SANKALP ARORA

PROPOSAL DRAFT: DATA GATHERING IN PHYSICAL SPACES

ROBOTICS INSTITUTE, CARNEGIE MELLON UNIVERSITY

PUBLISHED BY ROBOTICS INSTITUTE, CARNEGIE MELLON UNIVERSITY

First printing, April 2017

Contents

List of Figures

List of Tables

Introduction

Data acquisition is relatively easy in the virtual world, where the cost of accessing data can be equivalent to accessing a memory block. But the cost of data gathering in physical spaces, where it is impractical to have sensory networks is not as trivial. Currently we rely on humans to carry or drive sensors around to digitize the physical world to collect data.

Gathering data from an oil pipeline to predict future leaks or detect current ones, collecting data from a lake to sample spatial distribution of water contaminants to predict outbreak of diseases, gathering data about flood survivors to aid in search and rescue missions, or to collect data about bridges to predict corrosion rates. All these applications require data gathering in the physical realm at an unprecedented scale. Using humans to do these tedious, boring and often risky tasks is far from ideal. Therefore, the need is urgent to develop autonomous robots to do the job for us.

We need safe data gathering agents that can effectively gather data while reasoning about their physical and sensory constraints. Unmanned Aerial Vehicles (UAVs) are well suited for such data gathering tasks. The primary reason is their ability to reach vantage points unattainable by other systems. UAVs are able to switch from viewing large spaces at a distance to flying in closely to obtain more accurate information. This capability of gaining information at different scales makes UAVs excellent for aforementioned applications.

Therefore last decade has seen a massive growth in applications for Unmanned Aerial Vehicles (UAVs), due in large part to their ability to quickly and inexpensively gather information with cameras, LiDAR and various other sensors. This makes them invaluable for applications such as search and rescue , insfrastructure inspection [1], surveillance , crop and wildlife monitoring , and several more.

However, they are limited in the distance they can cover before their power runs out. Therefore, we must be able to plan routes that maximize the usefulness of the information gained by the vehicle while respecting a cost (power) budget.

A common trend in these applications is that not all possible loca-

[1] Luke Yoder and Sebastian Scherer. Autonomous Exploration for Infrastructure Modeling with a Micro Aerial Vehicle. In *FSR 2015*, 2015; and Andreas Bircher, Kostas Alexis, Michael Burri, Philipp Oettershagen, Sammy Omari, Thomas Mantel, and Roland Siegwart. Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6423–6430. IEEE, 2015

tions are of equal value; we are usually more interested in gathering information about specific targets, such as victims, vehicles, animals, etc. Often we do not know in advance the location of these targets, making it necessary to locate them before more detailed inspection. For example, in a flooded region we might be interested in scouting for survivors and collect data to ascertain their state and number. To illustrate the benefits of this approach, consider the scenario depicted in **??**, where we wish to make 3D models of cars located in the area. The car in the scene is highlighted in red. Using a standard "lawnmower" aerial surveying pattern of the whole block results in several hundreds of pictures with low resolution information of cars. On the other hand, a human operator using First Person View (FPV) control quickly guided the MAV towards the car and captured high resolution imagery of the car, from which a high quality model was reconstructed using less than a hundred frames.

Current robotics systems have severe difficulties with this kind of task. We have identified two major obstacles: the first is the inability to quickly extract from visual imagery the semantic and coarse geometric information required for the understanding the environment. For example - understanding what and where is a building of interest given the imagery? the second is the lack of a tractable algorithm for creating effective, feasible information-gathering plans online; This proposal addresses both of these problems. *In summary, our proposed algorithms will enable semantically guided autonomous exploration through UAVs. The core contributions are a tractable information gathering planner and a learning-based system for fast semantic and geometric scene understanding.* The major challenges in enabling efficient semantically guided exploration are –

- Online Semantic labelling of distant objects
- Reasoning about multi-resolution nature of information gain.
- Online long range obstacle avoidance without the use active range sensors.
- Efficient processing of data for accurate reconstruction.

This paper describes a system that addresses these challenges.

The key contribution of this paper is an anytime, provably optimal algorithm, Randomized Anytime Orienteering(RAOr), that can efficiently solve for routes that maximize correlated reward functions subject to constraints on route length. The key ideas behind RAOr is to restrict the search space to routes that incur minimum distance to visit selected nodes, and rapidly search this space using random sampling.

We provide empirical and theoretical analysis of RAOr. We prove it to be optimal while providing convergence rate analysis. Empiri-

cally the algorithm outperforms the state of the art by more than an order of magnitude in terms of run time required to solve benchmark problems near optimally.

Micro-aerial vehicles have long promised to be the agile sensing platforms of the future. MAV applications like stealth reconnaissance, search and rescue and cargo delivery etc. need fast aerial vehicles moving autonomously in cluttered environments, at low altitudes. Hence, fast and safe obstacle avoidance has remained an active research area. Achieving safe, autonomous, fast flight through cluttered environments on MAVs, presents two main challenges. *The need for a large sensing horizon to allow for adequate time to detect and avoid obstacles and fast and accurate world representation update for minimal latency in reacting to newly discovered obstacles, (Figure* **??***).*

Both of these challenges need to be addressed while keeping the sensing and computational payload to a minimum, to allow for maximizing MAV's flight time and agility. However, current state of the art systems that have demonstrated reliable autonomous flights in cluttered environments have either done so through active sensors like lidars [2], sacrificing agility and range of sensing, or have relied on monocular cameras and data driven techniques to provide proof of concept implementation in a specific environment [3]. Both schools of thoughts have led to pioneering demonstrations of obstacle avoidance capabilities of MAV's albeit at low speeds, either due to restricted sensing range or slow world/robot state updates.

I Navigation through partially known environments, localization, mapping and manipulation of objects etc. are all tasks for which the robot is expected to detect objects, free space or features in the environment. The rate and accuracy of detection of objects, free space or features dictate the performance of the robotic system. Sensors to detect such features of the environment are often limited by their properties like field of view (FOV), resolution, sampling rate and signal to noise ratios. These sensors are either fixed [4] with respect to the vehicle or in some cases move in constant pre-computed patterns [5]. The capabilities of such sensors and the robotic system can be augmented by actively controlling the sensor configuration to minimize the cost of completing the task assigned to the robot.

Robotic applications like cargo delivery, surveillance, people transport, reconnaissance etc. require the robots to operate in unstructured, partially known environments at high speeds. The robots should ensure safety while navigating in such environments without compromising on performance. A popular method to guarantee safety relies on limiting the vehicle speed such that it can come to a stop using longitudinal deceleration within the known obstacle-free volume [6]. This method fails to fully exploit either the vehicle's

[2] Sebastian Scherer, Joern Rehder, Supreeth Achar, Hugh Cover, Andrew Chambers, Stephen Nuske, and Sanjiv Singh. River mapping from a flying robot: state estimation, river detection, and obstacle mapping. *Autonomous Robots*, 33(1-2):189–214, 2012b

[3] Debadeepta Dey, Kumar Shaurya Shankar, Sam Zeng, Rupesh Mehta, M Talha Agcayazi, Christopher Eriksen, Shreyansh Daftry, Martial Hebert, and J Andrew Bagnell. Vision and learning for deliberative monocular cluttered flight. In *Field and Service Robotics*, pages 391–409. Springer, 2016

[4] Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *International Symposium on Robotics Research (ISRR)*, pages 1–16, 2011; Paul Furgale and Timothy D Barfoot. Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560, 2010; and Cooper Bills, Joyce Chen, and Ashutosh Saxena. Autonomous mav flight in indoor environments using single image perspective cues. In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 5776–5783. IEEE, 2011

[5] Sebastian Scherer, Joern Rehder, Supreeth Achar, Hugh Cover, Andrew Chambers, Stephen Nuske, and Sanjiv Singh. River mapping from a flying robot: state estimation, river detection, and obstacle mapping. *Autonomous Robots*, 33(1-2):189–214, 2012b; and Sebastian Scherer, Lyle Chamberlain, and Sanjiv Singh. Autonomous landing at unprepared sites by a full-scale

dynamics or known unoccupied volume, leading to unsatisfactory performance. Another method includes planning a trajectory such that its initial part takes the robot towards the goal while it ends in a control invariant set that lies within the known obstacle-free region [7] . Although this method fully exploits the known space it limits the planning horizon.

In this work, we examine the problem of ensuring safety for mobile autonomous systems while maintaining the capability to operate the vehicle at its performance limits. The key idea is to ensure that the vehicle is always in a safe state from which it can transition to a loiter pattern or come to a stop within the known obstacle-free space. All these states are inside the control invariant set of the robot [8], which is a well-known approach to ensure feasibility for model predictive control applications [9]. Determining loiter patterns or trajectories resulting in complete stop in various environments is computationally challenging especially when the robot has non-linear dynamics. Additionally, it is required that the safety evaluation has a low worst-case response time so that it can be used for on-line motion planning at high speeds.

In order to ensure the on-line capability of the safety evaluation, the problem is decoupled in an off-line and an on-line part. The off-line part generates an optimized set of trajectories allowing the robot to reach a safe state and stay within the known obstacle free region for an infinite time horizon. The trajectory set is designed to maximize the probability of finding at least one emergency maneuver in the known unoccupied environment. The on-line part determines if the set contains a collision-free trajectory regarding the current state and environment of the robot. Thereby, the off-line generated trajectory set reduces the search space for the on-line part. This safety evaluation approach serves as a computationally tractable algorithm with bounded run time. Our main contributions are as follows:

- Generation of an emergency maneuver library that allows for on-line safety assessment of robotic systems at high speeds in unknown environments
- Efficient algorithm for the library generation with bounded sub-optimality
- Experimental evaluation from field tests with an autonomous full-sized helicopter flying at speeds of $56^{m}/_{s}$

[7] Tom Schouwenaars, Jonathan How, and Eric Feron. Receding horizon path planning with implicit safety guarantees. In *American Control Conference, 2004. Proceedings of the 2004*, volume 6, pages 5576–5581. IEEE, 2004

[8] F. Blachini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999

[9] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993

Background

Decision theoretic formulation

Under the decision theoretic regime, autonomous data gathering agents integrate the actions and observations made by the robot to update robot's belief about the environment and select the action that maximizes a reward function in expectation. Let $\mathcal{U}$ be action space of the robot and $\mathcal{U}_s$ be the action space of the sensor, independent of the robot. Let the high dimensional state space of the environment be given by $\mathcal{M}$, where the environment includes the state of the sensor and the robot in the environment. Let, $b \in \mathcal{B}$ be the probability distribution over the state of the environment, $b : \mathcal{M} \rightarrow [0, 1]$. For the k^{th} planning cycle let the action taken by the robot be $u_k \in \mathcal{U}$, the sensory action be $u_k^s \in \mathcal{U}_s$. Let the . Let $o_k \in \mathcal{O}$ be the observations taken by sensors just after actions $\hat{u}_k = [\{u_k\}, \{u_k^s\}]$ are performed. The reward function, $R : \mathcal{U}X\mathcal{U}_sX\mathcal{B} \rightarrow \mathbb{R}$, is belief dependent and assigns rewards to belief, action pairs. The dynamics constraints on the robot and the sensors are given by $h(u_k, u_{k-1}, u_k^s, u_{k-1}^s) \leq 0$. The safety constraint is represented by $s_A(b_k, u_k) \leq 0$. The cost of the trajectory is given by $\sum_{k=1:N} C(u_k)$. And the total budget for the information gathering mission be given by B. Let, $\hat{u} = [\hat{u}_0, \hat{u}_1, \hat{u}_2 ... \hat{u}_N]$ be the set planned actions for the length of the mission, given by N.

$$\arg\min_{\hat{u}} \sum_{k=0:N} R(u_k, u_k^s, b_{k-1}) \tag{1}$$

$$\begin{aligned} h(u_k, u_{k-1}, u_k^s, u_{k-1}^s) \leq 0 \forall k = 0 : N \\ s_A(b_k, u_k) \leq 0 \\ b_k(w) = \alpha P(o_k|w) \sum_{\omega \in W} P(w|u_k, u_k^s, \omega) b_{k-1}(\omega) \\ \sum_{k=1:N} C(u_k) \leq B \end{aligned}$$

Where, α denotes a constant normalizer.

Related Work

Guaranteeing Safety

Substantial literature exists on ensuring safety for autonomous vehicles. We focus our attention on ensuring safety for static, uncertain environments.

Wikman et al. [10] introduced the notion of *Inevitable Collision State* (ICS), a state for which irrespective of future vehicle trajectory, a collision is inevitable. The general approach to safety is to avoid ICS, while assuming unknown regions in the environment to be obstacles. Some of the early work for ground robots relied on making sure that vehicle can stop within sensor range while applying maximum longitudinal deceleration [11]. This technique is effective for vehicles that have the capability to apply large longitudinal decelerations. Similar approach to safety has been adopted by aerial vehicles as well [12]. However, stopping distance based velocity limit does not exploit the complete dynamics of the vehicle, leading to conservative velocity limits.

Another paradigm is to simplify the non-linear dynamics of the UAVs and plan a path that is guaranteed to stay within the known unoccupied region. Mixed integer linear programming is used in [13] to plan paths that stay within the known region. Simplified dynamics in a sampling based graph is used in [14] while limiting the maximum planning time to ensure safety. The assumption is that the planner can always plan an obstacle free path if allowed to run until the maximum planning time. [15] uses Dubins curves to plan paths within the known space. Simplified of dynamics, coupled with a reliance on a planner to generate a safe path online leads to conservative robot behavior that does not fully exploit sensory and dynamics capabilities of the vehicle.

We present an emergency maneuver library based method that utilizes the true dynamics of the vehicle to find a positive control invariant set in the known unoccupied space. We formulate the problem of generating this library as a NP hard path survivability optimization [16]. We prove the path diversity problem to be monotonic, sub-modular leading to an efficient, bounded sub-optimal algorithm [17] to generate the trajectory set. In this section we discuss four exploration strategies that we consider as the state of the art, to make performance evaluation in different environments.

Planning for Data Gathering

The problem of planning for data gathering is a well studied one and finds its roots in sequential hypothesis testing [18]. Sequential

[10] Thierry Fraichard and Hajime Asama. Inevitable collision statesâĂŤa step towards safer robots? *Advanced Robotics*, 18(10):1001–1024, 2004b

[11] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1): 23–33, 1997

[12] Sebastian Scherer, Joern Rehder, Supreeth Achar, Hugh Cover, Andrew Chambers, Stephen Nuske, and Sanjiv Singh. River mapping from a flying robot: state estimation, river detection, and obstacle mapping. *Autonomous Robots*, 32(5):1–26, 2012c; C Goerzen and M Whalley. Minimal risk motion planning: a new planner for autonomous UAVs in uncertain environment. In *AHS International Specialists Meeting on Unmmaned Rotorcraft, Tempe, Arizona*, 2011; and F Adolf and J. S. Dittrich. Evaluation of the ARTIS sampling-based path planner using an obstacle field navigation benchmark. In *Proceedings of the American Helicopter Society 68th Annual Forum. Ft. Work, TX*, 2012

[13] Tom Schouwenaars, Jonathan How, and Eric Feron. Receding horizon path planning with implicit safety guarantees. In *American Control Conference, 2004. Proceedings of the 2004*, volume 6, pages 5576–5581. IEEE, 2004

[14] E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *AIAA Journal of Guidance and Control*, 25(1):116–129, 2002

[15] J. J. Enright, E. Frazzoli, K. Savla, and F. Bullo. On multiple UAV routing with stochastic targets: Performance bounds and algorithms. In *Proceedings AIAA Guidance, Navigation, and Control Conference and Exhibit, Aug. 2005*

[16] Debadeepta Dey, Tian Yu Liu, Boris Sofman, and Drew Bagnell. Efficient optimization of control libraries. Technical report, 2011

[17] Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. Technical report, 2007

[18] Abraham Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945

hypothesis testing have since been extended to account for mobile sensing within the framework of Bayesian reasoning [19]. The work on planning for data gathering can be broadly classified into three paradigms, decision theoretic approaches, myopic information theoretic approaches and non-myopic approximate solvers.

The problem of planning routes to gain information is an NP-hard problem [20]. Decision theoretic approaches to information gathering have been improved over the years either by computing approximate solutions through sampling the action space [21] or by restricting the space of problems to metric spaces, [22]. Chen et al. presented how POMDP-Lite [23] can be used to solve large information gathering problems in a relatively small time by augmenting the reward function with information theoretic rewards. Often, to reduce the run-time of these solvers, the maximum likelihood observations are used instead to define deterministic belief space dynamics[24]. Nonetheless, the computational requirements of these decision theoretic solvers render them unusable in an online setting, especially for MAVs, with limited computation carrying capabilities.

To circumvent the large run-times of decision theoretic approaches, Yamauchi et al.[25] suggested myopic approach for information gathering. Later works have resulted in various of solutions that incorporate information theoretic measures for problems like object recognition, mapping, and scene reconstruction [26]. While such algorithms have shown to be useful in their respective applications, they typically rely on restrictive assumptions on the representations, objective functions and do not have guarantees on global optimality. Finite-horizon model predictive control methods [27] provide improvement over myopic techniques, but they do not have performance guarantees beyond the horizon depth and the run times to operate online in large state-spaces. Charrow et al.[28] present a finite horizon, information theoretic approach where a set of sensing locations is identified and the system travels to the most promising one. While such strategies are computationally efficient, they fail to effectively account for the constraint on traveling distance. As a result the computed routes can lead to oscillatory behavior.

Another approach is to invoke a long horizon planner [29]. However, these approaches are far from real-time. The recursive greedy algorithm [30], Branch and bound [31] require computation exponential in the size of the problem instance due to the large blowup in the search space with increasing budget. An alternative is to utilize a finite-horizon solver that solves the problem for only a portion of the budget at a time [32]. Given the run-times of these algorithms, there is a need for faster methods to compute informative routes. We present Randomized Anytime Orienteering(RAOr), that can ef-

[19] Alec Cameron and Hugh Durrant-Whyte. A bayesian approach to optimal sensor placement. *The International Journal of Robotics Research*, 9(5):70–88, 1990

[20] Andreas Krause. *Optimizing Sensing: Theory and Applications*. PhD thesis, Carnegie Mellon University, December 2008

[21] Joelle Pineau, Geoff Gordon, Sebastian Thrun, et al. Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI*, volume 3, pages 1025–1032, 2003

[22] Zhan Wei Lim, David Hsu, and Wee Sun Lee. Adaptive informative path planning in metric spaces. *The International Journal of Robotics Research*, 35(5):585–598, 2016

[23] Min Chen, Emilio Frazzoli, David Hsu, and Wee Sun Lee. Pomdp-lite for robust robot planning under uncertainty. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 5427–5433. IEEE, 2016

[24] Kaijen Hsiao, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Robust belief-based execution of manipulation programs. In *Eighth Intl. Workshop on the Algorithmic Foundations of Robotics*. Citeseer, 2008

[25] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151. IEEE, 1997

[26] Shengyong Chen, Youfu Li, and Ngai Ming Kwok. Active vision in robotic systems: A survey of recent developments. *The International Journal of Robotics Research*, 30(11):1343–1377, 2011

[27] Frederic Bourgault, Tomonari Furukawa, and Hugh F Durrant-Whyte. Coordinated decentralized search for a lost target in a bayesian world. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 48–53. IEEE, 2003; and Allison Ryan and J Karl Hedrick. Particle filter based information-theoretic active sensing. *Robotics and Autonomous Systems*, 58(5): 574–584, 2010

[28] Benjamin Charrow, Sikang Liu, Vijay Kumar, and Nathan Michael. Information-theoretic mapping using cauchy-schwarz quadratic mutual information. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4791–4798. IEEE, 2015

[29] Jingjin Yu, Mac Schwager, and Daniela Rus. Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks. In *Intelligent*

ficiently solve for routes that maximize correlated reward functions subject to constraints on route length in chapter **??**.

