizing the information gathered for CO_2 flux calculations for a more accurate and complete flux calculation. In addition, future algorithms could adapt to changing plume conditions and optimize the spacing and orientation of transects based on real-time data.

Integration of different approaches offer significant potential through automatic behavior switching based on real-time data analysis across the swarm. After completing an initial rasterization survey to locate the plume, the system could automatically transition to using Sketch for efficient boundary mapping, followed by LoCUS-based gradient descent to precisely locate the source. This sequential, data-driven integration would enable a comprehensive characterization of emissions while optimizing resource usage. Further exploration of multi-UAS cooperation and development of more sophisticated in-situ sensing capabilities will continue to advance our understanding of both distributed sensing systems and their environmental applications.

These algorithms have immediate applications in several critical domains. The LoCUS algorithm's robust distributed coordination makes it ideal for disaster response scenarios, where UAV swarms could rapidly assess damage and coordinate search efforts while maintaining reliable communication in challenging conditions. Urban air quality monitoring could benefit from gradient descent and boundary tracing techniques, allowing precise mapping of pollution sources and dispersion patterns in complex city environments. The Sketch algorithm's efficient boundary tracing capabilities could be adapted for monitoring forest fire boundaries or mapping oil spills.

While this work focuses on volcanic monitoring, these algorithms address fundamental challenges in distributed sensing and coordination that apply broadly. The industrial sector presents particularly promising applications for this technology. Our algorithms for plume detection and source localization are directly applicable to detecting methane leaks in oil and gas infrastructure — a critical environmental and safety concern. Building on the success of this research, we are spinning off a startup company that will adapt our core technologies—distributed swarm coordination, efficient spatial search, and plume analysis—to industrial methane detection. This commercialization effort represents not just the culmination of our academic work, but the beginning of a new chapter in applying distributed robotics to critical environmental challenges. The transition from research to commercial application demonstrates our work's immediate practical value and its potential for broader impact in addressing global environmental concerns.

Bibliography

[1] Crazyswarm2. https://github.com/IMRCLab/crazyswarm2. Accessed: 2024-09-10.

[2] V. Acocella, R. Di Lorenzo, C. Newhall, and R. Scandone. An overview of recent (1988 to 2014) caldera unrest: Knowledge and perspectives, 2015.

[3] Thomas Adamek, Christopher A. Kitts, and Ignacio Mas. Gradient-Based Cluster Space Navigation for Autonomous Surface Vessels. *IEEE/ASME Transactions on Mechatronics*, 20(2):506–518, 4 2015.

[4] Abhinav Aggarwal. *Thwarting Adversaries with Randomness and Irrationality*. PhD thesis, University of New Mexico, 2019.

[5] Abhinav Aggarwal, Diksha Gupta, William F Vining*, G Matthew Fricke, and Melanie E Moses. Ignorance is not bliss: An analysis of central-place foraging algorithms. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS) IEEE/RSJ. Accepted and available at https://bit. ly/2QcPaX8*, 2019.

[6] Abhinav Aggarwal, Diksha Gupta, William F. Vining, G. Matthew Fricke, and Melanie E. Moses. Ignorance is Not Bliss: An Analysis of Central-Place Foraging Algorithms. In *IEEE International Conference on Intelligent Robots and Systems*, 2019.

[7] Abhinav Aggarwal, William F. Vining, Diksha Gupta, Jared Saia, and Melanie E. Moses. A most irrational foraging algorithm, 2019.

[8] A. Aiuppa, C. Federico, G. Giudice, and S. Gurrieri. Chemical mapping of a fumarolic field: La Fossa Crater, Vulcano Island (Aeolian Islands, Italy). *Geophysical Research Letters*, 32(13):1–4, 7 2005.

[9] A Aiuppa, C Federico, G Giudice, and S Gurrieri. Chemical mapping of a fumarolic field: la fossa crater, vulcano island (aeolian islands, italy). *Geophysical Research Letters*, 32(13), 2005.

[10] Alessandro Aiuppa, Mike Burton, Tommaso Caltabiano, Gaetano Giudice, Sergio Guerrieri, Marco Liuzzo, Filippo Mur, and Giuseppe Salerno. Unusually large magmatic CO2 gas emissions prior to a basaltic paroxysm. *Geophysical Research Letters*, 37(17), 2010.

[11] Alessandro Aiuppa, Cinzia Federico, Gaetano Giudice, Giovanni Giuffrida, Roberto Guida, Sergio Gurrieri, Marco Liuzzo, Roberto Moretti, and Paolo Papale. The 2007 eruption of Stromboli volcano: Insights from real-time measurement of the volcanic gas plume

CO2/SO2 ratio. *Journal of Volcanology and Geothermal Research*, 182(3-4):221–230, 5 2009.

[12] Alessandro Aiuppa, Luca Fiorani, Simone Santoro, Stefano Parracino, Marcello Nuvoli, Giovanni Chiodini, Carmine Minopoli, and Giancarlo Tamburello. New ground-based lidar enables volcanic CO2 flux measurements. *Scientific Reports 2015 5:1*, 5(1):1–12, 9 2015.

[13] Alessandro Aiuppa, Tobias P. Fischer, Terry Plank, and Philipson Bani. CO2 flux emissions from the Earth's most actively degassing volcanoes, 2005–2015. *Scientific Reports 2019 9:1*, 9(1):1–17, 4 2019.

[14] Helena Albert, Fidel Costa, and Joan Martí. Years to weeks of seismic unrest and magmatic intrusions precede monogenetic eruptions. *Geology*, 44(3), 2016.