Information Gathering Systems

There have been multiple information gathering systems with significant field results. We briefly describe relevant systems and their planning paradigms. Starting with the river exploration MAV by the Air lab at FRC, CMU [33]. The MAV used myopic frontier based planning, to follow a river and map its banks. The system worked in GPS-denied environments using visual odometry for state estimation. It primarily relied on using lidar for detecting water surface of the river and mapping its banks. Similar, sensor suite and data gathering methods were used by the bridge inspection MAV[34].

Infrastructure inspection often has hard constraints on the resolution at which the information needs to be gathered. Bircher et al. [35] leverage such constraints and the model of infrastructure to be inspected to generate an optimized information gathering route. Given the limited sensing range or the sensing constraints, the information gathering planners in the above use cases do not need to reason about multi-resolution information gathering. Both these approaches assume they have enough battery to complete the mission and do not consider the flight time or budget constraint while gathering information.

Wettergreen et al. [36] use relatively low resolution satellite imagery to guide a ground rover to efficiently collect science data. The onboard informative path planner uses a variant of a recursive greedy, generalized cost benefit algorithm [37]. The methodology reasons about multi-resolution nature of data gathering while applying an approximate algorithm of planning for data gathering under budget constraints. However, the rover itself gathers data at a constant resolution. Jones et al.[38] use a similar one-step greedy approach to reason about multi-resolution data gathering for mapping craters on lunar surface.

There is a need for developing safe, multi-resolution data gathering system that does not rely on myopic, greedy methods to overcome oscillatory behavior and enable efficient information gathering under budgeted constraints.

[33]

[34] Luke Yoder and Sebastian Scherer. Autonomous exploration for infrastructure modeling with a micro aerial vehicle. In *Field and service robotics*, pages 427–440. Springer, 2016

[35] Andreas Bircher, Kostas Alexis, Michael Burri, Philipp Oettershagen, Sammy Omari, Thomas Mantel, and Roland Siegwart. Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6423–6430. IEEE, 2015

[36] David Wettergreen, Greydon Foil, Michael Furlong, and David R Thompson. Science autonomy for rover subsurface exploration of the atacama desert. *AI Magazine*, 35(4):47–60, 2014

[37] Haifeng Zhang and Yevgeniy Vorobeychik. Submodular optimization with routing constraints. In *AAAI*, pages 819–826, 2016

[38] Heather L Jones, Uland Wong, Kevin M Peterson, Jason Koenig, Aashish Sheshadri, and William L Red Whittaker. Complementary flyover and rover sensing for superior modeling of planetary features. In *Field and Service Robotics*, pages 415–429. Springer, 2014

Completed Work

Global Exploration Planning

This section explains the global planning problem and why its important to solve it.

Problem Formulation

Let us formally define the problem of maximizing the reward in a given experiment while respecting a traveling budget B. Let $V = [v_1, v_2, v_3, ..., v_n]$ be the set of all sensing location in the workspace. Let the robot start from node $v_s \in V$ and end at node $v_e \in V$. Let $r = [v_s, v_1^r, ..., v_m^r, v_e]$ be an ordered set of the sensing locations where, $r : r \subseteq V$. For each r, let $I(r)$ be the reward gained by visiting each location in P. Let the cost of traversal be given by $C(r) = \sum_{i=1:|r|-1} C(v_i^r, v_{i+1}^r)$, where v_i is the i^{th} element in r , $\forall i \in [1, |r|]$. The problem then is defined by equation 2.

$$\arg\max_{r \subseteq V} I(r) \text{ subject to } C(r) \leq B \tag{2}$$

The Orienteering Problem (OP) [39] work is closely related to the exploration problem. The Orienteering Problem defines the reward at nodes to be independent of each other or in other words the the reward function is modular. Constant approximation ratio of $(2 + \epsilon)$ were given by [40], but such guarantee is unsatisfactory in practice. On the other hand Mixed Integer Programing(MIP) based solutions exist for OP and related problems[41]. But these solutions fail to capture the reward relationship amongst nodes, leading to sub-optimal paths.

[42] established that information gain is sub-modular and monotonic. Nemhauser in 1978, [43] provided an efficient method to optimize submodular functions. Unfortunately adding the traveling budget constraint makes the problem non-submodular and non-monotonic. [44] suggested using a *recursive-greedy* to find approximate solutions for the orienteering problem if the reward function is submodular. Due to large run-times none of these solutions scale well to

[39] Bruce L Golden, Larry Levy, and Rakesh Vohra. The orienteering problem. *Naval Research Logistics (NRL)*, 34(3): 307–318, 1987

[40] Chandra Chekuri, Nitish Korula, and Martin Pál. Improved algorithms for orienteering and related problems. *ACM Transactions on Algorithms (TALG)*, 8(3): 23, 2012

[41] Pieter Vansteenwegen, Wouter Souffriau, and Dirk Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, 2011

[42] Andreas Krause. *Optimizing Sensing: Theory and Applications*. PhD thesis, Carnegie Mellon University, December 2008

[43] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978

[44] Chandra Chekuri and Martin Pal. A recursive greedy algorithm for walks in directed graphs. In *Foundations of Computer Science*. IEEE, 2005; and Amarjeet Singh, Andreas Krause, Carlos Guestrin, and William J Kaiser.

real world problems. The runtimes of most of the algorithms exceed 3 minutes on a standard desktop PC for a graph of more than 100 nodes.

The MIP based methods solve for linearly relaxed versions of the problem and then impose integer constraints. This results in the method spending most of its time finding partial solutions that do not meet the budget constraints. Also, the state of the art algorithms optimize for the nodes to visit and the sequence in which to travel those nodes together. This leads to the algorithms searching a huge space of solutions for which the sequence of nodes traversed is sub-optimal.

The large run times and the failure to model the reward relationships amongst nodes, necessitates the development of better exploration planning algorithms for efficient autonomous information gathering systems. In the following section we propose reformulating the problem as a combination of Constraint Satisfaction and Traveling Salesman Problem to overcome these limitations.

From Orienteering to Set Selection and TSP

The solution to the correlated orienteering problem is a route in a graph, such that the reward attained by the route is maximized while the path cost stays within a specified value. We break the problem of finding the route, into finding the set of locations to visit and then finding the optimal order in which to visit those locations. Since, we assume the reward function is independent of the order in which the set is visited. This allows the set selection and set order optimization to run independently, without affecting the reward attained by a set. The set selection problem can be formulated as a Constraint Satisfaction Problem (CSP) and finding the optimal order for the selected set is studied as the Traveling Salesman Problem (TSP). In the following we pose the correlated orienteering as a combination of CSP and TSP. The resulting algorithm (RAOr) is presented in 1 and described in

Constraint Satisfaction Problem

In this section we describe how the Correlated Orienteering problem can be viewed as a Constraint Satisfaction Problem. Let r^* be the solution to equation 2. Let, V^* be the set of nodes that are present in r^*. Let, $a_r \in \{0,1\}^{|V|}$ signify the presence of the nodes in a route r, such that $a_r^i = 1$ if $v_i \in r$ and $a_r^i = 0, otherwise$. In order to solve the correlated orienteering problem we want to find the assignment a_{r^*} and then the optimal order in which nodes belonging to r^*, should be

Algorithm 1: Randomized Anytime Orienteering(RAOr)

Input: $G = [V, E]$, v_s, v_e, B, T_r
Output: The best route found in run-time T_r
1 $S = SampleSet(V)$ `// Random set is picked such that it contains` v_s `and` v_e
2 $r = TSP(S, v_s, v_e)$
3 $r_{best} = \phi$
4 **if** $RouteLength(r) <= B \wedge Reward(r) > Reward(r_b)$ **then**
5 $r_{best} = r$
6 **end**
7 **for** $i = 1 : 3|V|$ **do**
8 $v_{new} = Sample(V)$ `// sample a node`
9 **if** $IsInRoute(r, v_{new})$ **then**
10 $r = DeleteFromRoute(r, v_{new})$
11 **end**
12 **else**
13 $r = AddToRoute(r, v_{new})$
14 **end**
15 **if** $RouteLength(r) <= B \wedge Reward(r) > Reward(r_{best})$ **then**
16 $r_{best} = r$
17 **end**
18 **end**
19 **return** r

visited.

Randomized technique for efficient search for satisfying assignment of a binary tuple was presented in 1999 in [45] as a solution to the *CSP* problem. RAOr employs the same technique to search for the optimal assignment of $a = a_{r^*}$, *i.e.* randomly flipping the assignment of one of $|V|$ bits of a.

[45] Uwe Schöning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 410–414. IEEE, 1999a

Traveling Salesman Problem

Once the correct set of nodes are found, then finding the optimal route just requires finding the order in which they need to be visited. Traveling Salesman Problem solvers can provide us with a near-optimal order in polynomial time, [46]. Hence combining the TSP and CSP solvers allows us to develop an algorithm to solve the correlated orienteering problem near optimally and efficiently.

[46] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, DTIC Document, 1976b

Randomized Anytime Orienteering

In this section we describe our algorithm (Alg.1) in detail. We highlight its properties and drawbacks with examples. In the next section we will discuss methods to overcome these drawbacks.

The algorithm starts with uniformly sampling a set of nodes $S \subset V | v_s, v_e \in S$ (Alg. 1 Line 1). The order in which S should be visited is computed using a TSP solver (Alg. 1 Line 2). Running a TSP solver searches an exponential space of routes in polynomial time, leading to run time reduction. The generated route is then checked for satisfying the budget constraint and saved as the best available route if it is admissible (Alg.1 Line 2-3).

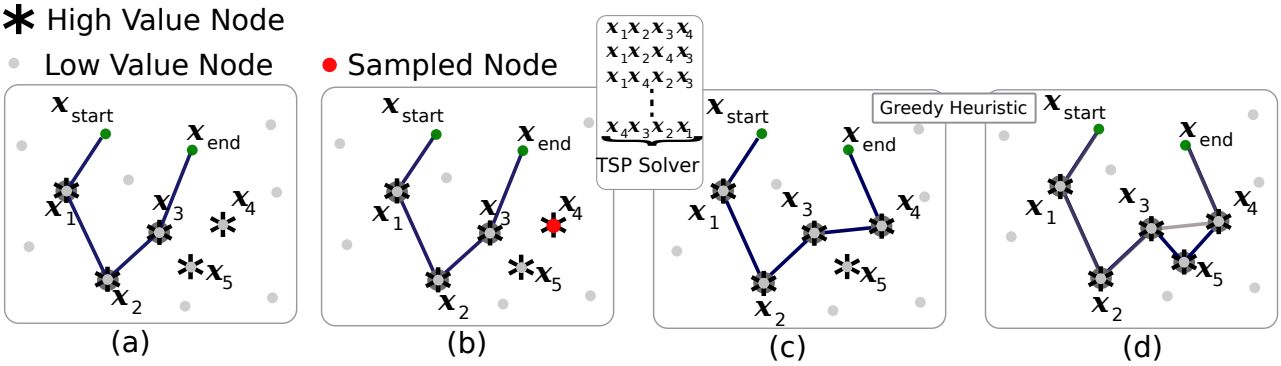

Figure 1: An illustration how RAOr-G combines CSP and TSP solvers in combination with local greedy heuristics to explore the space of routes rapidly, resulting in improvement of run times for finding near optimal solutions for the correlated orienteering problem. a) Using the CSP algorithm the current admissible route is given by $r = \{x_{start}, x_1, x_3, x_{end}\}$. b) At this step x_4 is sampled, and is to be added to the route. c) The exponential number of ways in which x_4 can be added to r is reduced to a near optimal order in polynomial time by the TSP solver. This step reduces an exponential search space with in polynomial computation costs. d) The new route obtained is then improved by conducting a local search using greedy heuristics.

The algorithm then uniformly samples a node $v_{new} \in V$ and changes it's status of being in route r (Alg. 1 Line 6-12). i.e if the node v_{new} was in route r, it is removed from route r or if it was not in route r, it is added to it. The new route obtained is checked for satisfying the budget and saved as best available route, if it exceeds the value of r_{best}. This process is repeated $3|V|$ times.

The algorithm described above is the standard probabilistic algorithm for constraint satisfaction problem as suggested in [47] changed to return the best route available in the budget. We now list some of the properties and drawbacks of the algorithm.

Theorem 1 (Optimality of Randomized Anytime Orienteering)**.** *Randomized Anytime Orienteering algorithm almost surely finds the optimal route from start to end nodes, within budget B, if there exists one, within a polynomial factor of $2(1-1/|V|)^{|V|}$ repetitions.*

The proof of the theorem is provided in the appendix.

Finding the optimal route for the TSP is an NP-Hard in a space that is exponential in number of nodes. But, polynomial time α approximation algorithms, where $\alpha \geq 1$, for a TSP exist in literature [48]. We leverage the polynomial time TSP solver to make Alg. 1 tractable.

Theorem 2 (α-Optimality of Randomized Anytime Orienteering)**.** *Randomized Anytime Orienteering algorithm almost surely finds the optimal route from start to end nodes, within budget B/α, if there exists one, within a polynomial factor of $2(1-1/|V|)^{|V|}$ repetitions. Given that the TSP solver in the inner loop is α-approximate.*

Drawbacks

Uniform Sampling of Sets

RAOr has low run-time for the case where a large number of routes lie within the budget and perform near optimally. But the problem

47 Uwe Schöning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 410–414. IEEE, 1999b

48 Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, DTIC Document, 1976a

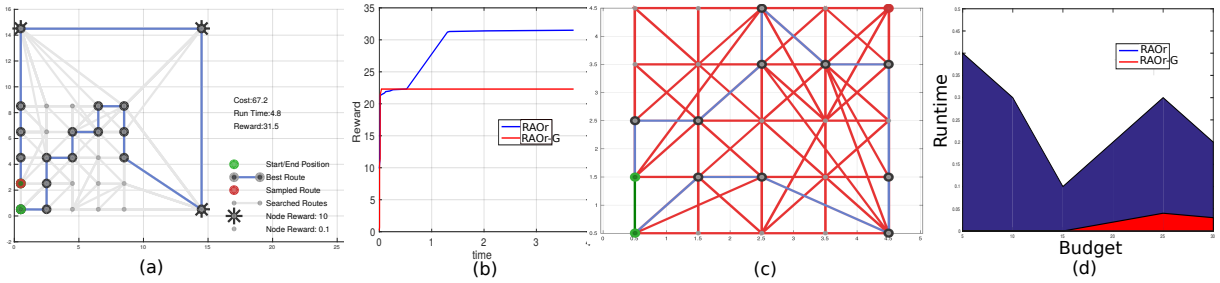

Figure 2: Illustration of drawbacks of RAOr algorithm. a) The problem shown in here consists of three high valued nodes and a budget just sufficient to visit all three. The optimal solution is to visit all three nodes but very few routes exist that have a reward close to optimal. The solution shown is found by running RAOr-G algorithm, Alg. 13. b) Shows the run time vs reward plot for the problem shown in (a) for both RAOr-G and RAOr.RAOr can only attain approximately 66% of the optimal value, as it has a hard time selecting the correct set of nodes. c) Illustrates the problem RAOr faces while running on problems with a small budget. Here a 5X5 grid of nodes, distributed uniformly at a resolution of 1 with a budget of 15 is shown. The routes in red are routes sampled by RAOr that exceed the budget, whereas routes in blue are routes that were sampled that do not exceed the budget. Visibly, red routes outnumber the blue routes. d) The same problem leads to poor run times of RAOr for relatively low budget problems. Here we show the run times of RAOr and RAOr-G on the 5X5 grid shown in (c) while varying the available budget. A runtime of 0 signifies that the algorithm did not converge. Clearly RAOr did not converge for low budget problems in the allotted time. Highlighting the limitation of RAOr in dealing with low budget problems.

scenarios where the probability measure of the optimal path is small the runtime of the algorithm is unacceptable. Fig. 2 (a) and (b) presents a problem where the route has to pass through the three high value nodes to achieve close to optimal reward, while the budget is just sufficient to do so. RAOr needs to be able sample from a highly restricted set of feasible near optimal sets. This leads to large run times.

Sampling Inadmissible Sets

RAOr can potentially spend a lot of time considering sets that are out of budget if budget is small as compared to graph size. As is demonstrated in Fig. 2 (c) and (d).

To alleviate the drawbacks of the problem we improve the algorithm such that it still keeps its global solution finding properties but improve its convergence properties in these pathological cases. These improvements are defined in the next section and the improved algorithm is described in Alg. 13.

Randomized Anytime Orienteering - Greedy (RAOr-G)

The RAOr algorithm (Alg. 1) uniformly samples in the space of sets to find the optimal set of nodes that should be in route. This provides with global solution optimality guarantees while sacrificing on run-times for some pathological cases.

We augment the algorithm to improve its anytime properties by leveraging the problem structure in the following ways -

1. Conducting local searches in the space of routes.

2. Restricting the local search to admissible sets.

3. Informed sampling to improve the likelyhood of sampling high value nodes.

Algorithm 2: RAOr-Greedy(RAOr-G)

Input: $G = [V, E]$, v_s, v_e, B, T_r
Output: The best route found in run-time T_r
$r_g = TSP(v_s, v_e)$ `// Seed local search if no global solution is found`
$R = r_g$
if $RouteLength(r_g) > B$ **then**
  return $\emptyset$
end
$S = SampleSet(V, v_s, v_e)$
$r = TSP(S, v_s, v_e)$
if $RouteLength(r) <= B$ **then**
  $R = R \cup r$
end
for $i = 1 : 3|V| \wedge runtime < T_r$ **do**
  $v_{new} = WeightedSample(V)$
  if $IsInRoute(r, v_{new})$ **then**
    $r = DeleteFromRoute(r, v_{new})$
  end
  else
    $r = AddToRoute(r, v_{new})$
  end
  if $RouteLength(r) <= B$ **then**
    $R = R \cup r$
  end
  $R = GreedyLocalSearch(G, v_{new}, R, B)$
end
return $\arg\max_{r \in R} I(R)$

The resulting algorithm is called Randomized Anytime Orienteering - Greedy (RAOr-G) and is presented in Alg. 13, Fig. 1.

Since, the RAOr-G (Alg. 13)is very similar to RAOr (Alg. 1) we highlight the differences here. RAOr-G is seeded with a route which consists of only v_s, v_e, Alg. 13, Line 1-2. The set of nodes is sampled as in RAOr (Alg. 1) but if the resulting route generated by running TSP on selected nodes lies within budget it is added to the set of feasible routes R, Alg. 13, Line 8-10. R is then used by the local search algorithm as the candidate set for running greedy local search, Alg. 13, Line 22. The *GreedyLocalSearch* function is presented in Alg 5 and described in Section . Another major difference is that instead of uniformly sampling for the node to add or delete from the route, the sampling is weighted in order to sample more valuable nodes often. The intuition behind weighted sampling is described in Section .