[15] Violeta T. Albertos, Guillermo Recio, Mar Alonso, Cecilia Amonte, Fátima Rodríguez, Claudia Rodríguez, Lia Pitti, Victoria Leal, Germán Cervigón, Judith González, Monika Przeor, José Manuel Santana-León, José Barrancos, Pedro A. Hernández, Germán D. Padilla, Gladys V. Melián, Eleazar Padrón, María Asensio-Ramos, and Nemesio M. Pérez. Sulphur dioxide (SO2) emissions by means of miniDOAS measurements during the 2021 eruption of Cumbre Vieja volcano, La Palma, Canary Islands. *EGU22*, 3 2022.

[16] Helmut Alt and Michael Godau. Computing the fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(01n02):75–91, 1995.

[17] Melanie J. Anderson, Joseph G. Sullivan, Timothy K. Horiuchi, Sawyer B. Fuller, and Thomas L. Daniel. A bio-hybrid odor-guided autonomous palm-sized air vehicle. *Bioinspiration and Biomimetics*, 16(2), 2020.

[18] R. J. Andres and W. I. Rose. Remote sensing spectroscopy of volcanic plumes and clouds. *Monitoring Active Volcanoes*, pages 301–314, 10 2022.

[19] Oisin Mac Aodha, Vassilios Stathopoulos, Gabriel J. Brostow, Michael Terry, Mark Girolami, and Kate E. Jones. Putting the Scientist in the Loop - Accelerating Scientific Progress with Interactive Machine Learning. In *Proceedings - International Conference on Pattern Recognition*, 2014.

[20] Svante Arrhenius. On the influence of carbonic acid in the air upon the temperature of the ground. *Philos. Mag*, 41:237, 1896.

[21] Saeid Asadzadeh, Wilson Jose de Oliveira, and Carlos Roberto de Souza Filho. Uav-based remote sensing for the petroleum industry and environmental monitoring: State-of-the-art and perspec-

tives. *Journal of Petroleum Science and Engineering*, 208:1–14, 2022.

[22] Mohamed Sami Assenine, Walid Bechkit, Ichrak Mokhtari, Hervé Rivano, and Karima Benatchba. Cooperative deep reinforcement learning for dynamic pollution plume monitoring using a drone fleet. *IEEE Internet of Things Journal*, 2023.

[23] Ralf Bachmayer and Naomi Ehrich Leonard. Vehicle networks for gradient descent in a sampled environment. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 1, pages 112–117. IEEE, 2002.

[24] Jim H Belanger and Mark A Willis. Biologically-inspired search algorithms for locating unseen odor sources. In *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell*, pages 265–270. IEEE, 1998.

[25] Bitcraze. cflib: Crazyflie python library. https://www.bitcraze.io/documentation/repository/crazyflie-lib-python/master/. Accessed: 2024-09-10.

[26] Bitcraze. Crazyflie 2.1. https://www.bitcraze.io/products/old-products/crazyflie-2-1/. Accessed: 2024-09-10.

[27] Emrah Biyik and Murat Arcak. Gradient climbing in formation via extremum seeking and passivity-based coordination rules. In *2007 46th IEEE Conference on Decision and Control*, pages 3133–3138. IEEE, 2007.

[28] Joseph R. Bourne, Matthew N. Goodell, Xiang He, Jake A. Steiner, and Kam K. Leang. Decentralized Multi-agent information-theoretic control for target estimation and localization: finding gas leaks. *International Journal of Robotics Research*, 39(13), 2020.

[29] Morten Breivik, Vegard E Hovstein, and Thor I Fossen. Ship formation control: A guided leader-follower approach. *IFAC Proceedings Volumes*, 41(2):16008–16014, 2008.

[30] Sarah K. Brown, Susanna F. Jenkins, R. Stephen J. Sparks, Henry Odbert, and Melanie R. Auker. Volcanic fatalities database: analysis of volcanic threat with distance and victim classification. *Journal of Applied Volcanology*, 6(1):1–20, 12 2017.

[31] Javier Burgués, Victor Hernández, Achim J. Lilienthal, and Santiago Marco. Smelling nano aerial vehicle for gas source localization and mapping. *Sensors (Switzerland)*, 19(3), 2019.

[32] Javier Burgués and Santiago Marco. Environmental chemical sensing using small drones: A review, 2020.

[33] Michael R. Burton, Georgina M. Sawyer, and Domenico Granieri. Deep Carbon Emissions from Volcanoes. *Reviews in Mineralogy and Geochemistry*, 75(1):323–354, 1 2013.

[34] Mike Burton, Alessandro Aiuppa, Patrick Allard, María Asensio-Ramos, Ana Pardo Cofrades, Alessandro La Spina, Emma J. Nicholson, Vittorio Zanon, José Barrancos, Marcello Bitetto, Margaret Hartley, Jorge E. Romero, Emma Waters, Alex Stewart, Pedro A. Hernández, João Pedro Lages, Eleazar Padrón, Kieran Wood, Benjamin Esse, Catherine Hayer, Klaudia Cyrzan, Estelle F. Rose-Koga, Federica Schiavi, Luca D'Auria, and Nemesio M. Pérez. Exceptional eruptive CO2 emissions from intra-plate alkaline magmatism in the Canary volcanic archipelago. *Communications Earth & Environment 2023 4:1*, 4(1):1–10, 12 2023.

[35] Gonçalo Cabrita and Lino Marques. Estimation of Gaussian plume model parameters using the simulated annealing algorithm. In *Advances in Intelligent Systems and Computing*, 2014.

[36] SA Carn, VE Fioletov, CA McLinden, C Li, and NA Krotkov. A decade of global volcanic so2 emissions measured from space, sci. rep., 7, 44095, 2017.

[37] J. C. Carracedo, E. R. Badiola, H. Guillou, J. De La Nuez, and F. J. Pérez Torrado. Geology and volcanology of la Palma and el Hierro,

western Canaries. *Estudios Geologicos*, 57(5-6), 2001.