Local Greedy Search Heuristic

Local greedy search heuristic is used to improve the runtime of RAOr algorithm and also to improve the anytime performance of the algorithm. The greedy search takes as input a set of feasible routes found so far, R and the sampled node v_{new}. It finds the route $r_c \in R$

Algorithm 3: GreedyLocalSearch

Input: $G = [V, E]$, v_{new}, R, B
Output: Updated set R after conducting local greedy search
$r_c = FindBestRouteInBudget(v_{new}, R)$
if $r_c == \emptyset$ **then**
  $r_{cn} = Route(v_e, v_{new}, v_{new})$ // TSP
end
else
  $r_{cn} = AddToBestRoute(v_{new}, r_c)$
  // Run a TSP with v_{new} added to r_c.
end
$R = R - r_c$
$r_{cn} = AddNearByNodes(G, r_{cn}, B, d, c)$
$R = R \cup r_{cn}$

return R

according to equation 3.

$$r_c = \arg\max_{r \in R} \frac{I(r \cup v_{new}) - I(r)}{RouteLength(r \cup v_{new})} \tag{3}$$

$$\text{subject to } RouteLength(r \cup v_{new}) \leq B$$

The reward of the updated route is further improved by adding nodes that are within distance d and increased the reward gained by the route by atleast c, while keeping it in budget 4.

Algorithm 4: AddNearByNodes

Input: $G = [V, E]$, r_{cn}, B, d, c
Output: Updated set R after conducting local greedy search
forall the $v \in V | DistanceFromRoute(v, r_{cn}) <= d$ **do**
  if $(I(r_{cn} \cup v) - I(r_{cn})) \geq c$ **then**
    if $DistanceFromRoute(v, r_{cn}) * 2 \leq B - RouteLength(r_{cn})$ **then**
      $r_{cn} = AddToRoute(v, r_{cn})$
      // Run a TSP with v added to r_{cn}.
    end
  end
end

return r_{cn}

The total computation cost of a single iteration of local greedy search heuristic is $O(|R||V|)$. In practice we have found that the speedup achieved far outweighs the cost, Fig. 2.

In the next Section we describe the intuition behind using weighted sampling instead of uniform sampling and how it affects the theoretical guarantees provided by the RAOr-G algorithm (Alg 13).

Weighted Sampling

Weighted sampling, samples nodes with the probability directly proportional to the reward they offer independently. The intuition behind weighted sampling is that nodes with high values tend to

be the part of optimal routes. This intuition is very similar to that of constructing macro-actions described in [49]. Unfortunately weighted sampling adversely affects the theoretical guarantees of the algorithm, but empirically it leads to alleviating the problems caused by uniform sampling .

[49] Ruijie He, Emma Brunskill, and Nicholas Roy. Puma: Planning under uncertainty with macro-actions. In *AAAI*, 2010

Theorem 3 (Optimality of Randomized Anytime Orienteering - Greedy)**.** *Randomized Anytime Orienteering - Greedy algorithm almost surely finds the optimal route from start to end nodes, within budget B/α, if there exists one in polynomial factor of $\left(\frac{2}{1+\zeta}\right)^{|V|}$ repetitions, where $\zeta = \left(\frac{((|V|-I_{min})I_{min})^{\beta}}{(|V|-1)^{(\beta+1)}}\right)$, $I_{min} = \min_{v \in V \text{ and } I(v) \neq 0} I(v)$, $\beta = \frac{1}{|V|-2}$.*

Results

We evaluate the computational performance of RAOr on various benchmark examples against the state of the art methods. Then, we apply the algorithm on a realistic coverage scenario in simulation. All simulation computations are performed on a computer equipped with Intel Core i7-4870HQ using Matlab. The algorithm was also deployed on an autonomous exploration UAV system, we present and a analyze an exploration flight.

Computational Performance

The benchmark problem, Fig 3 consists of a graph with nodes located in a grid at uniform resolution. The reward for visiting each cell in the grid is 1. There is no reward for visiting a cell twice. Table 1 shows the comparative run times of state of the art algorithms vs RAOr-G for different problem sizes and varying budgets. Both RAOr-G and eSIP are anytime in nature. Each algorithm was stopped when it reached 95% of the optimal value. RAOr-G outperforms the state of the art in all the benchmark problems. RAOr-G solves the problem with 400 nodes approximately 100 times faster than the eSIP.

Correlated Rewards

In the first scenario we compared the computational performance against algorithms that promise optimality but on a simple case, where rewards of the nodes were independent. Here we evaluate RAOr-G's performance against algorithms that can potentially work on-board robots given their relatively low runtimes.

The algorithm has to find a route given start and end locations such that the reward collected is maximized. Reward at a node is defined by the region visible from a 25^o field of view, downward

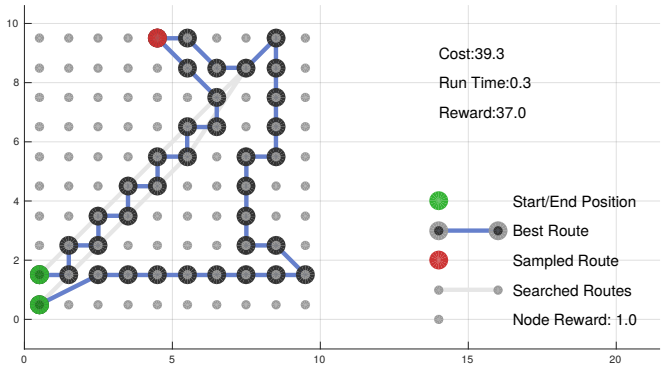

Figure 3: Illustrated here is 10*X*10 grid size benchmark problem. For all the benchmark problems the nodes are situated in a uniform grid with 1 resolution. For this particular problem *RAOr-G* is able to find a near-optimal solution in 6.9 seconds while the state of the art takes 143.6 seconds for finding same quality of solution.

Grid Size	Budget		eSIP	MIQP	RAOr-G
5X5	30	cost	29.8	29.6	29.4
		utility	24	25	24
		time(s)	6.8	15.3	0.5
7X7	60	cost	56.6	57.3	56.1
		utility	48	49	48
		time(s)	26.7	1358.5	2.4
10X10	100	cost	99.0	99.8	98.9
		utility	87	92	87
		time(s)	143.6	15330	6.9
20X20	100	cost	99.7	-	99.2
		utility	87	-	87
		time(s)	763.7	-	7.2

Table 1: Comparative run time analysis of state of the art v/s RAOr-G. RAOr-G consistently outperforms the state of the art in terms of run times for achieving near-optimal solutions. All the solutions obtained are atleast 95% of the optimal.

facing camera. Viewpoints exist on grid of $4m$, at a height of $10m$ and at a uniform distance of $20m$ at a height of $55m$. Resulting a total of 650 nodes/viewpoints. The area is randomly strewn with 5 high value objects, covering those from lower viewpoints results in a 10 times higher reward than other locations. Two distance metrics are implemented, Case1 Euclidean distance and Case2 Euclidean distance with travel along z-direction costing 3 times more vs. traveling in x-y plane, see figure 4. Since nodes/viewpoints are distributed in a uniform fashion, greedy solutions are able to perform near optimally in Case 1.RAOr-G performs comparably to the greedy solutions, while incurring a bit more runtime. In Case 2, greedy solutions are clearly stuck in a local maxima, leading to a much worst performance than RAOr-G. In both cases RAOr-G outperforms RIG algorithm both in runtimes and solution quality,see figure 4.

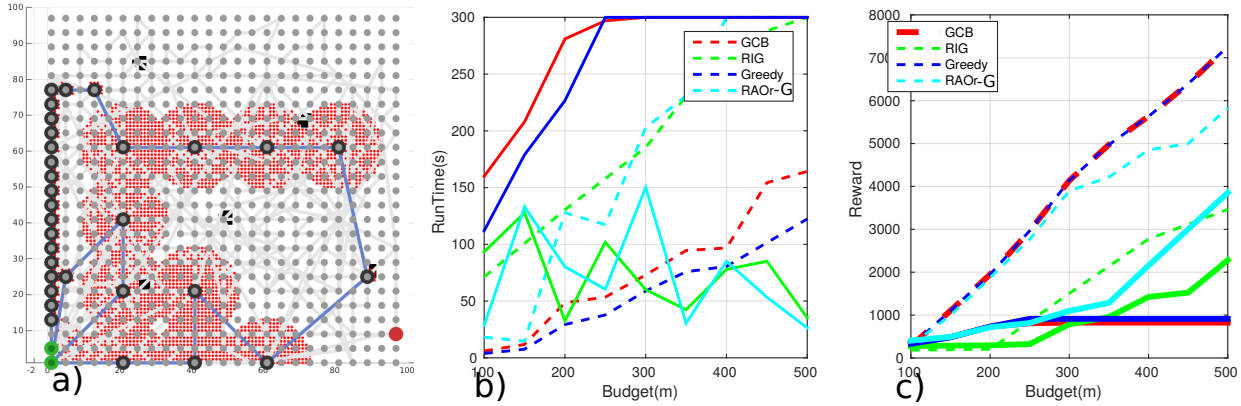

Figure 4: a)RAOr-G planned path with a $500m$ budget on the $100X100$ area for *Case*2. Red marks the sensor footprint. Notice how the path visits high value regions, displayed in black. Grey paths show all the paths searched by RAOr-G.b) and c) Case 1 is displayed in dashed line, Case 2 in solid lines. RAOr-G is competitive with greedy algorithms in Case 1 and dramatically outperforms greedy and RIG for Case 2, where greedy algorithms are stuck in local maxima.

Robotic Exploration

The algorithm was deployed on an autonomous UAV system described in this link-`https://goo.gl/bNr6Du`. It ran at $3Hz$ on an Odroid-C2. Figure 5 describes the vehicle mission. The vehicle is deployed to scout for cars and collect high resolution data if a car is found. The exploration reward function is weighted probability distance between current robot's belief and expected updated belief and a fixed reward of 20000 is allocated to scanning a car, for more details see following link-`https://goo.gl/bNr6Du`.The algorithm is run adaptively as the vehicle's representation is updated and is able to guide the vehicle to explore the environment and locate and map both the cars, see figure 5.

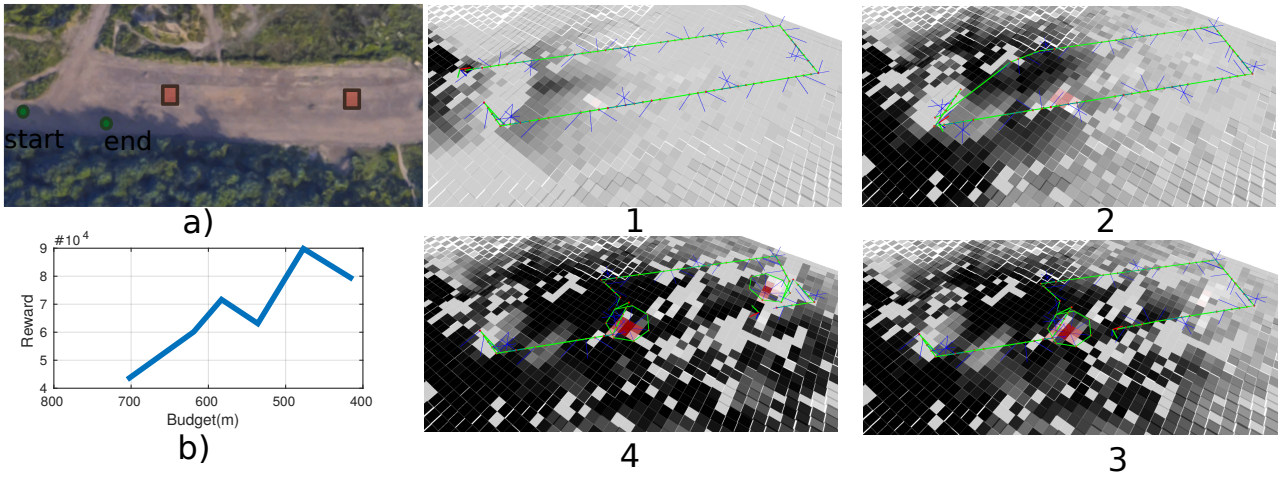

Figure 5: a) Testing site, start and end are marked by green nodes and car locations are shown in orange. b) Vehicle starts with a budget of 700m, the reward increases as likelihood of finding the car increases, the crest in reward marks the time at which the global planner found and decided to map the car. Figures 1,2,3 and 4 show the series of plans at various stages of the exploration mission, Dark squares indicate absence of cars and red squares presence of cars. Shades of grey and red signify certainty. Once the car is recognized, a 360 view of the car is associated with an extra reward.

Appendix

Proposition 1 (Returning optimal route on optimal set selection)**.** *If RAOr finds the optimal set of nodes to visit during it's run-time, it is guaranteed to return the corresponding optimal route, if an optimal TSP solver is used.*

Proof. In every iteration, RAOr (Alg. 1) checks if the set of nodes selected in that iteration can be visited within the budget constraint. The optimal order to visit the selected set is given by a *TSP* solver. If the selected set can be visited within the budget, it is stored as the best set encountered if it is more rewarding than the previous best set encountered. The most rewarding set encountered in the budget is returned. Therefore, if set V^* is encountered it is guaranteed that r^* will be found and returned, given we have an optimal *TSP* solver. Hence, if RAOr finds the optimal set of nodes to visit during it's run-time, it is guaranteed to return the corresponding optimal route, if an optimal *TSP* solver is used. □

We next compute the probability of finding the optimal assignment or equivalently the optimal path using *RAOr*.

Proposition 2 (Probability of finding the correct path)**.** *Randomized Anytime Orienteering algorithm finds the optimal route from start to end nodes, within budget B, if there exists one, with a probability of at least* $\left(\frac{1}{2}\left(1+\frac{1}{|V|-1}\right)\right)^{|V|}$ *in one run.*

Proof. The proof of this theorem is taken from the work on probabilistic algorithm for constraint satisfaction problem [50].

Given a_{r^*} is the optimal solution, we want estimate the lower bound on the probability that Alg. 1 finds a_{r^*}. Once we have found

[50] Uwe Schöning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 410–414. IEEE, 1999a

this success probability p, the expected number of independent repetitions of the procedure until we find the optimal solution is $1/p$.

Now, we calculate p. It is clear that the random variable X that counts the number of bits in which the random assignment a and the fixed assignment a_{r^*} disagree (i. e. the Hamming distance between a and a_{r^*}) is binomially distributed. That is, $Pr(X = j) = \binom{|V|}{j}2^{-|V|}$. If the system is in state 0, this means, an optimal assignment has been found.

At any given point in the algorithm, if a is not the optimal assignment then there must be atleast one vertex out of $|V|$ that needs to be flipped (included or excluded from the set selection) to reduce the hamming distance of a to a_{r^*} reduces by 1. Selecting the correct vertex would mean that the current state transfers j to transfers to state $j-1$ with probability at least $1/|V|$, and transfers to state $j+1$ with probability at most $\frac{|V|-1}{|V|}$. This markov chain is the same as described by [51]. We present the proof of value of p here for completeness.

[51] Uwe Schöning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 410–414. IEEE, 1999a

Given that the process has initially transferred into state j, we calculate the probability q_j that the process reaches the absorbing state 0. For this to happen the process needs at least j steps. We consider the case that the random walk takes $i \leq j$ steps in the "wrong" direction and $i+j$ steps are required toward the "right" direction so that the process stops in state 0 after $j+2i$ steps. To calculate this probability requires us to calculate the number of paths on a rectangular grid (which represents the possible movements over the Markov chain over the time scale) which transfers the process from state j to state 0 while exactly i steps in the "wrong" direction. Using the ballot theorem from [52], page 73, it can be seen that this number is $\binom{j+2i}{i}.\frac{j}{j+2i}$. Therefore, the probability can be estimated as follows.

[52] William Feller. An introduction to probability theory and its applications. vol. i. 1950

$$\begin{aligned} q_j &\geq \sum_{i=0}^{j} \binom{j}{j+2i} \cdot \frac{j}{j+2i} \cdot \left(\frac{|V|-1}{|V|}\right)^i \cdot \left(\frac{1}{|V|}\right)^{i+j} \\ q_j &\geq \frac{1}{3} \sum_{i=0}^{j} \binom{j}{j+2i} \cdot \left(\frac{|V|-1}{|V|}\right)^i \cdot \left(\frac{1}{|V|}\right)^{i+j} \end{aligned} \tag{4}$$

Further we can lower bound the above sum by its largest term as follows. We use the following fact [53]. $\binom{n}{\beta n} \sim 2^{h(\beta)n} = \left(\frac{1}{\beta}\right)^{\beta n}\left(\frac{1}{1-\beta}\right)^{(1-\beta)n}$ where $h(\beta) = -\beta\log_2\beta - (1-\beta)\log_2(1-\beta)$ is the binary entropy function. In particular, the two functions

[53] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Chapman & Hall/CRC, 2010

$$\binom{(1+2\beta)j}{\beta j} \text{ and } \left[\left(\frac{1+2\beta}{\beta}\right)^{\beta} \cdot \left(\frac{1+2\beta}{1+\beta}\right)^{1+\beta}\right]^j \tag{5}$$

are within polynomial factors of each other. We lower bound the

above estimation for q_j by setting $\beta = \frac{1}{|V|-2}$.

$$\begin{aligned} q_j &\geq \frac{1}{3}\sum_{i=0}^{j}\binom{j}{j+2i}\cdot\left(\frac{|V|-1}{|V|}\right)^{i}\cdot\left(\frac{1}{|V|}\right)^{i+j} \\ &\geq \left[\left(\frac{1+2\beta}{\beta}\right)^{\beta}\cdot\left(\frac{1+2\beta}{1+\beta}\right)^{1+\beta}\left(\frac{|V|-1}{|V|}\right)^{\beta}\cdot\left(\frac{1}{|V|}\right)^{1+\beta}\right]^{j} \\ &\text{where } \beta = \frac{1}{|V|-2} \\ &= \left(\frac{1}{|V|-1}\right)^{j} \end{aligned} \tag{6}$$

where the last inequality holds up to some polynomial factor. Therefore, up to some polynomial factor, using the binomial theorem, we obtain the following estimate for success probability p

$$\begin{aligned} p &\geq (\frac{1}{2})^{|V|}\sum_{j=0}^{|V|}\binom{|V|}{j}\left(\frac{1}{|V|-1}\right)^{j} \\ &= \left(\frac{1}{2}\left(1+\frac{1}{|V|-1}\right)\right)^{|V|} \end{aligned} \tag{7}$$

□

Theorem 1 (Optimality of Randomized Anytime Orienteering)**.** *Randomized Anytime Orienteering algorithm almost surely finds the optimal route from start to end nodes, within budget B, if there exists one, within a polynomial factor of $2(1-1/|V|)^{|V|}$ repetitions.*

Proof. According to Proposition 2 the RAOr algorithm finds the optimal set with probability atleast $p \geq (\frac{1}{2})^{|V|}\sum_{j=0}^{|V|}(\binom{|V|}{j})\left(\frac{1}{|V|-1}\right)^{j} = \left(\frac{1}{2}\left(1+\frac{1}{|V|-1}\right)\right)^{|V|}$. The expected number of independent repetitions of the *RAOr* algorithm are $1/p$. The probability that *RAOr* will not find a satisfying assignment after h repetitions is at most $(1-p)^h \leq e^{-ph}$. Therefore, to achieve acceptable chance of missing the optimal path, say e^{-50}, we can choose $h = 50/p$. So the complexity is within a polynomial factor of $1/p$. Therefore, Randomized Anytime Orienteering algorithm almost surely finds the optimal route from start to end nodes, within budget B, if there exists one, within a polynomial factor of $(2(1-1/|V|))^{|V|}$ repetitions. □

Theorem 2 (α-Optimality of Randomized Anytime Orienteering Algorithm)**.** *Randomized Anytime Orienteering algorithm almost surely finds the optimal route from start to end nodes, within budget B/α, if there exists one, within a polynomial factor of $(2(1-1/|V|))^{|V|}$ repetitions. If the TSP solver in the inner loop is α-approximate.*

Proof. If the *TSP* solver is α optimal, if V^* is encountered during set selection, the length of the route covering the set may not fit inside the budget constraint.