[38] J. C. Carracedo, S. Day, H. Guillou, E. Rodríguez Badiola, J. A. Canas, and F. J. Pérez Torrado. Hotspot volcanism close to a passive continental margin: the Canary Islands. *Geological Magazine*, 135(5):591–604, 1998.

[39] Jonathan M. Castro and Yves Feisel. Eruption of ultralow-viscosity basanite magma at Cumbre Vieja, La Palma, Canary Islands. *Nature Communications 2022 13:1*, 13(1):1–12, 6 2022.

[40] Yicun Chen, Hao Cai, Zhilong Chen, and Qilin Feng. Using multi-robot active olfaction method to locate time-varying contaminant source in indoor environment. *Building and Environment*, 118:101–112, 2017.

[41] G. Chiodini, R. Cioni, M. Guidi, B. Raco, and L. Marini. Soil CO2 flux measurements in volcanic and geothermal areas. *Applied Geochemistry*, 13(5), 1998.

[42] Soon Jo Chung, Aditya Avinash Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. A Survey on Aerial Swarm Robotics. *IEEE Transactions on Robotics*, 34(4):837–855, 8 2018.

[43] Matthew Conlen. How Much Carbon Dioxide Are We Emitting? – Climate Change: Vital Signs of the Planet, 7 2021.

[44] David Crisp, Harold Pollock, Robert Rosenberg, Lars Chapsky, Richard Lee, Fabiano Oyafuso, Christian Frankenberg, Christopher Dell, Carol Bruegge, Gary Doran, Annmarie Eldering, Brendan Fisher, Dejian Fu, Michael Gunson, Lukas Mandrake, Gregory Osterman, Florian Schwandner, Kang Sun, Tommy Taylor, Paul Wennberg, and Debra Wunch. The on-orbit performance of the Orbiting Carbon Observatory-2 (OCO-2) instrument and its radiometrically calibrated products. *Atmospheric Measurement Techniques*, 10(1):59–81, 1 2017.

[45] Tung Dang, Marco Tranzatto, Shehryar Khattak, Frank Mascarich, Kostas Alexis, and Marco Hutter. Graph-based subterranean exploration path planning using aerial and legged robots. *Journal of Field Robotics*, 37(8), 2020.

[46] Varsha Dani, Abir Islam, and Jared Saia. *Boundary Sketching with Asymptotically Optimal Distance and Rotation*. Lecture Notes in Computer Science. Springer Nature Switzerland, 2023.

[47] Varsha Dani, Abir Islam, and Jared Saia. Boundary sketching with asymptotically optimal distance and rotation. *Theoretical Computer Science*, 1010:114714, 2024.

[48] S. J. Day, J. C. Carracedo, H. Guillou, and P. Gravestock. Recent structural evolution of the Cumbre Vieja volcano, La Palma, Canary

Islands: Volcanic rift zone reconfiguration as a precursor to volcano flank instability? *Journal of Volcanology and Geothermal Research*, 94(1-4), 1999.

[49] C De Luca, E Valerio, F Giudicepietro, G Macedonio, F Casu, and R Lanari. Pre-and co-eruptive analysis of the september 2021 eruption at cumbre vieja volcano (la palma, canary islands) through dinsar measurements and analytical modeling. *Geophysical Research Letters*, 49(7):e2021GL097293, 2022.

[50] J. M. de Moor, A. Aiuppa, J. Pacheco, G. Avard, C. Kern, M. Liuzzo, M. Martínez, G. Giudice, and T. P. Fischer. Short-period volcanic gas precursors to phreatic eruptions: Insights from Poás Volcano, Costa Rica. *Earth and Planetary Science Letters*, 442:218–227, 5 2016.

[51] Jorge Andres Diaz et al. Utilization of in situ airborne MS-based instrumentation for the study of gaseous emissions at active volcanoes. *International Journal of Mass Spectrometry*, 295(3):105–112, 2010.

[52] Jorge Andres Diaz, David Pieri, C. Richard Arkin, Eric Gore, Timothy P. Griffin, Matthew Fladeland, Geoff Bland, Carlomagno Soto, Yetty Madrigal, Daniel Castillo, Edgar Rojas, and Sergio Achí. Utilization of in situ airborne MS-based instrumentation for the study

of gaseous emissions at active volcanoes. *International Journal of Mass Spectrometry*, 295(3), 2010.

[53] Jorge Andres Diaz, David Pieri, Kenneth Wright, Paul Sorensen, Robert Kline-Shoder, C. Richard Arkin, Matthew Fladeland, Geoff Bland, Maria Fabrizia Buongiorno, Carlos Ramirez, Ernesto Corrales, Alfredo Alan, Oscar Alegria, David Diaz, and Justin Linick. Unmanned Aerial Mass Spectrometer Systems for In-Situ Volcanic Plume Analysis. *Journal of The American Society for Mass Spectrometry*, 26(2):292–304, 2015.

[54] Marie-Pierre Dubuisson and Anil K. Jain. A modified hausdorff distance for object matching. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 566–568. IEEE, 1994.

[55] John Ericksen, Abhinav Aggarwal, G. Matthew Fricke, and Melanie E. Moses. LoCUS: A Multi-Robot Loss-Tolerant Algorithm for Surveying Volcanic Plumes. *Proceedings - 4th IEEE International Conference on Robotic Computing, IRC 2020*, pages 113–120, 11 2020.

[56] John Ericksen, Kevin Aubert, Abir Islam, George Matthew Fricke, Varsha Dani, Rafael Fierro, Tobias P. Fischer, Scott Nowicki, Jared Saia, and Melanie Moses. From proof to practice: Asymptoti-

cally optimal boundary mapping by two robots. *Contributed Paper*, September 2024. Submitted on September 13, 2024.