Although, if V_b^* is the most rewarding set that can be visited in $\frac{B}{\alpha}$, then it can be guaranteed that if V_b^* set is selected, it will be returned as the most rewarding set by *RAOr*. The probability of finding V_b^* is presented in equation 7 and hence the *RAOr* algorithm almost surely finds the optimal route from start to end nodes, within budget B/α, if there exists one, within a polynomial factor of $(2(1-1/|V|))^{|V|}$ repetitions. If the TSP solver in the inner loop is α-approximate. □

Theorem 3 (Optimality of Randomized Anytime Orienteering - Greedy)**.** *Randomized Anytime Orienteering - Greedy algorithm almost surely finds the optimal route from start to end nodes, within budget B/α, if there exists one in polynomial factor of $\left(\frac{2}{1+\zeta}\right)^{|V|}$ repetitions, where $\zeta = \left(\frac{((|V|-I_{min})I_{min})^{\beta}}{(|V|-1)^{(\beta+1)}}\right)$, $I_{min} = \min_{v \in V \text{ and } I(v)\neq 0} I(v)$, $\beta = \frac{1}{|V|-2}$.*

Proof. $RAOr-G$ relies on the same process as *RAOr* for its global convergence characteristics. The random variable X that counts the number of bits in which the random assignment a_r and the fixed assignment a_{r^*} disagree (i. e. the Hamming distance between a and a_{r^*}) is binomially distributed. That is, $Pr(X = j) = \binom{|V|}{j} 2^{-|V|}$. If the system is in state 0, this means, an optimal assignment has been found.

At any given point in the algorithm, if a is not the optimal assignment then there must be atleast one vertex out of $|V|$ that needs to be flipped to reduce the hamming distance of a to a_{r^*} reduces by 1. Selecting the correct vertex would mean that the current state transfers j to transfers to state $j-1$ with probability at least $\frac{I_{min}}{|V|}$ and transfers to state $j+1$ with probability at most $1-\frac{I_{min}}{|V|}$. The change in probabilities is due to the weighted sampling of nodes instead of uniform sampling.

Substituting these values in equation 6 and following the algebra we get the following.

$$q_j \geq \left(\frac{((|V|-I_{min})I_{min})^{\beta}}{(|V|-1)^{(\beta+1)}}\right)^{j} = \zeta^j \tag{8}$$

where the inequality holds up to some polynomial factor. Therefore, up to some polynomial factor, using the binomial theorem , we obtain the following estimate for success probability p

$$p \geq (\frac{1}{2})^{|V|} \sum_{j=0}^{|V|} \binom{|V|}{j} \zeta^j = \left(\frac{1}{2}(1+\zeta)\right)^{|V|} \tag{9}$$

Hence the number of repetitions required for $RAOr - G$ to find the optimal route is given by a polynomial factor of $\left(\frac{2}{1+\zeta}\right)^{|V|}$. Where ζ is given by $\left(\frac{((|V|-I_{min})I_{min})^{\beta}}{(|V|-1)^{(\beta+1)}}\right)$. □

Emergency Maneuver Library

This section explains the safety problem and why its important to solve it.

Defining Safety

The safety of a mobile autonomous system is dependent on its sensory and dynamic capabilities. In a fully-known environment a mobile system is unsafe if it enters a state for which there exists no trajectory that avoids a collision, such a state is called an Inevitable Collision State [54]. In a static partially-known environment the unknown regions may contain obstacles. Therefore, to ensure the safety of the mobile robot its state should be constrained such that it can always transition to a terminal feasible invariant set [55] that allows the robot to stay within the known obstacle-free volume for an infinite time horizon. We now formally define safety for robots operating in uncertain environments. Let, $\mathbf{x}(t)$ be the state of the robot at time t in the state space $\mathcal{X}$ which is in a manifold $\mathcal{X} \subset \mathbb{R}^n$. The workspace of the robot is defined as $\mathcal{W}$ and the occupancy of the robot system in the workspace at a certain state is given as $\mathcal{A}(\mathbf{x}(t)) \subset \mathcal{W}$. The known space of the workspace at a given time t is denoted as $\mathcal{K}_t \subset \mathcal{W}$. The occupancy of the known obstacles at time t is given by $\mathcal{O}_t \subset \mathcal{K}_t \subset \mathcal{W}$. Let $\Phi_F(\mathbf{x})$ be the search space of trajectories for a given state $\mathbf{x}$, that end in a terminal feasible invariant set. Let $\phi(\mathbf{x})$ be such a trajectory and let $\phi(\mathbf{x}, \tau)$ be the state of the vehicle at time τ, along the trajectory $\phi(\mathbf{x})$, which is by definition rooted at state $\mathbf{x}$. Then any trajectory followed by the vehicle can be considered safe if for all states on the trajectory there exists a trajectory $\phi(\mathbf{x})$ which completely lies inside the known obstacle-free space at that time. Equation (10) presents this definition formally:

[54] T. Fraichard and H. Asama. Inevitable collision states. A step towards safer robots? *Advanced Robotics*, 18(10): 1001–1024, 2004a

[55] Tom Schouwenaars, Jonathan How, and Eric Feron. Receding horizon path planning with implicit safety guarantees. In *American Control Conference, 2004. Proceedings of the 2004*, volume 6, pages 5576–5581. IEEE, 2004

Definition 1 (Motion Safety):

$$\forall t, \forall \tau, \exists \phi(\mathbf{x}) : \mathcal{A}(\phi(\mathbf{x}, \tau)) \subset (\mathcal{K}_t \setminus \mathcal{O}_t) \tag{10}$$

In the next section, we discuss how this safety definition can be enforced on mobile autonomous vehicles in real-time with the use of an emergency maneuver library.

Approach

Finding a trajectory that satisfies (10) online is non-trivial due to computation costs involved, especially if the robot's dynamics are non-linear. The current methods in use, lead to a very conservative behavior, with robots acting well below their dynamics and sensory capabilities. We provide a method that allows for guaranteeing the safety of the vehicle in static environments, while exploiting the limits of vehicle dynamics and considering the available known obstacle-free space. We split the problem into two parts, first, we efficiently compute a reduced set of terminal invariant trajectories offline then use this reduced set to search for safe terminal invariant trajectories online. In the next section we discuss how to compute the terminal invariant trajectory set resulting in the emergency maneuver library. Before discussing how this emergency maneuver library can be used online to ensure safety of the mobile robot.

Emergency Maneuver Library

Instead of solving for dynamically feasible trajectories that end in a terminal invariant set on-line, we approximate the search space by a finite set of such trajectories. The trajectory set is designed such that the probability that at least one of the trajectories stays collision-free, given a prior on the obstacle configuration for a given state is maximized for a static environments. We show that this problem is NP hard [56] and then prove that it is monotonic sub-modular, providing a sub-optimality bound for a greedy algorithm.

[56] Michael S Branicky, Ross A Knepper, and James J Kuffner. Path and trajectory diversity: Theory and algorithms. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1359–1364. IEEE, 2008; and Lawrence H Erickson and Steven M LaValle. Survivability: Measuring and ensuring path diversity. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2068–2073. IEEE, 2009

Problem Definition

We want to find a set of trajectories that are control invariant and maximize the probability that the set contains at least one collision-free trajectory. The trajectory set is optimized for a spatial stochastic process defined in $r \in \mathcal{W}$, that captures the distribution of obstacle configurations that the robot is likely to encounter during its lifetime. We assume this spatial field is give by $\zeta(u, r)$, where u is the event of point r being unoccupied/free. The probability density function defined by this process is then given by $p_\zeta(u, r)$. Probability that there is no obstacle inside a volume $V \subset \mathcal{W}$ is thus given $P_{\mathrm{u}}(V) = \int_V p_\zeta(u, r)dr$. The volume which is swept by the robot following a certain trajectory ϕ is expressed as $V(\mathcal{A}(\phi)) = \{\mathbf{r} | \mathbf{r} \in \mathcal{W}, \exists \tau \ \ \mathbf{r} \in \mathcal{A}(\phi(\tau))\}$, [57]. In the rest of the paper we use the shorter notation $V_\phi = V(\mathcal{A}(\phi))$ to denote the volume swept by the robot. The probability of a path being safe is given by the probability of the

[57] Karim Abdel-Malek, Yingzhou Yang, Denis Blackmore, and Ken Joy. Swept volumes: Foundations, perspectives, and applications. *Int. J. Shape Modeling*, 12(1):87–127, 2006

swept volume being free of obstacles.

$$P_{\mathrm{u}}(\phi) \qquad = P_{\mathrm{u}}(V_{\phi})$$

which allows to determine the probability of path ϕ_1 or ϕ_2 being unoccupied

$$P_{\mathrm{u}}(\phi_1 \cup \phi_2) = P_{\mathrm{u}}(\phi_1) + P_{\mathrm{u}}(\phi_2) - P_{\mathrm{u}}(\phi_1 \cap \phi_2)$$

where,

$$P_{\mathrm{u}}(\phi_1 \cap \phi_2) = P_{\mathrm{u}}(V_{\phi_1} \ \cap \ V_{\phi_2}).$$

Using the inclusion-exclusion principle, we get the probability of a path set $\Phi = \{\phi_1, \phi_2, ..., \phi_n\}$ having an obstacle free path as

$$P_{\mathrm{u}}(\Phi) = \sum_{k=1}^{n} (-1)^{k-1} (\sum_{1 \le i_1 \le ... \le i_k \le n}^{n} P_{\mathrm{u}}(\phi_{i_1} \cap ... \cap \phi_{i_k})). \tag{11}$$

In order to maximize the safety of the robot, a finite path set Φ must be determined maximizing the probability that at least one path is collision-free. This is formulated as the path diversity problem:

Problem 1 (Path Survivability): *The desired trajectory set Φ_d maximizes the probability of finding at least one obstacle-free path.*

$$\begin{aligned} &\Phi_{\mathrm{d}} := \arg\max P_{\mathrm{u}}(\Phi) \\ &\textit{subject to } \|\Phi_{\mathrm{d}}\| < N_{\Phi} \\ &\textit{where, } \Phi_{\mathrm{d}} \subseteq \Phi \subseteq \Phi_{\mathrm{F}} \end{aligned} \tag{12}$$

Φ_{F} is the search space of trajectories. Since the path diversity problem is known to be NP-hard, we present a greedy method to optimize (12). But before that we prove that greedily optimizing equation (12) is bounded sub-optimal. To prove bounded sub-optimality we prove the $P_{\mathrm{u}}(\Phi)$ is sub-modular and monotonically increasing in the cardinality of Φ.

Monotonicity Proof

In the following we show that the probability of at least one path in a path set being collision-free is monotonically increasing by the cardinality of the path set.

Proposition 3 (Monotonicity of Path Sets)**.** *Given a path set Φ_A, a path ϕ_a and a path set $\Phi_B = \{\Phi_A, \{\phi_a\}\}$ the probability that the set Φ_B contains at least one collision-free path is bigger or equal than for the set Φ_A*

$$P_{\mathrm{u}}(\Phi_B) - P_{\mathrm{u}}(\Phi_A) \geq 0.$$

Proof. The probability of the path set Φ_b having at least one obstacle-free path is given by

$$P_u(\Phi_B) = P_u(\Phi_A \cup \{\phi_a\}).$$

Using inclusion exclusion principle lead to

$$\begin{aligned} P_u(\Phi_B) &= P_u(\Phi_A) + P_u(\phi_a) - P_u(\Phi_A \cap \phi_a) \\ P_u(\Phi_B) - P_u(\Phi_A) &= P_u(\phi_a) - P_u(\Phi_A \cap \phi_a). \end{aligned} \tag{13}$$

For all ϕ_a, $P_u(\phi_a) \geq 0$ and $\max[P_u(\Phi_A \cap \phi_a)] = P_u(\phi_a)$, which is the case when all the volume covered by path ϕ_a is already covered by Φ_A. This implies that

$$P_u(\phi_a) - P_u(\Phi_A \cap \phi_a) \geq 0$$

and inserted in (13) leads to $P_u(\Phi_B) - P_u(\Phi_A) \geq 0$ □

Prop. 3 shows that adding more trajectories to a path set, cannot decrease the probability of finding an obstacle-free path in the set. In other words, $P_u(\Phi)$ is monotonically increasing in the cardinality Φ.

Sub-Modularity Proof

In order to show that a greedy algorithm for the path diversity problem (Prop. 1) is bounded sub-optimal, we will show that $P_u(\Phi)$ is a submodular set function. This means, that the difference in the probability P_u that a single trajectory makes when added to the path set decreases as the size of the path set increases.

Proposition 4. *Let there be a path set $\Phi_\Gamma \subseteq \Phi_Y \subseteq V$, where $P_u : 2^V \to R$. Now, assume a path ϕ_e, such that $\phi_e \subseteq V \backslash Y$. Define $\Phi_{\Gamma+e} = \{\Phi_\Gamma, \phi_e\}$, $\Phi_{Y+e} = \{\Phi_Y, \phi_e\}$. For sub-modularity*

$$\triangle(e|Y) < \triangle(e|\Gamma)$$

where, $\triangle(\cdot)$ is the discrete derivative.

Proof. The discrete derivative of $\triangle(e|\Gamma)$ is defined as

$$\begin{aligned} \triangle(e|\Gamma) &= P_u(\Phi_{\Gamma+e}) - P_u(\Phi_\Gamma) \\ &= P_u(\Phi_\Gamma \cup \phi_e) - P_u(\Phi_\Gamma) \\ &= P_u(\Phi_\Gamma) + P_u(\phi_e) - P_u(\Phi_\Gamma \cap \phi_e) - P_u(\Phi_\Gamma) \\ &= P_u(\phi_e) - P_u(\Phi_\Gamma \cap \phi_e) \end{aligned}$$

and similarly

$$\triangle(e|Y) = P_u(\phi_e) - P_u(\Phi_Y \cap \phi_e).$$

Taking the difference of the discrete derivatives

$$\begin{aligned}
\triangle(e|\Gamma) - \triangle(e|\mathrm{Y}) &= \\
&= P_{\mathrm{u}}(\phi_e) - P_{\mathrm{u}}(\Phi_\Gamma \cap \phi_e) - P_{\mathrm{u}}(\phi_e) + P_{\mathrm{u}}(\Phi_\mathrm{Y} \cap \phi_e) \\
&= P_{\mathrm{u}}(\Phi_\mathrm{Y} \cap \phi_e) - P_{\mathrm{u}}(\Phi_\Gamma \cap \phi_e) \\
&= P_{\mathrm{u}}((\Phi_\Gamma \cup \Phi_{\mathrm{Y}/\Gamma}) \cap \phi_e) - P_{\mathrm{u}}(\Phi_\Gamma \cap \phi_e) \\
&= P_{\mathrm{u}}((\Phi_\Gamma \cap \phi_e) \cup (\Phi_{\mathrm{Y}/\Gamma} \cap \phi_e)) - P_{\mathrm{u}}(\Phi_\Gamma \cap \phi_e) \\
&= P_{\mathrm{u}}(\Phi_\Gamma \cap \phi_e) + P_{\mathrm{u}}(\Phi_{\mathrm{Y}/\Gamma} \cap \phi_e) \\
&\quad - P_{\mathrm{u}}(\Phi_\Gamma \cap \phi_e \cap \Phi_{\mathrm{Y}/\Gamma}) - P_{\mathrm{u}}(\Phi_\Gamma \cap \phi_e) \\
&= P_{\mathrm{u}}(\Phi_{\mathrm{Y}/\Gamma} \cap \phi_e) - P_{\mathrm{u}}(\Phi_\Gamma \cap \phi_e \cap \Phi_{\mathrm{Y}/\Gamma})
\end{aligned}$$

Applying Baye's Rule
$P_{\mathrm{u}}(\Phi_\Gamma \cap \phi_e \cap \Phi_{\mathrm{Y}/\Gamma}) = P_{\mathrm{u}}(\Phi_\Gamma|(\phi_e \cap \Phi_{\mathrm{Y}/\Gamma}))P_{\mathrm{u}}(\Phi_{\mathrm{Y}/\Gamma} \cap \phi_e)$
it follows that

$$\triangle(e|\Gamma) - \triangle(e|\mathrm{Y}) = P_{\mathrm{u}}(\Phi_{\mathrm{Y}/\Gamma} \cap \phi_e)(1 - P_{\mathrm{u}}(\Phi_\Gamma|\phi_e \cap \Phi_{\mathrm{Y}/\Gamma})).$$

With $P_{\mathrm{u}}(\Phi_{\mathrm{Y}/\Gamma} \cap \phi_e)(1 - P_{\mathrm{u}}(\Phi_\Gamma|\phi_e \cap \Phi_{\mathrm{Y}/\Gamma})) \geq 0$ the equation can be rewritten as

$$\triangle(e|\Gamma) - \triangle(e|\mathrm{Y}) \geq 0.$$

□

Greedy Algorithm

Since, $P_{\mathrm{u}}(\Phi)$ is monotonic sub-modular, the path diversity problem (Prop. 1) can be greedily optimized while maintaining a sub-optimality bound of $(1 - 1/e) \approx 63\%$ [58]. We describe the greedy algorithm in Alg. 5. We start with an empty trajectory set and search

[58] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978; and Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI*, volume 7, pages 1650–1654, 2007

Algorithm 5: Greedy Optimization for a Emergency Maneuver Trajectory Set

Initialize: $\Phi_G = \emptyset$

while $|\Phi_G| < N_{\mathfrak{C}}$ **do**
  $\phi_s = \underset{\phi \in \Phi_F/\Phi_G}{\arg\max}\, P_{\mathrm{u}}(\Phi_G \cup \{\phi\})$
  $\Phi_G = \{\Phi_G \cup \{\phi_s\}\}$
end

through Φ_F to find the trajectory that maximizes P_{u}. This trajectory is saved in Φ_G and in the next step, the search for trajectory that maximizes P_{u} is conducted in Φ_F/Φ_G, and added to Φ_G. The process of greedily selecting trajectories from Φ_F/Φ_G and adding them to Φ_G is repeated till the desired number of trajectories $N_{\mathfrak{C}}$ have been added.

In the next section we explain how to use this greedily generated set to guarantee safety.

Safety Algorithm

We ensure the safety of the mobile autonomous system by using the emergency maneuver library to enforce the constraint that the current and next state of the system always lies in the positive invariant set, which does not intersect the obstacles and stays within the known volume. The algorithm to ensure safety is explained in Alg. 6. Let $\sigma : [0, T] \to \mathcal{X}$ be the nominal trajectory that the vehicle is following to reach the goal. Let, Φ_{S_t} be the emergency maneuver library for state at time t in σ and Δt be the time interval between safety checks.

Algorithm 6: Emergency Maneuver Trajectory Set Application for Reactive Safety

Initialize: $t = 0$
$\Phi_{\text{Previous}} = \Phi_{S_t}$

while *mission active* **do**
  $\Phi_{\text{New}} = \{\emptyset\}$
  for $\forall \phi_c \in \Phi(s_{t+\Delta t})$ **do**
    if $\forall \tau \;\; \mathcal{A}(\phi_c(\tau)) \subseteq (\mathcal{K}_t \setminus \mathcal{O}_t)$ **then**
      $\Phi_{\text{New}} = \{\Phi_{\text{New}}, \{\phi_c\}\}$
    end
  end
  if $\Phi_{\text{New}} = \{\emptyset\}$ **then**
    execute $\phi_e \in \Phi_{\text{Previous}}$
  else
    $\Phi_{\text{Previous}} = \Phi_{\text{New}}$
    follow σ
  end
end

The algorithm queries the emergency maneuver library at a future state of the system, and ensures it can transition to an emergency maneuver which lies in known obstacle free space. If there are no such maneuvers, one of the emergency maneuvers computed at the previous step (for the current state) are executed. Otherwise the vehicle carries on its nominal trajectory. This algorithm has a maximum response time and is guaranteed to keep the vehicle safe. The algorithm is explained with a working example in Fig. 6.