[57] John Ericksen, Tobias P. Fischer, G. Matthew Fricke, Scott Nowicki, Nemesio M. Pérez, Pedro Hernández Pérez, Eleazar Padrón González, and Melanie E. Moses. Drone CO2 measurements during the Tajogaite volcanic eruption. *Atmospheric Measurement Techniques*, 17(15):4725–4736, 8 2024.

[58] John Ericksen, G. Matthew Fricke, Scott Nowicki, Tobias P. Fischer, Julie C. Hayes, Karissa Rosenberger, Samantha R. Wolf, Rafael Fierro, and Melanie E. Moses. Aerial Survey Robotics in Extreme Environments: Mapping Volcanic CO2 Emissions With Flocking UAVs. *Frontiers in Control Engineering*, 0:7, 3 2022.

[59] John Ericksen, Abir Islam, Carter Frost, Kevin Aubert, G. Matthew Fricke, Varsha Dani, Rafael Fierro, Tobias Fischer, Scott Nowicki, Jared Saia, and Melanie Moses. Navigating the Edge: UAS Boundary Tracing for Efficient Volcanic Plume Monitoring. *IEEE International Symposium on Safety, Security, and Rescue Robotics 2024, SSRR 2024*, pages 52–57, 2024.

[60] Juliane Euler, Andreas Horn, Dominik Haumann, Jürgen Adamy, and Oskar Von Stryk. Cooperative n-boundary tracking in large scale environments. In *2012 IEEE 9th International Conference on*

Mobile Ad-Hoc and Sensor Systems (MASS 2012), pages 1–6. IEEE, 2012.

[61] Daniele Facinelli, Matteo Larcher, Davide Brunelli, and Daniele Fontanelli. Cooperative uavs gas monitoring using distributed consensus. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 463–468. IEEE, 2019.

[62] José Fernández, Joaquín Escayo, Zhongbo Hu, Antonio G. Camacho, Sergey V. Samsonov, Juan F. Prieto, Kristy F. Tiampo, Mimmo Palano, Jordi J. Mallorquí, and Eumenio Ancochea. Detection of volcanic unrest onset in La Palma, Canary Islands, evolution and implications. *Scientific Reports*, 11(1), 2021.

[63] Tobias P. Fischer and Alessandro Aiuppa. AGU Centennial Grand Challenge: Volcanoes and Deep Carbon Global CO2 Emissions From Subaerial Volcanism—Recent Progress and Future Challenges. *Geochemistry, Geophysics, Geosystems*, 21(3), 2020.

[64] Tobias P. Fischer, Santiago Arellano, Simon Carn, Alessandro Aiuppa, Bo Galle, Patrick Allard, Taryn Lopez, Hiroshi Shinohara, Peter Kelly, Cynthia Werner, Carlo Cardellini, and Giovanni Chiodini. The emissions of CO2 and other volatiles from the world's subaerial volcanoes. *Scientific Reports*, 9(1), 2019.

[65] Tobias P Fischer, Santiago Arellano, Simon Carn, Alessandro Aiuppa, Bo Galle, Patrick Allard, Taryn Lopez, Hiroshi Shinohara, Peter Kelly, Cynthia Werner, et al. The emissions of co2 and other volatiles from the world's subaerial volcanoes. *Scientific reports*, 9(1):18716, 2019.

[66] Tobias P. Fischer and Taryn M. Lopez. First airborne samples of a volcanic plume for δ13C of CO2 determinations. *Geophysical Research Letters*, 43(7):3272–3279, 4 2016.

[67] Tobias P Fischer and Taryn M Lopez. First airborne samples of a volcanic plume for δ13c of co2 determinations. *Geophysical Research Letters*, 43(7):3272–3279, 2016.

[68] Tobias P Fischer, Céline L Mandon, Scott Nowicki, John Ericksen, Felipe Rojas Vilches, Melissa A Pfeffer, Alessandro Aiuppa, Marcello Bitetto, Angelo Vitale, G Matthew Fricke, et al. Co2 emissions during the 2023 litli hrútur eruption in reykjanes, iceland: δ13c tracks magma degassing. *Bulletin of Volcanology*, 86(6):1–10, 2024.

[69] Tobias P. Fischer, Céline L. Mandon, Scott Nowicki, John Ericksen, Felipe Rojas Vilches, Melissa A. Pfeffer, Alessandro Aiuppa, Marcello Bitetto, Angelo Vitale, G. Matthew Fricke, Melanie E. Moses, and Andri Stefánsson. CO2 emissions during the 2023 Litli Hrútur

eruption in Reykjanes, Iceland: d13C tracks magma degassing. *Bulletin of Volcanology*, 86(6), 6 2024.

[70] Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Distributed computing by mobile entities. *Current Research in Moving and Computing*, 11340(1), 2019.

[71] G. Matthew Fricke, Joshua P. Hecker, Antonio D. Griego, Linh T. Tran, and Melanie E. Moses. A distributed deterministic spiral search algorithm for swarms. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2016-November, 2016.

[72] Bo Galle, Santiago Arellano, Nicole Bobrowski, Vladimir Conde, Tobias P Fischer, Gustav Gerdes, Alexandra Gutmann, Thorsten Hoffmann, Ima Itikarai, Tomas Krejci, et al. A multi-purpose, multi-rotor drone system for long-range and high-altitude volcanic gas plume measurements. *Atmospheric Measurement Techniques*, 14(6):4255–4277, 2021.

[73] Bo Galle, Santiago Arellano, Nicole Bobrowski, Vladimir Conde, Tobias P. Fischer, Gustav Gerdes, Alexandra Gutmann, Thorsten Hoffmann, Ima Itikarai, Tomas Krejci, Emma J. Liu, Kila Mulina, Scott Nowicki, Tom Richardson, Julian Rüdiger, Kieran Wood, and Jiazhi Xu. A multi-purpose, multi-rotor drone system for long-range

and high-altitude volcanic gas plume measurements. *Atmospheric Measurement Techniques*, 14(6), 2021.