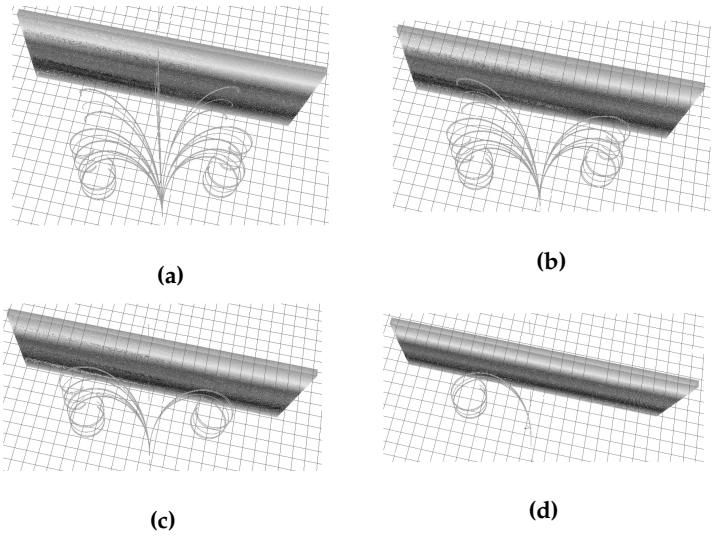

Figure 6: Data from a flight test conducted on 18th December 2013 in Manassas,Virgina. a) Helicopter approaches a large simulated wall with the emergency trajectory libraries wih no emergency maneuver in contact with the wall. b) As the helicopter gets closer to the wall, the emergency maneuvers intersect the wall and become invalid. Only valid maneuvers are displayed. c) More emergency maneuvers are pruned away as they come in contact with the wall d) An emergency maneuver is executed as the future state is no longer safe.

Results

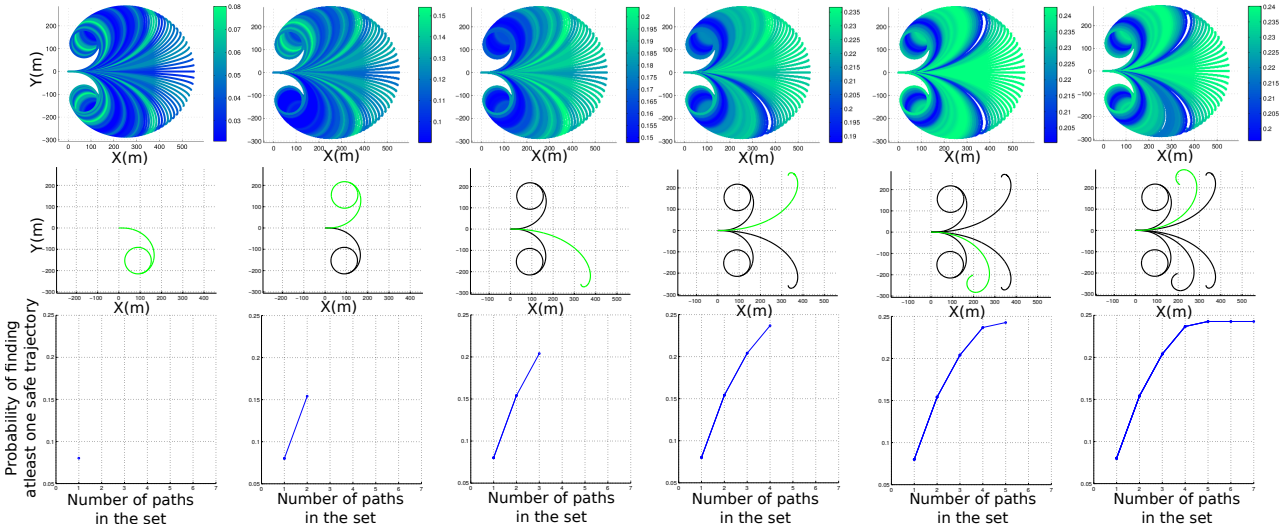

Figure 7: Generation of emergency maneuver library for one state. From left to right the plots step through the generation of emergency maneuver library for 6 iterations. The top row displays the search space from which the current trajectory is picked, where each trajectory is colored according to the probability of not passing through an obstacle in the set. The middle row shows the greedily selected maneuver in the current step in green and existing maneuvers in the set in black. The bottom row shows the total probability of finding at least one maneuver in the set not passing through an obstacle. The robot starts at 25m/s longitudinal velocity for all the maneuvers and for illustration

We generated the emergency maneuver library to ensure the safety of the autonomous Boeing Unmanned Little Bird Helicopter, equipped with a large field of view range sensor. The dynamic constraints of the helicopter are given in Tab. 2.

Given these constraints we approximate Φ_F, by five hundred trajectories each forming a positive control invariant set. The trajectories for this application end in a hover and can trivially be extended to end in a loiter if desired. Each trajectory slows down the helicopter using the maximum allowed deceleration. The trajectories are generated by sampling the roll rate and z acceleration uniformly. Once the

Constraint	Velocity $\|v(t)\|$	
	$\geq 20\,\mathrm{m/s}$	$< 20\,\mathrm{m/s}$
Roll [°]	25.00	28.50
Roll rate [°/s]	15.00	–
Heading rate [°/s]	–	28.50
Longitudinal vel. [m/s]	60.00	20.00
Vertical vel. [m/s]	5.00	5.00
Longitudinal accel. [m/s²]	0.75	0.75
Vertical accel. [m/s²]	1.00	1.00

Table 2: Constraints on trajectory

helicopter has made a 180° coordinated turn the radius of the turn is fixed and the vertical velocity is forced to be $0\,\mathrm{m/s}$. We use a constant resolution three dimensional grid as our representation and assume uniform probability of occupation of each voxel. The probability of a trajectory set containing at least one unoccupied trajectory is calculated using inclusion-exclusion principle as suggested in [59]. Fig. 7 steps through the emergency maneuver library generation process for the robot motion restricted to a plane starting at $25\mathrm{m/s}$ forward longitudinal velocity. The probability of at least one maneuver in the set surviving reduces with each trajectory being added and almost levels off at about 5 trajectories. Given a trajectory set we can calculate the sensor range required for different velocities. Given an emergency maneuver library, the minimum sensor range required for a certain velocity is calculated as

$$\text{range} = \min_{\phi_c}(\max(\xi(\phi_c))). \tag{14}$$

The function ξ returns a vector of the euclidean distances between starting state $\mathbf{x}$ and all the states in $\phi_c \in \Phi_G(\mathbf{x})$.

[59] Michael S Branicky, Ross A Knepper, and James J Kuffner. Path and trajectory diversity: Theory and algorithms. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1359–1364. IEEE, 2008

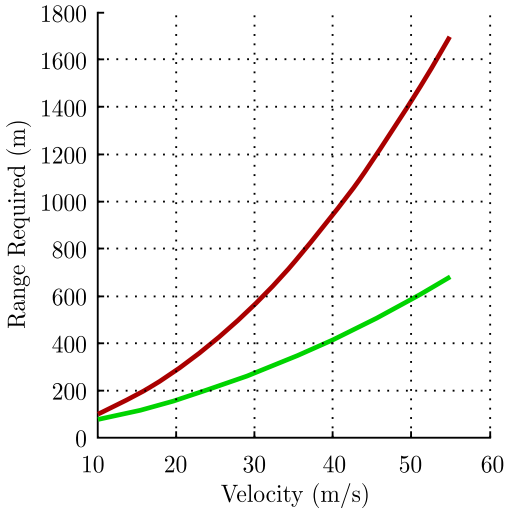

Figure 8: Changes in Sensor Requirements. The sensor range required for safe operation of the vehicle when using stopping distance for safety is displayed in red, in green is the sensor range required for safe operation of the vehicle when using emergency maneuver library for helicopter safety.

The best case sensor range required while using the emergency maneuver library is given by (14). The worst case is the same as the stopping distance. Hence, the emergency maneuver library is guaranteed to provide at least as much performance as using only the stopping distance for the safety evaluation. In Fig. 8 the different requirements on the sensor range for stopping distance and the emergency maneuver library are illustrated. We can quantify the performance of an emergency maneuver trajectory by calculating the maximum safe velocity it allows the helicopter during a mission and the planning time it allows the planner before it becomes imperative for the helicopter to execute the emergency maneuver library. Fig. 9 shows the maximum safe velocity and allowed planning times for a flight test conducted in Quantico, Virginia. The red line shows the path where the helicopter is turning towards the landing zone. The orange part of the path corresponds to the part of the mission for which the sensor on the helicopter focuses on the landing zone for its evaluation. This implies, when the helicopter is moving through the path in orange the sensor stops looking for obstacles and the helicopter comes increasingly close to the known/unknown volume boundary, leading to a drop in maximum safe velocity and allowed planning time. The red part of the path corresponds to turns, it should be noted how the maximum safe velocity according to the stopping distance decreases as the vehicle turns. This happens due to a reduction in effective range of the sensor because of the sparsity of observations in front of the vehicle while turning. The maximum safe speed by the emergency maneuver library is unaffected, as it efficiently utilizes the known space.

Fig. 11 shows the maximum safe velocity and allowed planning times for seven flight tests conducted in Quantico, Virginia which are shown in Fig. 10. As can be seen in Fig. 11, the maximum safe speed is always greater than the helicopter speed, which means the helicopter is always safe. Furthermore, the stopping distance based safe velocity limit is always considerably below the executed velocity which shows that the emergency maneuver library approach is less conservative than the stopping distance approach. The use of the emergency maneuver library also allows higher available planning times allowing for a better overall performance of the motion planning approach due to longer available computation time.

Conclusions

The main contribution of this paper is the development and evaluation of emergency maneuver libraries to guarantee safety of high speed mobile autonomous systems online in unknown environments.

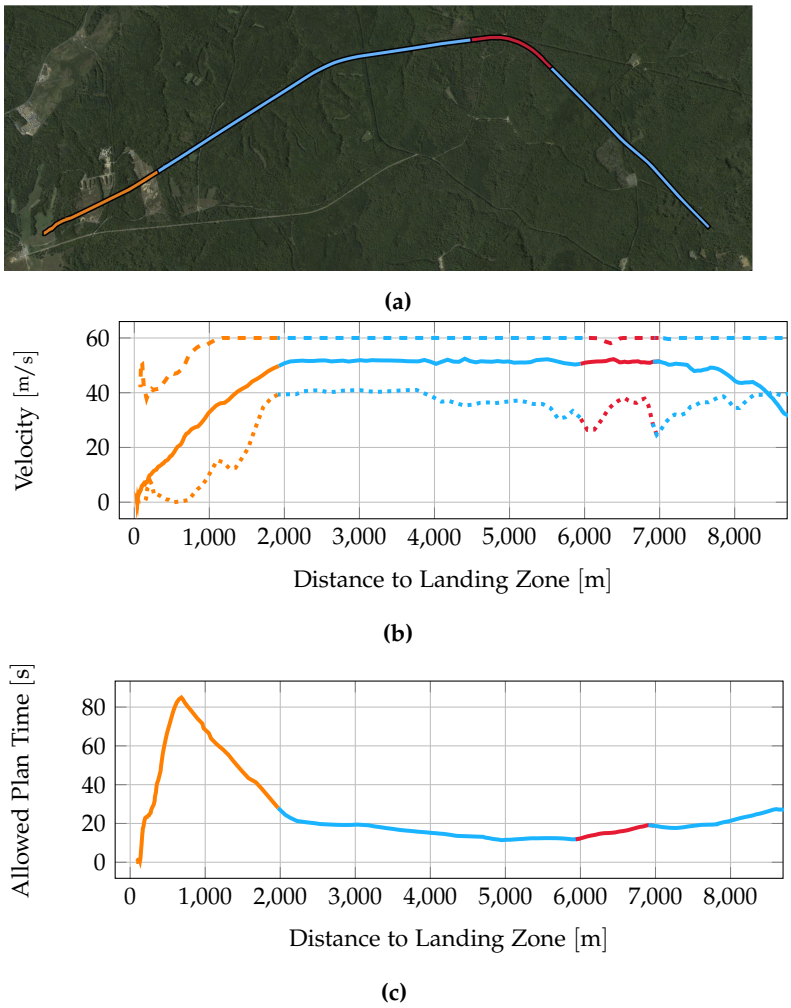

Figure 9: Safety Quantization: Flight test in Quantico, Virginia. a) Shows an autonomous landing mission conducted in Quantico, Virginia on Unmanned Little Bird. b) Shows the safe velocity of the helicopter with the emergency maneuver library during the flight tests in dashed line, the executed velocity in solid line and the safe velocity if stopping distance is used in dotted line. c) This figure shows planning time available to the planner, before the vehicle will reach the edge of known space and execute one of the emergency maneuvers. The planning time calculated assuming the helicopter will follow the current planned trajectory.

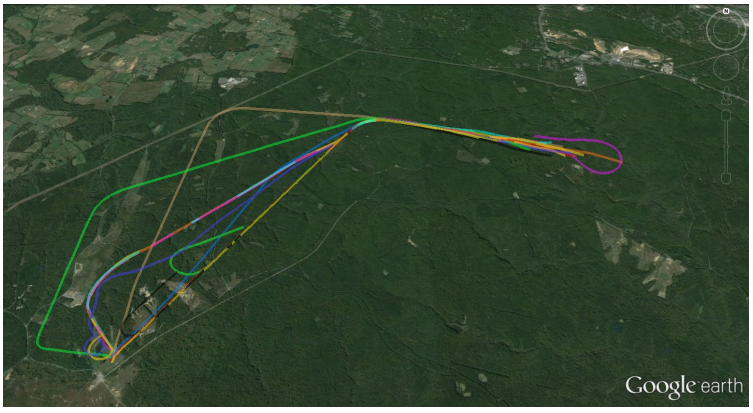

Figure 10: Paths taken by the autonomous Unmanned Little Bird during landing and wave-off missions conducted in Quantico, on 26 Feb. 2014.

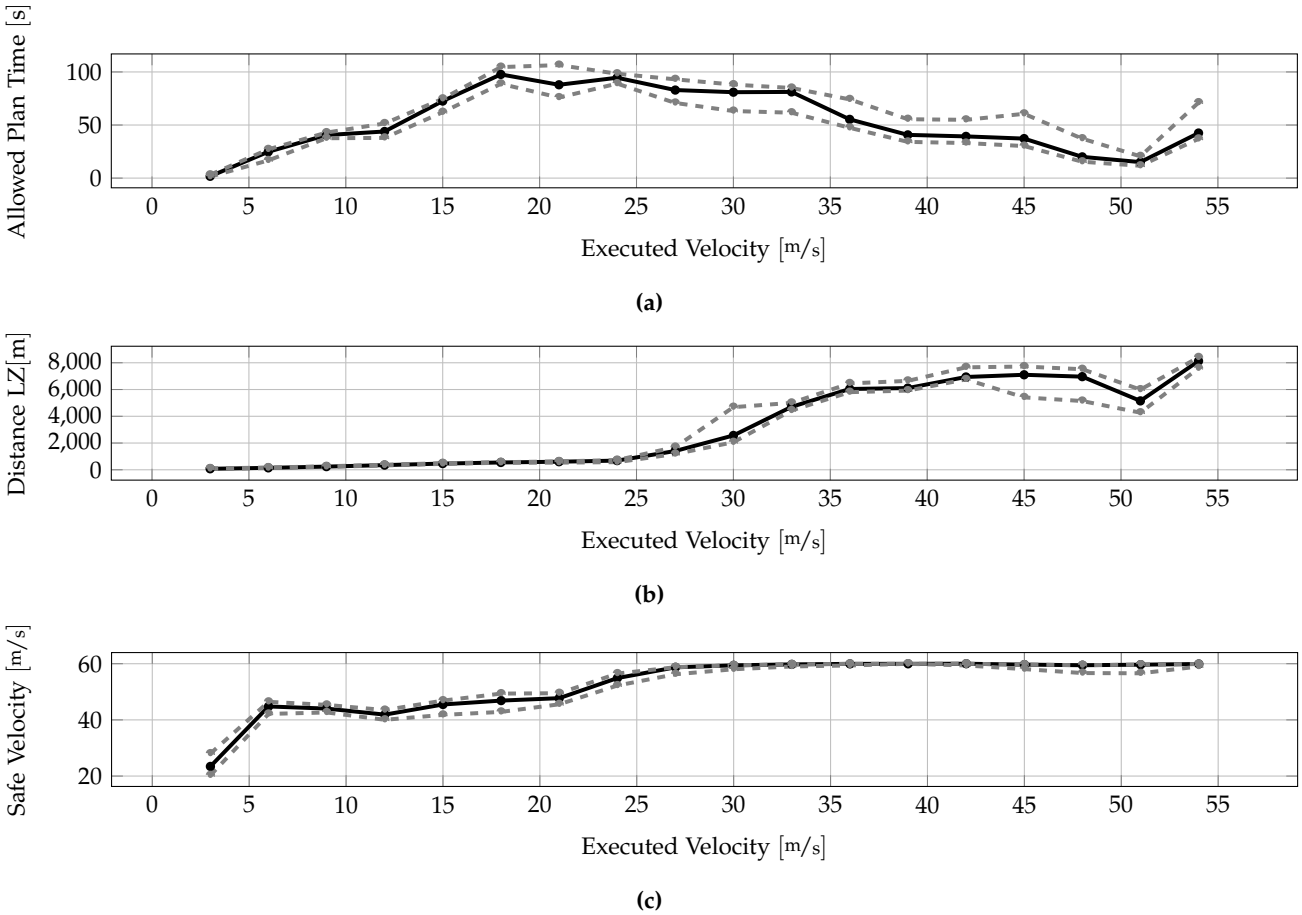

Figure 11: The figures show the allowed plan time, distance to the landing zone (LZ) and safe velocity relative to the executed velocity of the helicopter. The black line shows the mean and the gray dashed line illustrates the upper and the lower bound of the measurements of all considered flight tests.

The algorithm determines the maximum velocity for which safety can be ensured, given a future path of the robot. Therefore, it takes into account the constraints of the perception system as well as the dynamics of the rotorcraft. An off-line optimized set of control invariant trajectories (the emergency maneuver library) represents the core part of the presented approach. As a result, the approach is independent of the planner guiding the rotorcraft to the goal and the dynamics of the rotorcraft. Furthermore, the off-line generated trajectories enable real-time evaluation of the safety of the rotorcraft.

The experimental evaluation of the proposed approach shows, that it is always less conservative than the stopping distance based safety evaluation. Thus, the helicopter is able to operates at its performance limits while the presented maneuver based approach still ensures safety at all time.

In the future, we want to further decrease the conservativeness of the safety evaluation by considering a closer coupling between the emergency maneuvers and the motion planner. Therefore, the

planner pro-actively responds to potentially unsafe situations rather than just react to them. Another focus of our current research is to consider wind disturbances and obstacle distributions to further improve the robustness of the emergency maneuver library.

Sensor Planning

This section explains the sensor planning problem and why its important to solve it.