[74] Bo Galle, Clive Oppenheimer, Andreas Geyer, Andrew J.S. McGonigle, Marie Edmonds, and Lisa Horrocks. A miniaturised ultraviolet spectrometer for remote sensing of SO2 fluxes: a new tool for volcano surveillance. *Journal of Volcanology and Geothermal Research*, 119(1-4):241–254, 1 2003.

[75] T. M. Gerlach, H. Delgado, K. A. McGee, M. P. Doukas, J. J. Venegas, and L. Cárdenas. Application of the LI-COR CO2 analyzer to volcanic plumes: A case study, volcán Popocatépetl, Mexico, June 7 and 10, 1995. *Journal of Geophysical Research: Solid Earth*, 102(B4):8005–8019, 4 1997.

[76] T M Gerlach, K A Mcgee, T Elias, A J Sutton, M P Doukas, T A Mcgee, A J Elias, and M P Sutton. Carbon dioxide emission rate of Kīlauea Volcano: Implications for primary magma and the summit reservoir. *Journal of Geophysical Research: Solid Earth*, 107(B9):3–1, 9 2002.

[77] Khaled A Ghamry, Mohamed A Kamel, and Youmin Zhang. Cooperative forest monitoring and fire detection using a team of uavs-ugvs. In *2016 International conference on unmanned aircraft systems (ICUAS)*, pages 1206–1211. IEEE, 2016.

[78] W. F. Giggenbach. Chemical Composition of Volcanic Gases. *Monitoring and Mitigation of Volcano Hazards*, pages 221–256, 1996.

[79] Fraser Goff and Cathy J. Janik. Gas geochemistry of the Valles caldera region, New Mexico and comparisons with gases at Yellowstone, Long Valley and other geothermal systems. *Journal of Volcanology and Geothermal Research*, 116(3-4), 2002.

[80] Jon Golla. Natural Salinization of the Jemez River, New Mexico: An Insight From Trace Element Geochemistry. *Earth and Planetary Sciences ETDs*, 7 2019.

[81] Christopher Gomez and Heather Purdie. UAV- based Photogrammetry and Geocomputing for Hazards and Disaster Risk Monitoring – A Review. *Geoenvironmental Disasters*, 3(1):1–11, 12 2016.

[82] Christoffer R. Heckman, Ira B. Schwartz, and M. Ani Hsieh. Toward efficient navigation in uncertain gyre-like flows. *The International Journal of Robotics Research*, 34(13):1590–1603, 2015.

[83] T. Ilanko, T. P. Fischer, P. Kyle, A. Curtis, H. Lee, and Y. Sano. Modification of fumarolic gases by the ice-covered edifice of Erebus volcano, Antarctica. *Journal of Volcanology and Geothermal Research*, 381:119–139, 9 2019.

[84] Instituto Geográfico Nacional. Noticias e informe mensual de vigilancia volcánica, 2022.

[85] B Jacob and G Guennebaud. Eigen is a c++ template library for linear algebra: matrices, vectors, numerical solvers, and related algorithms, 2012.

[86] Alexander Jahn, Reza Javanmard Alitappeh, David Saldaña, Luciano C. A. Pimenta, Andre G. Santos, and Mario F. M. Campos. Distributed multi-robot coordination for dynamic perimeter surveillance in uncertain environments. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 273–278, 2017.

[87] Mike R. James, Brett B. Carr, Fiona D'Arcy, Angela K. Diefenbach, Hannah R. Dietterich, Alessandro Fornaciai, Einat Lev, Emma J. Liu, David C. Pieri, Mel Rodgers, Benoît Smets, Akihiko Terada, Felix W. von Aulock, Thomas R. Walter, Kieran T. Wood, and Edgar U. Zorn. Volcanological applications of unoccupied aircraft systems (UAS): Developments, strategies, and future challenges, 2020.

[88] Shumaila Javaid, Nasir Saeed, Zakria Qadir, Hamza Fahim, Bin He, Houbing Song, and Muhammad Bilal. Communication and control in collaborative uavs: Recent advances and future trends. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[89] Matthew S. Johnson, Florian M. Schwandner, Christopher S. Potter, Hai M. Nguyen, Emily Bell, Robert R. Nelson, Sajeev Philip, and Christopher W. O'Dell. Carbon Dioxide Emissions During the 2018 Kilauea Volcano Eruption Estimated Using OCO-2 Satellite Retrievals. *Geophysical Research Letters*, 47(24):e2020GL090507, 12 2020.

[90] Niklas Karbach, Nicole Bobrowski, and Thorsten Hoffmann. Observing volcanoes with drones: studies of volcanic plume chemistry with ultralight sensor systems. *Scientific reports*, 12(1):17890, 2022.

[91] Charles D. Keeling. The concentration and isotopic abundances of atmospheric carbon dioxide in rural areas. *Geochimica et Cosmochimica Acta*, 13(4), 1958.

[92] Nathan Koenig and Andrew Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, 2004.

[93] Hyunwoo Lee, James D. Muirhead, Tobias P. Fischer, Cynthia J. Ebinger, Simon A. Kattenhorn, Zachary D. Sharp, and Gladys Kianji. Massive and prolonged deep carbon emissions associated with continental rifting. *Nature Geoscience 2016 9:2*, 9(2):145–149, 1 2016.

[94] José Manuel Santana de León, Gladys V. Melián, Claudia Rodríguez, Germán Cervigón-Tomico, Victor Ortega, David Martínez van Dorth, Iván Cabrera-Pérez, María Cordero, Monika Przeor, Rui Filipe Fagundes Silva, Sandro Branquinho de Matos, Eleonora Baldoni, Maria Margarida Pires Ramalho, Fátima Viveiros, David Calvo, and Nemesio M. Pérez. Long-term variations of diffuse CO2 at Cumbre Vieja volcano, La Palma, Canary Islands. *EGU22*, 3 2022.

[95] Xiaojuan Lin, Ronald Van Der A, Jos De Laat, Henk Eskes, Frédéric Chevallier, Philippe Ciais, Zhu Deng, Yuanhao Geng, Xuanren Song, Xiliang Ni, Da Huo, Xinyu Dou, and Zhu Liu. Monitoring and quantifying CO2emissions of isolated power plants from space. *Atmospheric Chemistry and Physics*, 23(11):6599–6611, 6 2023.

[96] E. J. Liu, A. Aiuppa, A. Alan, S. Arellano, M. Bitetto, N. Bobrowski, S. Carn, R. Clarke, E. Corrales, J. M. De Moor, J. A. Diaz, M. Edmonds, T. P. Fischer, J. Freer, G. M. Fricke, B. Galle, G. Gerdes, G. Giudice, A. Gutmann, C. Hayer, I. Itikarai, J. Jones, E. Mason, B. T. McCormick Kilbride, K. Mulina, S. Nowicki, K. Rahilly, T. Richardson, J. Rüdiger, C. I. Schipper, I. M. Watson, and K. Wood. Aerial strategies advance volcanic gas measurements at inaccessible, strongly degassing volcanoes. *Science Advances*, 6(44),

2020.

[97] E J Liu et al. Multi-species volatile fluxes from manam, papua new guinea. in prep.

[98] Emma J Liu, Alessandro Aiuppa, Alfredo Alan, S Arellano, Marcello Bitetto, Nicole Bobrowski, Simon Carn, Robert Clarke, Ernesto Corrales, J Maarten de Moor, et al. Aerial strategies advance volcanic gas measurements at inaccessible, strongly degassing volcanoes. *Science Advances*, 6(44):eabb9103, 2020.

[99] Emma J. Liu et al. Dynamics of Outgassing and Plume Transport Revealed by Proximal Unmanned Aerial System (UAS) Measurements at Volcán Villarrica, Chile. *Geochemistry, Geophysics, Geosystems*, 2019.

[100] Kai-Chieh Ma, Zhibei Ma, Lantao Liu, and Gaurav S Sukhatme. Multi-robot informative and adaptive planning for persistent environmental monitoring. In *Distributed Autonomous Robotic Systems (DARS): The 13th International Symposium*, pages 285–298. Springer, 2018.

[101] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science robotics*, 7(66):eabm6074, 2022.

[102] Michael Manga, Simon A. Carn, Katharine V. Cashman, Amanda B. Clarke, Charles B. Connor, Kari M. Cooper, Tobias Fischer, Bruce Houghton, Jeffrey B. Johnson, Terry A. Plank, Diana C. Roman, Paul Segall, Stephen McNutt, Gene Whitney, R. Lyndon Arscott, Christopher Cameron, Rodney C. Ewing, Carol P. Harden, T. Mark Harrison, Thorne Lay, Ann S. Maest, Zelma Maine-Jackson, Martin W. Mccann, James M. Robertson, James Slutz, Shaowen Wang, Elizabeth A. Eide, Anne M. Linn, Deborah Glickson, Sammantha L. Magsino, Nicholas D. Rogers, Courtney R. Gibbs, Eric J. Edkin, and Raymond M. Chappetta. Volcanic Eruptions and Their Repose, Unrest, Precursors, and Timing. *Volcanic Eruptions and Their Repose, Unrest, Precursors, and Timing*, pages 1–122, 4 2017.

[103] Ray Nassar, Timothy G. Hill, Chris A. McLinden, Debra Wunch, Dylan B.A. Jones, and David Crisp. Quantifying CO2 Emissions From Individual Power Plants From Space. *Geophysical Research Letters*, 44(19):045–10, 10 2017.

[104] Ray Nassar, Jon Paul Mastrogiacomo, William Bateman-Hemphill, Callum McCracken, Cameron G. MacDonald, Tim Hill, Christopher W. O'Dell, Matthäus Kiel, and David Crisp. Advances in quantifying power plant CO2 emissions with OCO-2. *Remote Sensing of Environment*, 264:112579, 10 2021.

[105] C. A. Neal, S. R. Brantley, L. Antolik, J. L. Babb, M. Burgess, K. Calles, M. Cappos, J. C. Chang, S. Conway, L. Desmither, P. Dotray, T. Elias, P. Fukunaga, S. Fuke, I. A. Johanson, K. Kamibayashi, J. Kauahikaua, R. L. Lee, S. Pekalib, A. Miklius, W. Million, C. J. Moniz, P. A. Nadeau, P. Okubo, C. Parcheta, M. R. Patrick, B. Shiro, D. A. Swanson, W. Tollett, F. Trusdell, E. F. Younger, M. H. Zoeller, E. K. Montgomery-Brown, K. R. Anderson, M. P. Poland, J. L. Ball, J. Bard, M. Coombs, H. R. Dietterich, C. Kern, W. A. Thelen, P. F. Cervelli, T. Orr, B. F. Houghton, C. Gansecki, R. Hazlett, P. Lundgren, A. K. Diefenbach, A. H. Lerner, G. Waite, P. Kelly, L. Clor, C. Werner, K. Mulliken, G. Fisher, and D. Damby. Volcanology: The 2018 rift eruption and summit collapse of Kilauea Volcano. *Science*, 363(6425), 2019.

[106] Patrick P. Neumann, Victor Hernandez Bennetts, Achim J. Lilienthal, and Matthias Bartholmai. From insects to micro air vehicles—a comparison of reactive plume tracking strategies. In *Advances in Intelligent Systems and Computing*, 2015.