Problem Setup

Mobile robots are designed to navigate from start to end state while minimizing the cost of traversal. Let the robot's state space be $\mathcal{X} \subset \mathbb{R}^n$. Let $\sigma : [0, T] \to \mathcal{X}$ be the state space trajectory and $C : \mathcal{X} \to \mathbb{R}$ be the cost function, where T is the time horizon. The boundary values are $\sigma(0) = \sigma_0$ and $\sigma(T) = \sigma_f$. Let the dynamics constraints on the robot be given by $h(\sigma(t), \dot{\sigma}(t), \ddot{\sigma}(t)) \leq 0$. The cost of the trajectory in a fully deterministic environment is $\int_0^T C(\sigma(t))\,\mathrm{dt}$.

Operating in partially known, unstructured environments the robot has to decide on its next action based on its current belief. Let the high dimensional state space of the world be $\mathcal{W} \subset \mathbb{R}^m$, where world includes the uncertainty of the robot about its environment and it's pose. Let, belief be a probability distribution over the state of the world, $b : \mathcal{W} \to \{0, 1\}$. Let's assume the belief of the robot at the start of the mission is b_0. The belief of the robot changes as observations are made using a sensor. Let the sensor's state space be $\mathcal{S} \subset \mathbb{R}^s$. Let $\sigma_s : [0, T] \to \mathcal{S}$ be the sensor trajectory.

The cost function changes as the belief of the robot evolves. To highlight this fact we represent the cost functional as $C_b : \mathcal{X} \times \mathcal{W} \to \mathbb{R}$. Due to the stochasticity of the belief of the robot, it can only reason about expected cost of its policy, $\mathbb{E}_{p(b|\sigma,\sigma_s,b_o)}\, C_b(.)$. The distribution of belief trajectories $p(b|\sigma, \sigma_s, b_o)$ at any given time is dependent on both state and sensor trajectory. Let the dynamics constraint on the motion of the sensor be given by $h_s(\sigma_s(t), \dot{\sigma}_s(t), \ddot{\sigma}_s(t)) \leq 0$. The full optimization problem can then be defined as follows.

$$\underset{\sigma(t),\sigma_s(t)}{\arg\min} \underset{p(b|\sigma,\sigma_s,b_o)}{\mathbb{E}} \int_0^T C_b(\sigma(t), \sigma_s(t), b(t))\,\mathrm{dt} \tag{15}$$

$$h(\sigma(t), \dot{\sigma}(t), \ddot{\sigma}(t)) \leq 0$$
$$h_s(\sigma_s(t), \dot{\sigma_s}(t), \ddot{\sigma_s}(t)) \leq 0$$

In this paper we study the problem of optimizing the sensor trajectory given a fixed vehicle trajectory $\sigma(.)$. The problem is then reduced to.

$$\arg\min_{\sigma_s(t)} \mathop{\mathbb{E}}_{p(b|\sigma,\sigma_s,b_o)} \int_0^T C_b(\sigma(t),\sigma_s(t),b(t))\,\mathrm{dt} \tag{16}$$

The constraints on the sensor actuation specified in eq. 15 also apply to eq. 16. Notice that, in this formulation the sensor motion can only result in gaining of information about the environment. Therefore, the minimization of the cost function with the $\sigma_s(t)$ as the only variable, results in sensor gaining information that is important to minimize the required cost function. We call this information contextually important information, as it is important to sense this information to reduce the cost function. Let $\mathbb{IG}(x,\sigma(t),\sigma_s(t),b(t))$ be the information gain at a state x and time t. Let $M(x,\sigma(t),b(t)) \in \{0,1\}$ return 1 if the information is contextually important and 0 otherwise. The optimization problem then reduces to eq: 17. Proof in [60].

[60]

$$\arg\max_{\sigma_s(t)} \mathop{\mathbb{E}}_{p(b|\sigma,\sigma_s,b_o)} \int_0^T \int_{\forall x\in\mathbb{X}} M(x,\sigma(t),b(t)) \mathbb{IG}(x,\sigma(t),\sigma_s(t),b(t))\mathrm{dxdt} \tag{17}$$

In our application,the robot is a full scale autonomous helicopter, the sensor is a nodding lidar, with an actively controlled pitch axis. The trajectory $\sigma_s(t)$ of the nodding sensor relative to the helicopter can be completely defined by its nodding angle profile in time $\rho(t)$, laser configuration ρ_{conf}, which consists laser's fast axis resolution ρ_{frr} and its pulse rate, ρ_{prr}. The pose of the robot is given by a high accuracy GPS/INS system, the world representation is an occupancy grid map [61]. The information gain evaluation for an occupancy grid map is presented in detail in [62]. $M(x,\sigma(t),b(t))$, the contextual importance function is discussed in detail in section . In the next section we present an overview of our approach to solve eq. 17 for a mobile autonomous robot.

[61] Sebastian Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2):111–127, 2003

[62]

Approach Overview

Information gain based path planning is an NP-hard problem [63]. Moreover, the calculation of information gain itself is computationally expensive. In this work, we use an occupancy grid map based representation. For a laser sensory action with R rays, interacting with N cells of an occupancy grid map, the computation complexity of calculating information gain is $O(3^R N)$ [64]. As a result, the computational cost involved in calculating information gain for a large

[63] Amarjeet Singh, Andreas Krause, Carlos Guestrin, and William J Kaiser. Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research*, pages 707–755, 2009

[64]

number of rays over a large range, makes it non-trivial to conduct gradient based optimizations.

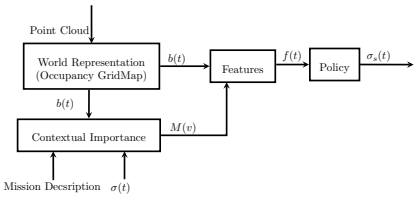

Figure 12: Approach Overview: The sensor is actively controlled through a policy that takes in the features that are extracted from robot's belief. The features take expected information gain and contextual importance function into account.

To overcome the computational complexity issues and solve the problem online, we propose learning a one step policy function that maps from features to action, (see Fig. 12). The policy learnt, is designed to maximize the information gain in the next time step. For the application considered in the paper, the policy learnt keeps the robot safe at high speeds and enables quick landing zone evaluation, allowing for low mission times. The data flow in the block diagram (Fig. 12) is explained in the rest of this section to provide an intuitive understanding of the suggested approach and its application.

The point cloud generated by the laser is used by the perception representation to form the robot's belief about the world, in our case an occupancy grid map is used as the world representation. The belief of the robot, along with the mission description and robot trajectory are used to infer the contextual importance function. The contextual importance function presents the locations from which it is important for the robot to gain information to complete its task safely. The construction of this function is further described in section . The contextual importance function along with the belief of the robot are used to calculate features, that are used by the policy to decide which action to take. The features are the only input to the policy. Therefore, they should capture the saliencies of the interaction between sensory actions, the representation and the environment. The construction of one such set of features is described in section .

The policy takes the features as input and provides the trajectory for the sensor, $\sigma_s(.)$. For the application considered in the paper, the policy is designed for high range laser, nodding in pitch axis with respect to the vehicle. The construction of the policy is described in section . The parameters of the constructed policy function need to be learnt to maximize the contextually important information gain. The large partially known state space in which the robot operates makes the learning of the correct policy parameters intractable. We reduce the scenarios for which the policy is to be learnt to overcome this problem. The procedure to learn the policy parameters is presented in section . In the next section we describe the construction of contextual importance function for our application.

Contextual Importance Function

Finding the contextual importance function is as hard as solving for the original optimization problem [65]. We approximate this function by defining $M(x, \sigma(t), b(t)) = 1$, for all x, that might be important to the vehicle given the belief $b(t)$ and the robot state σ. In this work, we consider the structure of the contextual importance function to have two components - safety $M_{safe}(.)$ and landing zone evaluation $M_{lz}(.)$.

[65]

Safe Navigation

Ensuring vehicle safety while navigation requires that the vehicle can transition to a trajectory that stays in the known obstacle free volume forever at any given time. We use the emergency maneuver library [66] to enforce the safety constraint. It suggests using an offline computed emergency maneuver library to ensure vehicle safety. If there exists a trajectory in the emergency maneuver library that starts from the current state of the vehicle and stays within the known obstacle free region for infinite time, the vehicle can be considered safe. Therefore, it is important to sense the volume occupied by emergency maneuver trajectories corresponding to the planned trajectory.

[66] Sankalp Arora, Sanjiban Choudhury, Sebastian Scherer, and Daniel Althoff. A principled approach to enable safe and high performance maneuvers for autonomous rotorcraft. In *American Helicopter Society 70th Annual Forum*. AHS, 2014

Given a vehicle state $\sigma(t)$ at time t, let there be a function $\kappa(x, \sigma(t)) \in \{0, 1\}$ which assigns the points inside the volume occupied by emergency maneuver library for $\sigma(t)$ if the state is not safe and returns 0 if the state is safe. The the contextual importance function for safety is then given as.

$$M_{safe}(x, \sigma(t), b(t)) = \kappa(x, \sigma(t)) \tag{18}$$

Landing Zone

The mission objective of the helicopter is to navigate safely from start to goal and then land. Therefore, it is important to evaluate landing zones before the helicopter commits to a landing. The landing zone is represented as a two and a half dimensional grid, with each cell storing the probability of that cell being a safe landing site. Let, the landing zone grid is represented by $\mathbb{LZ}_{\mathbb{G}}$. So each cell inside the landing zone is allocated contextual importance of one.

$$M_{lz}(x, \sigma(t), b(t)) = 1 \; \forall x \in \mathbb{LZ}_{\mathbb{G}} \tag{19}$$

The combined contextual importance function of safety and landing zone evaluation is given as

$$M(.) = \max(M_{safe}(.), M_{lz}(.)) \tag{20}$$

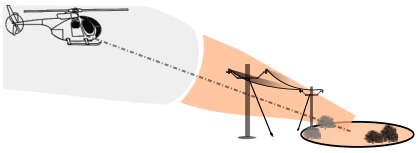

Figure 13: Contextual Importance Function – The figure illustrates the region with contextual importance function greater than one in orange, and regions which were contextually important in grey. As the robot navigates, the regions known in the past are now irrelevant, as there is no more information to be gained that might affect its future actions. The volume around the trajectory, bounded by the emergency maneuver library for the future as yet unsafe states and volume inside the landing zone is contextually important.

Fig. 13, illustrates the contextual importance function for a helicopter navigating towards a landing zone to land.

Policy Parametrization

We have discussed that the problem of directly optimizing the cost function itself is NP hard and the cost function evaluation is computationally expensive, making local optimization intractable. Therefore, we need a mapping from features to actions that can be used online.

We learn a policy offline that maximizes the information gain in the worst case scenario. The worst case scenario is presented in section . The action space is reduced to constant velocity nodding motion. We want to optimize the use of sensor bandwidth and avoid wasting sensor's time looking at regions that are already known, inaccessible or unimportant to the mission. Therefore the features have to cover visibility, information gain and contextual importance function. In the next section we cover the feature design and then present the policy function.

Features

To calculate the relevant field of view for the sensor, we calculate contextually important expected information gain along a grid of pitch and yaw directions, with rays originating from laser's center, creating a 2.5D map, ζ_m of expected contextually weighed information gain. We generate this map by tracing rays through 3D occupancy grid representation and calculating contextually weighed expected information gain along the rays. The algorithm for calculating the expected information gain along the rays is presented in [67]. Average range at which information is gained r_e is also used as a feature for the policy. r_e can be calculated while calculating ζ_m. The algorithm to calculate ζ_m and r_e are presented in [68].

The policy function can then be represented as eq. 21 below.

$$[\dot{\rho}, \rho_{configuration}, \rho_{max}, \rho_{min}] = \pi(\zeta_m, r_e) \tag{21}$$

The FOV extent(ρ_{max}, ρ_{min}) is inferred using ζ_m. The ρ_{max} is given by maximum pitch in ζ_m for which the information gain is greater than 0. The ρ_{min} is given by minimum pitch in ζ_m for which the

[67] Brian J. Julian, Sertac Karaman, and Daniela Rus. On mutual information-based control of range sensing robots for mapping applications. *The International Journal of Robotics Research*, 33(10):1375–1392, 2014. DOI: 10.1177/0278364914526288

[68]

information gain is greater than 0. The velocity of the nod is decided by r_e at which the information to be sensed. We now look at how we define the mapping between range and velocity that maximizes the information gain at that range.

Policy Search

We want to find a nodding speed at which to scan the volume to maximize the gain of contextually important information. The standard method to find the optimal policy is to run the system either in simulation or for real and search the parameter space for values that minimize the average cost over a range of scenarios. But given we are designing a policy to keep the vehicle safe in all conditions, we search for the optimal policy for the worst case scenario. In the worst case scenario, the time available to the sensor for scanning is minimal and the volume to be covered is maximum. The following sections develop the worst case scenario for sensor planning for an autonomous mobile robot navigating through unknown environments and present how sensing actions can be efficiently evaluated, enabling search of policy parameters in reasonable times.

Policy Search Scenario

We present the scenario in which the scanner has minimum time to scan the largest region to keep the vehicle while flying towards its goal. In the worst case scenario, the vehicle can be assumed to be safe only till the time it takes to update the representation. For our system the maximum time the representation takes to update is when the occupancy grid mapping and the distance transform grids are moved for scrolling due to vehicle movement. Let this time be given by t_r seconds. Since the volume covered by the emergency maneuvers is monotonic with velocity, we assume that vehicle is moving at maximum safe velocity that is allowed for the range at which the information is queried.

Let us assume, the information is requested at range r. The vehicle is moving at speed v, the maximum update time of the world representation (an occupancy gridmap in our case) be t_r seconds, the time to process the point cloud collected over t_r seconds be t_{pcl}. The minimum range required to fly safely at velocity v is given by $vt_{lookahead} + r_v$, where r_v is the minimum range in which an emergency maneuver fits for velocity v. At this range the vehicle is committed towards executing an emergency maneuver. Now, given the maximum refresh time of t_r, the minimum range required to guaranty that the vehicle at least has a chance to continue on the pre-planned trajectory

is given by $R_1 = v[t_{lookahead} + t_r + t_{pcl}] + r_v$. If enough volume hasn't been cleared after $t_r + t_{pcl}$ the vehicle will execute an emergency maneuver. So the sensor has t_r seconds to detect the free unoccupied space available, at east till the range of $R_2 = R_1 + v(t_r + t_{pcl})$ to ensure safety.

Given the environment in which an autonomous system operates in, one needs to reason about the smallest obstacle against which the system guarantees safety. For example, an autonomous helicopter should guarantee provide limits that it will at least detect wires and steer clear of them. So, the sensory system has to ensure that it will maximize the probability of detection of such an object if it exists and at the same time minimizing false positives. In the following section we show how this intuition develops into maximizing the information gain in the relevant volume. And we then present an efficient method to evaluate the information gain for the case of ensuring safety.

Information gain as probability of detection of worst case obstacle

Let us assume, the smallest obstacle against which the autonomous system has to guarantee safety is given by o, the probability that an object is present at x is given by $p_x(o)$. For brevity we use $p(o)$ instead of $p_x(o)$. Let the probability of detection of obstacle given there is exists an obstacle at x be given by $p(d|o)$ and probability of detecting an obstacle given there is no obstacle at x is given by $p(d|o')$. We express the the expected information gain given $p_x(o)$ as a function of $p(d|o)$ and $p(d|o')$

.

$$\begin{aligned}\mathbb{E}\,\mathbb{IG}(o) = p(o)(p(d|o)(H(o) - H(o|d))+ \\ p(d'|o)(H(o) - H(o|d'))) + \\ p(o')(p(d|o')(H(o) - H(o|d))+ \\ p(d'|o')(H(o) - H(o|d')))\end{aligned} \tag{22}$$

Assuming there are no false positives, $p(d|o') = 0, p(d'|o') = 1$. For brevity we change the notation to $p(d|o) = p_a, p(d'|o) = p'_a$The equation reduces to the following.

$$\begin{aligned}\mathbb{E}\,\mathbb{IG}(o) = H(o) - p(o)p_a H(o|d) - \\ p(o)p'_a H(o|d') - p(o')H(o|d')\end{aligned} \tag{23}$$

Gven $p(d|o') = 0, p(d'|o') = 1$, $p(o|d) = 1$, can be easily verified using Baye's Rule. Since $p(o|d) = 1$, $H(o|d) = 0$.

$$\mathbb{E}\,\mathbb{IG}(o) = H(o) - (1 - p(o)p_a)H(o|d') \tag{24}$$

where,

$$H(o|d') = \sum_{p(o|d')\in[0,1]} p(o|d')log(p(o|d')) \tag{25}$$

$p(o|d')$ can be expressed as a function of p_a and $p(o)$, and is given by following equation.

$$p(o|d') = (1 - p_a)p(o)/(1 - p_a p(o)) \tag{26}$$

Using equation 24,25 and 26 the expected information gain can be completely expressed as a function of $p(d|o), p(o)$. For a given p(o), the information gain monotonically increases with $p(d|o)$. Fig. 14, illustrates this point. To find the effectiveness of a policy we

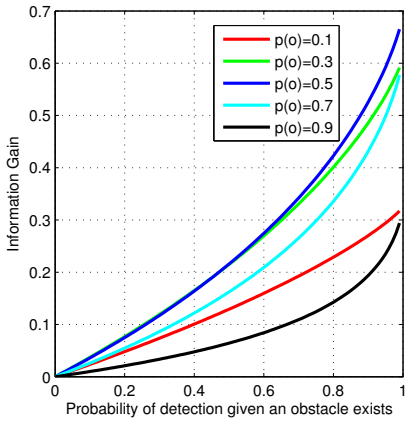

Figure 14: Information Gain – The expected information gain given a prior ($p(o)$) is monotonic in probability of detecting the event w if the event occurred, $p(d|o)$. The assumption being, there are no false positives, $p(d|o') = 0$. This implies for maximizing information gain, one may maximize $p(d|o)$

evaluate $p(d|o)$ for different configurations of the obstacle and calculate the information gain in the mission relevant region. The action that provide with maximum information gain is selected as policy for given feature inputs. In the next section we present an efficient algorithm to calculate $p(d|o)$.

Efficient calculation of probability of detection

We describe a method that exploits the symmetry and the structure of the problem to calculate information gain for a range sensor efficiently. The naive method of calculating $p(d|o)$ is through sampling configurations of the object from $p(o)$ and computing ray intersections with the object given the sensor nodding motion. This algorithm has the complexity of $O(RM)$, where R is the total number of rays generated by the laser and M is the number of samples drawn from $p(o)$. Let's assume an objects exists in a $6D$ space, and

for each dimension we sample N particles. The computational complexity is then $O(RN^6)$. We introduce an algorithm that reduces this computational complexity to $O(R^3N^4)$ by reasoning about ray-object intersections in the object's cspace in spherical coordinates. We further exploit the symmetry and structure of the problem of sensing for safety to reduce the complexity to $O(R_l^3N)$, where R_l is a fraction of the actual number of rays. The algorithm for fast calculation $P(d|o)$ is presented in Alg. 7. The input to the algorithm is sensor trajectory, σ_s whose information gain is to be calculated, an object's partial pose $x_{partial}$, only stating the distance of the object from the sensor and its relative orientation. We assume that the size of the object is small as compared to the query range, which results in negligible change in projection of the object to $[\theta, \phi]$ plane of the spherical coordinate frame, centered at the laser. The algorithm returns a function $p(d|\theta, \phi, x_{partial})$ that provides the probability of detection of object if its in $x_{partial}$ configuration at any $[\theta, \phi]$. Lets, assume the object is detected if n points hits are detected on the object. Each laser beam is represented as polygon in $[\theta, \phi]$ space of spherical coordinates It is assumed that if the ray and the object intersect, the hit will be reported with a probability of p_h.