[107] An Nguyen, Dominik Krupke, Mary Burbage, Shriya Bhatnagar, Sándor P Fekete, and Aaron T Becker. U sing a uav for destructive surveys of mosquito population. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7812–7819.

IEEE, 2018.

[108] Office of Science, US DOE. Fluxnet, 2023.

[109] Eleazar Padrón, Nemesio M. Pérez, Fátima Rodríguez, Gladys Melián, Pedro A. Hernández, Hirochika Sumino, Germán Padilla, José Barrancos, Samara Dionis, Kenji Notsu, and David Calvo. Dynamics of diffuse carbon dioxide emissions from Cumbre Vieja volcano, La Palma, Canary Islands. *Bulletin of Volcanology*, 77(4), 2015.

[110] Claudio Paliotta, Dennis JW Belleter, and Kristin Y Pettersen. Adaptive source seeking with leader-follower formation control. *IFAC-PapersOnLine*, 48(16):285–290, 2015.

[111] Nemesio M. Pérez, Pedro A. Hernández, Gladys V. Melián, Eleazar Padrón, María Asensio-Ramos, José Barrancos, Germán D. Padilla, Fátima Rodríguez, Luca D'Auria, Cecilia Amonte, Mar Alonso, Alba Martín-Lorenzo, David Calvo, Claudia Rodríguez, William Hernández, Beverley Coldwell, and Matthew J. Pankhurst. The 2021 Cumbre Vieja eruption: an overview of the geochemical monitoring program. *EGU22*, 3 2022.

[112] Carlo Pinciroli et al. Argos: a modular, multi-engine simulator for heterogeneous swarm robotics. In *2011 IEEE/RSJ International*

Conference on Intelligent Robots and Systems, pages 5027–5034. IEEE, 2011.

[113] Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, and Frederick Ducatelle. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm intelligence*, 6(4):271–295, 2012.

[114] Andrew Pressley. *Elementary Differential Geometry*. Springer, 2012.

[115] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, volume 3, 2009.

[116] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, Thomas Lagkas, and Ioannis Moscholios. A compilation of UAV applications for precision agriculture. *Computer Networks*, 172:107148, 5 2020.

[117] Kristen E. Rahilly. *Diffuse Flux and Carbon Isotope Composition of Carbon Dioxide Emitted from Valles Caldera, Yellowstone Caldera, and Southwestern Utah Geothermal Sites*. PhD thesis, University of New Mexico, 2020.

[118] Craig W Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87*, 21(4), 1987.

[119] Claudia Rodríguez-Pérez, José Barrancos, Pedro A Hernández, Nemesio M Pérez, Eleazar Padrón, Gladys V Melián, Fátima Rodríguez, María Asensio-Ramos, and Germán D Padilla. Continuous monitoring of diffuse co2 emission from cumbre vieja volcano: early evidences of magmatic co2 surface arrival. Technical report, Copernicus Meetings, 2022.

[120] Maurizio Rossi and Davide Brunelli. Autonomous gas detection and mapping with unmanned aerial vehicles. *IEEE Transactions on Instrumentation and measurement*, 65(4):765–775, 2015.

[121] Maurizio Rossi and Davide Brunelli. Gas sensing on unmanned vehicles: Challenges and opportunities. In *Proceedings - 2017 1st New Generation of CAS, NGCAS 2017*, pages 117–120. Institute of Electrical and Electronics Engineers Inc., 9 2017.

[122] Maurizio Rossi, Davide Brunelli, Andrea Adami, Leandro Lorenzelli, Fabio Menna, and Fabio Remondino. Gas-drone: Portable gas sensing system on UAVs for gas leakage localization. *Proceedings of IEEE Sensors*, 2014-December(December):1431–1434, 12 2014.

[123] Julian Rüdiger, Jan Lukas Tirpitz, J. Maarten De Moor, Nicole Bobrowski, Alexandra Gutmann, Marco Liuzzo, Martha Ibarra, and Thorsten Hoffmann. Implementation of electrochemical, optical and denuder-based sensors and sampling techniques on UAV for volcanic gas measurements: Examples from Masaya, Turrialba and Stromboli volcanoes. *Atmospheric Measurement Techniques*, 11(4), 2018.

[124] David Saldaña, Renato Assunção, M Ani Hsieh, Mario FM Campos, and Vijay Kumar. Estimating boundary dynamics using robotic sensor networks with pointwise measurements. *Autonomous Robots*, 45:193–208, 2021.

[125] A. Sandoval-Velasquez, F. Casetta, T. Ntaflos, A. Aiuppa, M. Coltorti, M. L. Frezzotti, M. Alonso, E. Padrón, M. Pankhurst, N. M. Pérez, and A. L. Rizzo. 2021 Tajogaite eruption records infiltration of crustal fluids within the upper mantle beneath La Palma, Canary Islands. *Frontiers in Earth Science*, 12:1303872, 2 2024.

[126] Andres Sandoval-Velasquez, Andrea Luca Rizzo, Alessandro Aiuppa, Samantha Remigi, Eleazar Padrón, Nemesio M. Pérez, and Maria Luce Frezzotti. Recycled crustal carbon in the depleted mantle source of El Hierro volcano, Canary Islands. *Lithos*, 400-401, 2021.

[127] Julien Schleich, Athithyaa Panchapakesan, Grégoire Danoy, and Pascal Bouvry. UAV fleet area coverage with network connectivity constraint. In *MobiWac 2013 - Proceedings of the 11th ACM International Symposium on Mobility Management and Wireless Access, Co-located with ACM MSWiM 2013*, 2013.

[128] Rodney Albert Schmidt Jr. *A study of the real-time control of a computer-driven vehicle*. Stanford University, 1971.

[129] H. U. Schmincke. Volcanic and chemical evolution of the Canary Islands. *Geology of the northwest African continental margin*, pages 273–306, 1982.

[130] Florian M. Schwandner, Michael R. Gunson, Charles E. Miller, Simon A. Carn, Annmarie Eldering, Thomas Krings, Kristal R. Verhulst, David S. Schimel, Hai M. Nguyen, David Crisp, Christopher W. O'Dell, Gregory B. Osterman, Laura T. Iraci, and James R. Podolske. Spaceborne detection of localized carbon dioxide sources. *Science*, 358(6360), 10 2017.

[131] Kunal Shah, Grant Ballard, Annie Schmidt, and Mac Schwager. Multidrone aerial surveys of penguin colonies in Antarctica. *Science Robotics*, 5(47), 2020.

[132] Samia Souissi, Taisuke Izumi, and Koichi Wada. Oracle-based flocking of mobile robots in crash-recovery model. *Theoretical Computer Science*, 412(33):4350–4360, 2011.

[133] John Stix and J. Maarten de Moor. Understanding and forecasting phreatic eruptions driven by magmatic degassing. *Earth, Planets and Space*, 70(1), 2018.

[134] John Stix, J. Maarten de Moor, Julian Rüdiger, Alfredo Alan, Ernesto Corrales, Fiona D'Arcy, Jorge Andres Diaz, and Marcello Liotta. Using Drones and Miniaturized Instrumentation to Study Degassing at Turrialba and Masaya Volcanoes, Central America. *Journal of Geophysical Research: Solid Earth*, 123(8):6501–6520, 8 2018.

[135] John M Stockie. The mathematics of atmospheric dispersion modeling. *Siam Review*, 53(2):349–372, 2011.

[136] John M. Stockie. The mathematics of atmospheric dispersion modeling. *SIAM Review*, 53(2), 2011.

[137] Wolfgang Stremme, Michel Grutter, Jorge Baylón, Noemie Taquet, Alejandro Bezanilla, Eddy Plaza-Medina, Benedetto Schiavo, Claudia Rivera, Thomas Blumenstock, and Frank Hase. Direct solar ftir measurements of co2 and hcl in the plume of popocatépetl volcano, mexico. *Frontiers in Earth Science*, 11:1022976, 2023.

[138] Xiang Sun, Nirwan Ansari, and Rafael Fierro. Jointly Optimized 3D Drone Mounted Base Station Deployment and User Association in Drone Assisted Mobile Access Networks. *IEEE Transactions on Vehicular Technology*, 69(2), 2020.

[139] Yoonchang Sung, Zhiang Chen, Jnaneshwar Das, Pratap Tokekar, et al. A survey of decision-theoretic approaches for robotic environmental monitoring. *Foundations and Trends® in Robotics*, 11(4):225–315, 2023.

[140] Abdel Aziz Taha and Allan Hanbury. An efficient algorithm for calculating the exact hausdorff distance. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2153–2163, 2015.

[141] Dinesh Thakur, Giuseppe Loianno, Wenxin Liu, and Vijay Kumar. Nuclear Environments Inspection with Micro Aerial Vehicles: Algorithms and Experiments. In *Springer Proceedings in Advanced Robotics*, volume 11. 2020.

[142] G. Vásárhelyi, Cs Virágh, G. Somorjai, N. Tarcai, T. Szörényi, T. Nepusz, and T. Vicsek. Outdoor flocking and formation flight with autonomous aerial robots. In *IEEE International Conference on Intelligent Robots and Systems*, 2014.

[143] Csaba Virágh, Gábor Vásárhelyi, Norbert Tarcai, Tamás Szörényi, Gergő Somorjai, Tamás Nepusz, and Tamás Vicsek. Flocking algo-

rithm for autonomous flying robots. *Bioinspiration & biomimetics*, 9(2):025012, 2014.

[144] Steven N. Ward and Simon Day. Cumbre Vieja Volcano-Potential collapse and tsunami at La Palma, Canary Islands. *Geophysical Research Letters*, 28(17), 2001.

[145] Weber and Timothy R. An analysis of Lemmings: a swarming approach to mine countermeasures in the VSW/SZ/BZ. 1995.

[146] Wei Li, J.A. Farrell, Shuo Pang, and R.M. Arrieta. Moth-inspired chemical plume tracing on an autonomous underwater vehicle. *IEEE Transactions on Robotics*, 22(2):292–307, 4 2006.

[147] Cynthia Werner, Peter J. Kelly, Michael Doukas, Taryn Lopez, Melissa Pfeffer, Robert McGimsey, and Christina Neal. Degassing of CO2, SO2, and H2S associated with the 2009 eruption of Redoubt Volcano, Alaska. *Journal of Volcanology and Geothermal Research*, 259:270–284, 6 2013.

[148] Kieran Wood, Emma J. Liu, Tom Richardson, Robert Clarke, Jim Freer, Alessandro Aiuppa, Gaetano Giudice, Marcello Bitetto, Kila Mulina, and Ima Itikarai. BVLOS UAS Operations in Highly-Turbulent Volcanic Plumes. *Frontiers in Robotics and AI*, 7, 2020.

[149] Xin Xi, Matthew S. Johnson, Seongeun Jeong, Matthew Fladeland, David Pieri, Jorge Andres Diaz, and Geoffrey L. Bland. Constraining the sulfur dioxide degassing flux from Turrialba volcano, Costa Rica using unmanned aerial system measurements. *Journal of Volcanology and Geothermal Research*, 325:110–118, 10 2016.

[150] Yan Yang, Samia Souissi, Xavier Défago, and Makoto Takizawa. Fault-tolerant flocking for a group of autonomous mobile robots. *Journal of systems and Software*, 84(1):29–36, 2011.