The $Intersect(.)$ function (Alg. 8), calculates the intersection of set of unique polygons amongst themselves and stores it as Y_{int} and also returns the list of members of its input set that intersected to form a polygon $\forall i \in Y_{int}$ as $\lambda^i_{int} \in \Lambda_{int}$. The output of Alg. 8 are used by $Probability(.)$ function (Alg. 9), to calculate the probability of detection given an object at $x_{partial}$ $\forall [\theta, \phi]$. It can be trivially proven

Algorithm 7: $EfficientCalculationofP(d|o)$

Input : $\sigma_s, x_{partial}$

Output: $p(d|\theta, \phi, x_{partial})$

$S_o \leftarrow ProjSpherical(s_{partial})$
$S_L \leftarrow GeneratePolygons(\sigma_s)$
$S_{L,o} = S_L \bigoplus S_o$
// The minkowski sum projects polygons in
// S_L to the space of S_o.
$[Y_{int}, \Lambda_{int}] \leftarrow Intersect(S_{L,o})$
$p(d|\theta, \phi, s_{partial}) \leftarrow DetectionProbability(Y_{int}, \Lambda_{int})$
Return : $p(d|\theta, \phi, s_{partial})$

that given a fixed angular motion profile for a range sensor the number of rays intersecting an object decreases with increasing distance. This implies that the probability of detection of the object decreases

Algorithm 8: *Intersect*(.)

Input : $S_{L,o}$

Output : $Y_{int}^{s_{partial}}, \Lambda_{int}$

Initialize : $Y_{int}^{s_{partial}} = S_{L,o}, \lambda_{int}^{i} = i \forall i \in Y_{int}^{s_{partial}}, Y_{prev} = S_{L,o}$

while $|Y_{prev}| > 1$ **do**
  $Y_{new} =$
  for $j \in Y_{prev}$ **do**
    for $k \in Y_{prev} - [1 : j]$ **do**
      $Y_{new} = Y_{new} \cup (j \cap k)$
      $\lambda_{int}^{end+1} = \lambda_{int}^{j} \cup \lambda_{int}^{k}$
    end
  end
  $Y_{int}^{s_{partial}} = Remove(Y_{int}^{s_{partial}}, Y_{new})$
  $Y_{int}^{s_{partial}} = Y_{int}^{s_{partial}} \cup Y_{new}$
  $Y_{pref} = Y_{new}$
end
Return : Y_{int}, Λ_{int}

Algorithm 9: *Probability*(.)

Input : $Y_{int}^{s_{partial}}, \Lambda_{int}^{s_{partial}}, \phi, \theta$

Output : $p(d|\phi, \theta, s_{partial})$

$i \leftarrow ReturnPolygonContaining(\theta, \phi, Y_{int}^{s_{partial}})$
$p(d|\phi, \theta, s_{partial}) = \sum_{k=n}^{|\lambda_{int}^{i}|} \binom{|\lambda_{int}^{i}|}{k} p_h^k (1 - p_h)^{|\lambda_{int}^{i}| - k}$
Return : $p(d|\phi, \theta, s_{partial})$

with increasing distance [69]. We maximize the information gain at the range at which the information is requested, maximizing the minimum information gain. This results on maximizing the information gain on a 2d spherical manifold $[\theta, \phi]$. So the configuration space of the object restricted to a 2D plane is x, y, θ. Assuming, we use N particles for each dimension to approximate $p(o)$, the computational cost of evaluating an action is given by $O(R^3N)$. We restricted the action space to constant velocity nods,hence the laser points lie at a constant distance form each other in pitch and yaw space, forming a grid. This symmetry can be exploited by finding $p(d|o)$ for every configuration in a single grid cell. Also, the intersection of the object need only be checked with the points in l^2 radius of the grid cell. Let these number of points in l^2 radius be R_l. The complexity of evaluating an action is $O(R_l^3N)$. For further details about reduction in computational complexity please refer to [70]. The action providing maximum information gain is stored as the nodding velocity for that range. Fig 15 presents plots for expected probability of detection of a wire, given a uniform distribution. This gives a more intuitive evaluation of the nodding trajectories.

[69]

[70]

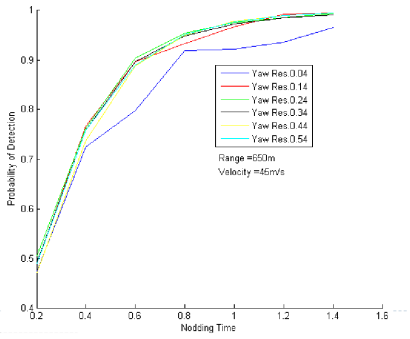

Figure 15: Expected Probability of Detection: The expected probability of detection of a wire if it exists, $\mathbb{E}_{p(o)}\, p(d|o)$ for a given action is an intuitive indicator of how good an action is. The plot shows $\mathbb{E}_{p(o)}\, p(d|o)$ versus varying nodding time period for the worst case scenario for the sensor at vehicle velocity, $v = 45\ m/s$. Each sensor velocity corresponding to different nodding time periods is evaluated till $t_r = 1.4 seconds$. The evaluation shows scans with slower scanning speed are better, this is also intuitively the correct behavior as slower nodding speeds means more uniform point distribution in the $[\theta, \phi]$ manifold.

The policy for scanning for landing zone evaluation may also be calculated by the method presented above, but for our application continuously scanning the region of landing zone to get a uniform point distribution on it sufficed as an adequate policy. The sampling rate of the laser is fixed for safety scans to 29000 points per second, as changing the sampling rate means switching off the laser for 7 seconds and reducing the maximum range of the laser, which rendered the vehicle unsafe. Once the safety of the entire mission is ensured, the sampling rate of the sensor is increased for faster landing zone evaluation.

Results

We test our approach on an autonomous full-sized helicopter which uses the Near Earth Autonomy, Mark 3 active nodding range sensor. We tested our approach in Mesa, AZ, Manassas, VA, Pittsburgh, PA and Quantico, VA for more than 30 autonomous missions in varying conditions. Fig. 16 shows a typical collection of missions. All the missions need the vehicle to navigate from start to the landing zone at high speeds, evaluate the landing zone and land without hovering if the landing zone is safe. The sensor had to clear enough region to keep the vehicle safe, even at the speeds of 56 m/s and sample a 100m diameter landing zone with enough number of points to evaluate it for landing the vehicle on it. On an average the time available to the laser to sample the landing zone (LZ) was below 20seconds. Such missions were impossible to conduct with a passive scanning approach. Passive scans designed to ensure safety and evaluate landing zone are only able to keep the vehicle safe up to a maximum speed of 18 m/s and evaluate the complete landing zone in 112 seconds. Fig. 17 shows a detailed analysis of the sensor

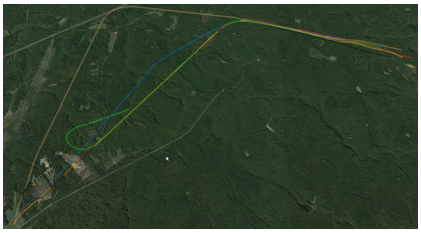

Figure 16: Mission Definition: The helicopter navigates from loiter point to landing zone in less than 210 seconds. It has to navigate the environment while being provably safe and touch down at the LZ without hovering over it to look for potential sites.

angles during a mission in Mesa, AZ. The policy scans for safety in the beginning when the vehicle is navigating towards its destination. The focus of the sensor is moved to LZ evaluation once safety is ensured till the end of the mission. Notice the laser configuration is changed only after confirming the safety of the vehicle for the rest of the mission. This mitigates the problem of reduced sensor range at high sampling rates affecting vehicle safety. Fig. 18 shows the

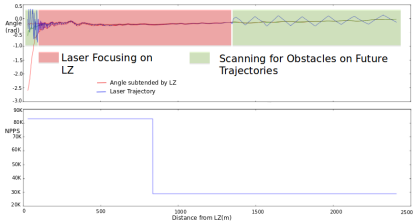

Figure 17: Sensor Angles and Configuration: The top figure shows the sensor angles. In the initial part of the mission, the sensor plans to keeps the vehicle safe (oscillation of the blue line). It switches to focussing on the landing zone when safety for the remainder of the mission has been guaranteed. This focussing of the laser is shown by the narrow peak to peak of the nodding angles. At the very end, once enough points on the landing zone has been focussed, the sensor reverts back to ensuring that the vehicle can be safe should it desire to waveoff. The bottom figure shows the configuration switch. Since switching configuration reduces

performance of the policy in keeping the vehicle safe. The sensor always keeps the vehicle safe and only switches to scanning the

landing zone when it has guaranteed safety for the entire trajectory leading to touchdown. Notice that the executed vehicle speed is always well below the safe speed of the vehicle.

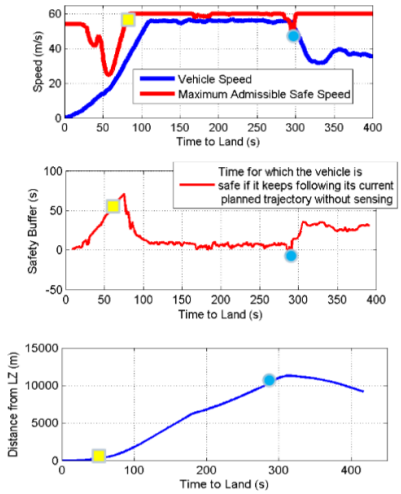

Figure 18: Safety analysis for a single mission. In each figure, two events are marked - first (blue) is when a manual pilot makes an aggressive maneuver, second (yellow) is when the sensor switches to scanning the landing zone. The figure in the top shows that in autonomous mode the vehicle is flying slower than the safe speed limit. The middle figure shows the time after which the vehicle will be unsafe given its current belief. As the vehicle approaches to land, this time increases till its high enough for the sensor to switch to scanning the landing zone. The bottom figure shows the progress of the mission.

Fig. 19 shows performance of the system over 12 missions executed in Quantico, VA. The vehicle is always safe and evaluates the landing zone in time for all the missions, this demonstrates the reliability of the policy in the real world.

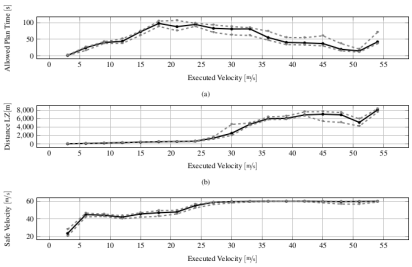

Figure 19: Multiple Mission Safety Analysis: The safety analysis shown over several missions. The worst case performance of the system is still shown to be guaranteed safe.

Conclusion

The main contribution of this paper is to define an algorithm for actively controlling the sensor mounted on a vehicle to maximize the vehicle's performance. We reduce the problem of actively controlling the sensor to minimize trajectory cost to the problem maximizing contextually important information gain. We suggest a policy based approach to solve the NP hard problem of maximizing contextually important information gain. The method is evaluated through application on a full scale autonomous helicopter. We have shown clear benefits of actively controlling the sensor using our method as opposed to passive nodding. The method enabled the helicopter to

perform autonomous high speed, guaranteed safe flights and land on landing sites without the need to hover.

In the future, we want to explore the possibility of inferring the contextual importance function, given a cost function. We are currently woking towards applying the method on micro aerial vehicles and extending the policy search space to allow for richer sensory trajectories.

Related Work

Recently many research groups have ventured into the vision only obstacle detection and avoidance. Most approaches generate point clouds from disparity images and fuse with point clouds from other sensors such as lasers. Appropriate fusion of multimodal sensory data is still a work of active research. Most prevalent approach has been to generate evidence grids or occupancy grids to determine occupancy and do collision checks [71], [72]. Working with 3D gridmaps is both memory intensive for large occupancy maps and require more computation for registration of data and book keeping when scrolling or moving the grid along with the robot. Trade-off between high resolution gridmap vs grid size is another reason why occupancy grids usually cannot be used to map a large volume with higher accuracy. OctoMaps [73] have recently become popular due to their efficient structure for occupancy mapping. However, due to excess noise in stereo sensor generated data at long ranges, often a smaller map is maintained and full stereo sensor data is not used.

Gohl et. al [74] propose to use a spherical coordinate based gridmap for stereo sensors but it also suffers from the problem of computationally expensive step of map warping or scrolling as the robot moves.

A pushbroom stereo scanning method is proposed in [75] for obstacle detection for MAVs flying at high speeds. As the robot moves, disparity measurements equal to a fixed value are collected to generate a map of the environment. Since the collected disparity measurements are at a fixed distance, usually not very far to obtain reliable measurements from stereo camera, it is only suitable for short distance planning.

We base our work on [76] which proposed a C-space expansion step to apply an extra padding around disparities based on robot size. The method in [77] works when planning in spaces where the stereo system disparity is not very noisy i.e. in close range making the planning system myopic in nature and prevents long planning horizon.

Although cameras provide rich information about the environ-

[71] Lionel Heng, Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer, and Marc Pollefeys. Autonomous obstacle avoidance and maneuvering on a vision-guided mav using on-board processing. In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 2472–2477. IEEE, 2011

[72] Franz Andert and Florian Adolf. Online world modeling and path planning for an unmanned helicopter. *Autonomous Robots*, 27(3):147–164, 2009

[73] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013. ISSN 1573-7527. DOI: 10.1007/s10514-012-9321-0. URL http://dx.doi.org/10.1007/s10514-012-9321-0

[74] Pascal Gohl, Dominik Honegger, Sammy Omari, Markus Achtelik, Marc Pollefeys, and Roland Siegwart. Omnidirectional visual obstacle detection using embedded FPGA. *IEEE International Conference on Intelligent Robots and Systems*, 2015-Decem:3938–3943, 2015. ISSN 21530866. DOI: 10.1109/IROS.2015.7353931

[75] A. J. Barry and R. Tedrake. Pushbroom stereo for high-speed navigation in cluttered environments. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3046–3052, May 2015. DOI: 10.1109/ICRA.2015.7139617

[76] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss. Stereo vision-based obstacle avoidance for micro air vehicles using disparity space. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3242–3249, May 2014. DOI: 10.1109/ICRA.2014.6907325

[77] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss. Stereo vision-based obstacle avoidance for micro air vehicles using disparity space. In *2014 IEEE International Conference on Robotics and Au-*

ment, the aforementioned approaches fall short of utilizing this rich information.

Planning Pipeline

Mathies et. al [78] presented an approach to use disparity images generated by a stereo pair for obstacle avoidance. In this approach the occupied pixels in the disparity image obtained from the stereo pair are expanded to account for robot's size. The expanded disparity images are used as a spatial representation to plan collision free paths.

[78] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss. Stereo vision-based obstacle avoidance for micro air vehicles using disparity space. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3242–3249, May 2014. DOI: 10.1109/ICRA.2014.6907325

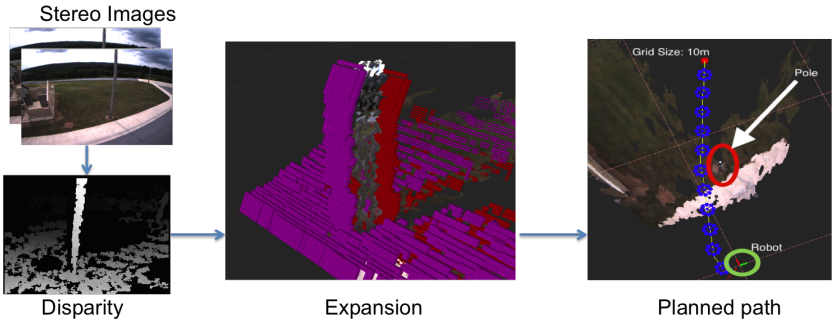

Figure 20: Planning pipeline based on inverse depth obstacle perception. The frontal expansion and back expansion are shown in pink and red point cloud around the original point cloud of pole. Planned path around the pole is also shown with the current robot position circled in green.

We maintain a similar pipeline to process the disparity images but improve the expansion step through the inclusion of the observation noise model in the disparity expansion. Furthermore, we compute two image expansions; frontal and back to probabilistically capture the occupancy region. Figure 20 and Figure 22 show how the two images capture the pole obstacle. The frontal expansion is shown in pink point cloud and the back expansion in red. We also improve the path planning by using multiple disparity images to infer occupied volumes. The use of multiple disparity images allows the planner to reason about long range obstacles. The improved expansion algorithm and multi-image occupancy inference are presented in section . Furthermore, all the planned paths end at hover position with zero velocities ensuring safety of the vehicle. Figure 20 briefly shows the planning pipeline.

Local Perception & Planning

We use disparity image or inverse depth image for obstacle representation as it naturally captures spatial volume according to the sensor resolution [79]. This representation is befitting for noisy stereo data as explained in Section . We employ C-space expansion where the

[79] Pascal Gohl, Dominik Honegger, Sammy Omari, Markus Achtelik, Marc Pollefeys, and Roland Siegwart. Omnidirectional visual obstacle detection using embedded FPGA. *IEEE International Conference on Intelligent Robots and Systems*, 2015-Decem:3938–3943, 2015. ISSN 21530866. DOI: 10.1109/IROS.2015.7353931

original disparity image is expanded, allowing us to treat the robot as a point when doing collision checks during planning [80].

Our method incorporates a stereo sensor error model and allows us to reason about space behind obstacles. We use an additional padding in disparity both in front and behind obstacles. This padding varies from 3σ for close obstacles to 1σ for far obstacles, where σ is the standard deviation of disparity error and the multiplier is represented by λ in later sections. By varying λ we ensure safe planning at short range and a more optimistic planning at long range. This enables the deliberative planning required for exploration tasks.

[80] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss. Stereo vision-based obstacle avoidance for micro air vehicles using disparity space. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3242–3249, May 2014. DOI: 10.1109/ICRA.2014.6907325

Disparity error and its effects

Disparity is a measure of the proximity of an obstacle. We can derive how close the obstacle is in depth using triangulation in stereo vision as follows.

$$z = \frac{bf}{d} \tag{27}$$

Where, z is the depth of a pixel(u, v) with disparity d, b is baseline and f is the focal length in pixels.

The actual 3D point can be derived as

$$P(x, y, z) = (uz/f, vz/f, z) \tag{28}$$

The accuracy of the stereo setup is drastically affected as the disparity decreases. The error in depth increases quadratically with depth as shown in equation(31). Differentiating equation(27) wrt d

$$\frac{\partial z}{\partial d} = -\frac{bf}{d^2} \tag{29}$$

$$\partial z = -\frac{z^2}{bf}\partial d \tag{30}$$

$$\partial z \sim z^2 \tag{31}$$

Disparity error is primarily caused due to correspondence error while matching pixels along the epipolar line. It can be modelled using a Gaussian pdf. Assuming correspondence error during disparity computation has a std deviation $\sigma = 0.5 pixels$, we define the Gaussian pdf $\mathcal{N}(d, \sigma^2)$. Figure 21 shows how this Gaussian pdf in disparity results in a difficult to model pdf for error in depth with an elongated tail on one side and a compressed tail on the other. This motivates to use disparity image space domain directly for occupancy inference rather than resorting to depth or 3D domain.

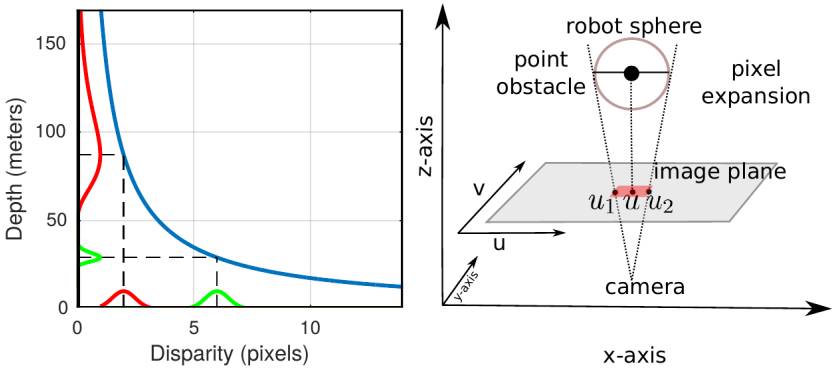

Figure 21: Left: Disparity vs Depth (blue) and probability distributions are shown in red and green. Red and Green PDF in disparity are same and easy to model but their corresponding Red and Green PDF in range vary and difficult to model. Hence we use inverse depth space to represent obstacles. Also, disparity i.e. inverse range captures space at multi-resolution suitable for registration of stereo sensor data. Right: Shows the pixel-wise expansion of a point obstacle according to robot size.

C-Space Expansion

C-Space expansion is required to represent obstacles such that a single point state query can be used for collision checks. Occupancy grids have been the default methods for registration of sensor data and C-Space expansion for occupancy inference. Usually point clouds are used to populate occupancy grids but point cloud generated using disparity images are highly uncertain at greater depths Figure 27(c) and hence occupancy grid based representation is infeasible. Moreover, 3D occupancy grids require a huge amount of memory to capture the planning workspace and hence fail to incorporate long range measurements available from stereo sensors. To overcome this limitation we use disparity images and apply disparity expansion step explained in section .

Disparity Expansion

In this section we explain the step of C-Space expansion as applied to disparity images. This step allows us to capture the volume occupied by an obstacle using two surfaces represented by two disparity images. These images represent front and back surface limits of the reported disparity. Each pixel in these two images effectively captures the range of disparity based on robot size and the sensor error model as shown in the Figure 21. This process can be divided into two steps.

The first step expands disparities along the image XY axis Figure 21(right) i.e. an obstacle at some pixel (u, v) after inflation occupies a manifold of pixels from $[u_1, u_2]$ and $[v_1, v_2]$. This is achieved by traversing through the image row-wise first and then column-wise. This is similar to [81] but we also incorporate sensor error. We omit the steps required to generate the look-up-table (LUT) to map $u \rightarrow [u_1, u_2]$ given disparity d and $v \rightarrow [v_1, v_2]$ given disparity d.

[81] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss. Stereo vision-based obstacle avoidance for micro air vehicles using disparity space. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3242–3249, May 2014. DOI: 10.1109/ICRA.2014.6907325

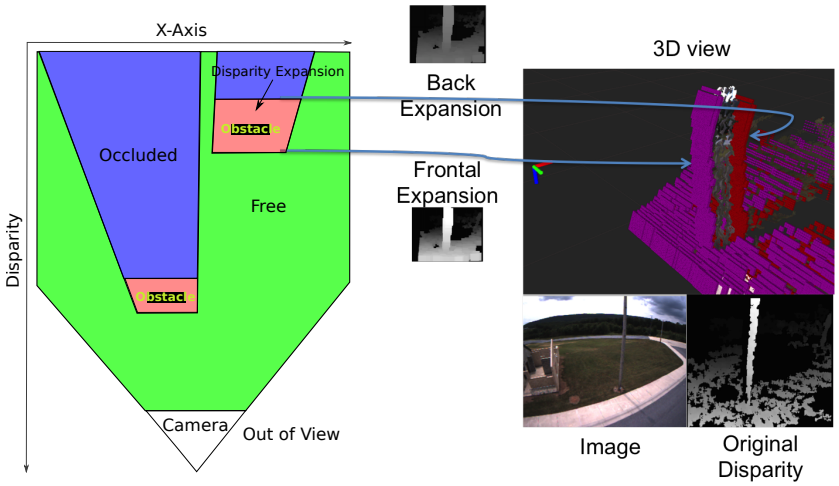

Figure 22: Disparity expansion shown as point cloud. The pink and red point cloud represent the foreground and background disparity limits.

Reader is advised to refer [82] for generation of the LUT, but unlike looking up for the raw disparity value d from table we look up for $(d + \lambda\sigma)$, where λ is the sigma multiplier dependent on the range as discussed previously in Section .

The second step expands disparities to get new values for front and back images using equation(32). These images represent the maximum and minimum disparities for every pixel respectively.

$$
\begin{aligned}
z &= \frac{bf}{d} \\
d_{\mathrm{f}} &= \frac{bf}{z - r_v} + \lambda\sigma \\
d_{\mathrm{b}} &= \frac{bf}{z + r_v} - \lambda\sigma
\end{aligned}
\tag{32}
$$

Where r_v is the expansion radius based on robot size, d_{f} and d_{b} are the computed front and back disparities which encompass the obstacle. As shown in illustration on left side of Figure 22, the red area around the original disparity of obstacle is the padding generated in the expansion step. This padding is based on the robot size and sensor error model.

Our approach uses the LUT as shown in Algorithm(10) which takes the original disparity image D as input and processes it to generate the expanded frontal and back disparity images D_f and D_g respectively. The function $expand(d)$ implements equation(32) with $\lambda = 0$ and $r_v = 0$ to prevent double expansion in depth. The function $connectedComponent()$ searches for minimum disparity connected to the maximum disparity over steps of provided *range* (set to a multiple of robot radius). This helps to find an obstacle bounding volume. We do not want to use the minimum disparity in

[82] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss. Stereo vision-based obstacle avoidance for micro air vehicles using disparity space. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3242–3249, May 2014. DOI: 10.1109/ICRA.2014.6907325

a window as that can be located very far with no connection to the actual obstacle and hence the $connectedComponent()$ step is required.

Algorithm 10: Disparity Expansion Algorithm

Input: Disparity image D
Output: Expanded disparity images: D_f, D_b
for $v = 1 : Height(D)$ **do**
  for $u = 1 : Width(D)$ **do**
    $\hat{d} = ceil(D(u,v) + \lambda\sigma)$
    $[u_1, u_2] = LUT(u, \hat{d})$
    $V = D(u_1 : u_2, v)$ `// Get vector of disparities`
    $d_f = expand(max(V))$
    $d_b = expand(connectedComponent(d_f, range))$
    for $i = u_1 : u_2$ **do**
      $D_f(i,v) = \max(d_f, D_f(i,v))$ $D_b(i,v) = \min(d_b, D_b(i,v))$
    end
  end
end

Algorithm(10) does row-wise expansion and its result is then subject to column-wise expansion in a similar fashion with λ and r_v set to default values when using $expand()$ function. The expanded disparity images constitute a single snapshot volumes occupied by obstacles. To maintain a spatial memory we create a pose graph consisting of multiple expanded disparity images as described in the following section.

Pose Graph of Disparity Images

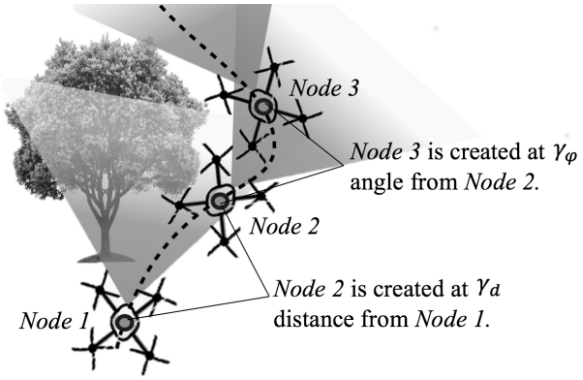

Figure 23: Pose Graph of expanded disparity images. Dashed path shows robot motion and stored nodes in the graph are shown as triangles. Nodes are stored at intervals of distance and orientation.

The motivation to maintain spatial memory of the previously seen environment as the vehicle is moving using a pose graph is because of the following reasons:

1. Previously seen obstacles might not be visible in the current image.

(a) The stereo sensor has a minimum range dependent on maximum perceivable disparity.

(b) Obstacles get occluded in different views.

(c) The field of view is limited.

2. Maintain a pose graph of disparity images (measurements) with nodes at regular intervals of distances and angles as shown in Figure 23.

3. Allows occupancy inference using multiple measurements.

Algorithm(11) shows how we construct this graph. Each node in

Algorithm 11: Pose Graph Algorithm

Input: $D_f, D_g, Pose, N_{graph}, \gamma_d, \gamma_\psi$
Output: Pose Graph of Expanded disparity images: $Graph$
$T_s^w \leftarrow Pose$
$Node = createNode(T_s^w, D_f, D_b)$
if $Graph.size() == 0$ **then**
  $Graph.push\_front(Node)$
  $Graph.push\_back(Node)$
end
$PrevNode = Graph.begin()$
$pos\_err = distance(Pose, PrevNode)$
$ang\_err = angle(Pose, PrevNode)$
if $pos\_err >= \gamma_d || ang\_err >= \gamma_\psi$ **then**
  if $Graph.size() == N_{graph}$ **then**
    $Graph.pop\_back()$
  end
  $Graph.push\_front(Node)$
end
$Graph.pop\_back()$
$Graph.push\_back(Node)$

the graph is comprised of the following:

1. D_f

2. D_b

3. T_s^w which is the transform between the processed sensor measurement(D_f, D_b) and world frame.

The algorithm takes as input the current robot position $Pose$, processed disparity images D_f, D_b, maximum number of nodes N_{graph} and two tolerance parameters γ_d, γ_ψ for position and angular displacement respectively. The constructed graph is used to project a given world point into all node images and do occupancy inference. Occupancy inference using the set of disparity images in the graph is explained in subsequent section.

Occupancy Inference

Evidence grids or occupancy maps are methods to allow fusion of different measurements taken over time. By maintaining a pose graph of expanded disparity images, we can also take advantage of similar fusion without building an occupancy grid which are not suited for stereo data as discussed previously. We devised an occupancy inference method by fusing information from all the images in the graph using the stereo sensor error model. Given the standard deviation of correspondence error σ, we compute confidence of a disparity state in the following manner.

$$C(d) = \frac{(d - \sigma)}{d} \tag{33}$$

Confidence measure from equation(33) gives us a measure of how much can we trust a given disparity for occupancy inference. Thus, long range or low disparity, uncertain measurements have low confidence and update the occupancy with lower values. We further discount measurements that mark an area safe or potentially safe(occluded) by 0.5 to be more conservative about clearing areas previously marked occupied. It should be noted that the potentially safe areas are behind obstacles and have lower disparity state, hence their contribution to occupancy clearance is less due to lower confidence value. In our experiments we get the final occupancy measure by projecting a world point P using equation(34) and equation(28) in disparity images of all nodes in the graph and accumulating the occupancy cost according to Table(3):

Check	Remark	occupancy cost $occ(d_s)$
$d_s > d_f(u,v)$	safe	$-0.5C(d_s)$
$d_s < d_f(u,v)$ and $d_s > d_b(u,v)$	obstacle	$C(d_s)$
$d_s < d_b(u,v)$	potentially safe	$-0.5C(d_s)$

Table 3: Occupancy update

Collision Checking

Collision checking is used to plan a new path and to validate if an existing path is safe to follow. Collision checking is performed using the following mapping of a 3D world point P to image pixel I with disparity d_s:

$$P(x,y,z) \leftrightarrow I(u,v,d_s) \tag{34}$$

A state is in collision if the occupancy measure as shown in equa-

tion(35) crosses a pre-defined threshold γ.

$$Occupancy = \sum_{nodes} occ(d_s) \tag{35}$$

$$Occupancy \geq 0.0 \tag{36}$$

If the occupancy for a state is below the threshold, we consider that state as not occupied by an obstacle.

Planning

We use a sampling based planner, BIT* [83] to draw samples in 3D space which are checked for collision as described in . The output is a collision free path connecting start to goal state.

In our experiments we found that disparity images fluctuate around obstacle edges leading to unwanted replanning due to the current plan being in collision. To remedy this we used two threshold values. A lower value γ_{low} is used during planning to find a path i.e. obstacles are observed sooner even at long distances and hence a more conservative path is obtained. A higher threshold value γ_{high} is used to check the current plan for collision and do replanning in case of collision. The advantage of using two threshold values is that an initial plan is found using a more conservative occupancy map while the replanning is done using a more reliable occupancy map. The reliable occupancy map is not affected by fluctuations in the disparity maps. The thresholds are chosen such that collisions at close range are always detected but have great advantage to not force replanning due to less reliable and fluctuating observations at long range when planning paths to longer distances. In our experiments we have planned paths at distances longer than $100m$ (Figure 27). Figure 20 shows a planned path that avoids a pole obstacle. This path is sent to the motion controller of the vehicle.

[83] Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. Bit*: Batch informed trees for optimal sampling-based planning via dynamic programming on implicit random geometric graphs. *CoRR*, abs/1405.5848, 2014. URL http://arxiv.org/abs/1405.5848

Motion Control

We developed a path tracker similar to [84]. It takes the current trajectory and uses feed-forward velocities specified in the trajectory and generates final velocity and heading rates for the low level velocity controller. The low level velocity controller runs on the quadrotor's flight control unit.

The trajectory controller obeys the dynamic limits of the robot and limits the velocities in event of hard turns or sharp changes in trajectory. This allows the planner to generate simple waypoint based paths and rely on the trajectory control to obey vehicle dynamics.

[84] Gabriel M Hoffmann, Steven L Waslander, and Claire J Tomlin. Quadrotor helicopter trajectory tracking control. In *AIAA guidance, navigation and control conference and exhibit*, pages 1–14, 2008

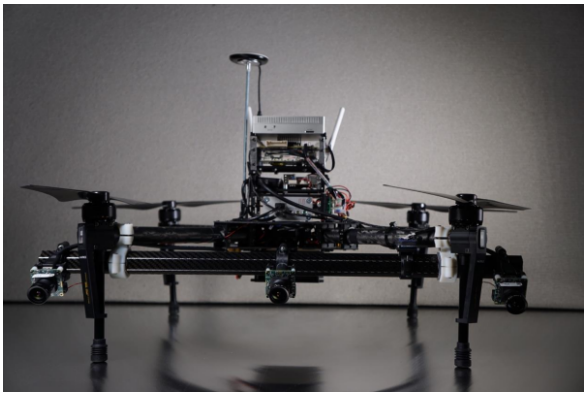

Figure 24: Quadrotor platform used for experiments: equipped with stereo camera sensor suite and onboard ARM computer

We test our algorithms on-board an autonomous UAV system, see Figure 24. The base platform is an off-the-shelf quadrotor vehicle retrofitted with in-house developed sensing and computing suite designed for semantic exploration. The sensor suite consists of a monochrome stereo camera pair, a monocular color camera, an integrated GPS/INS unit and a barometer. The stereo camera pair provides 640×480 resolution disparity image at 10 fps for the obstacle avoidance and 3D mapping systems. The central camera is operated at a lower frame rate, to provide high resolution color imagery for the semantic perception system. All cameras are forward-facing, tilted downwards at 15°, an orientation well suited for low-altitude ($< 40m$) operation. The GPS/INS system and the barometer are used for state estimation.

All computation for autonomous operation is performed onboard. To this end we equip the MAV with two embedded ARM computers; one of them is devoted primarily to planning tasks, while the other is devoted to perceptual tasks. In addition, we use a specialized FPGA processor [85] for stereo depth computation. The computers are networked through high-speed ethernet.

[85] Nerian SP1 Stereo Vision System. URL nerian.com

We conducted most of the experiments in the highlighted area shown in Figure 25. Some of the features of region were narrow trails, dense foliage and varying height tree line, all of which made for challenging and interesting obstacles. Tests involved manual take-off and sending a list of sparse global waypoints to the obstacle avoidance system with the desired velocity. Sparse global waypoints allowed obstacle avoidance system to plan around obstacles determining vehicles path and safety. Table(4) lists the values we used for conducting the experiments.

Figure 25: Location where experiments were carried out: highlighted area

Parameter	Value
Baseline: b	0.35m
Focal length: f	514.17 pixels
Correspondence error: σ	0.5
Connected component range: $range$	$2r_v$
Robot radius: r_v	1.5m
Lenient Occupancy Threshold: γ_{high}	1.8
Strict Occupancy Threshold: γ_{low}	0.9
No. of nodes in Pose graph: N_{graph}	10
Displacement between nodes: γ_d	1.5m
Angle between nodes: γ_ψ	30°

Table 4: Parameters Used

RESULTS

Figure 26 shows the time taken to process a single disparity image to compute the frontal and back expansions using Algorithm(10) on the onboard ARM computer. In our experiments we used CPU version at 320×240 resolution because the GPU was used for semantic classification algorithm as concurrent part of the experiments. A pose graph using Algorithm(11) was created and used for collision checks using equation(35). Using our approach a single occupancy inference and collision check takes on average $0.01ms$. Given $100ms$ between each frame we can do about 2000 collision checks which was usually sufficient for the BIT* planning algorithm.

Figure 27(a) Shows planned path going through two low height trees. The top left is the disparity image with left camera image shown on top right. The point cloud is only for visualization purpose and the trees are marked in red ellipses. Although the trees are not

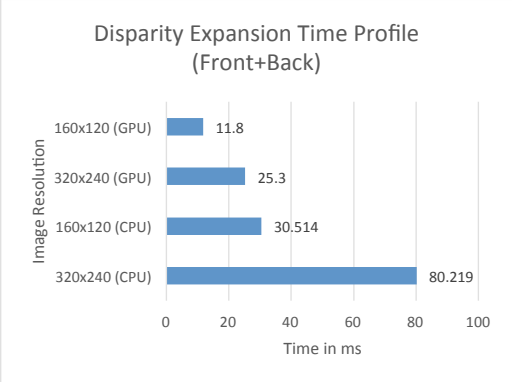

Figure 26: Time profile of expansion step.

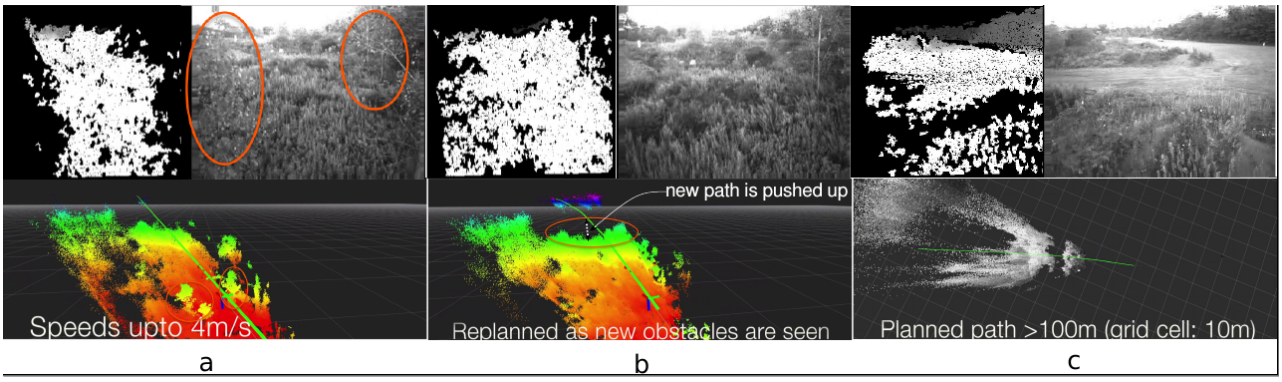

Figure 27: Point cloud is shown at the bottom in all three figures for reference. Point cloud is colored by height in (a) & (b) and by actual intensity in (c). (a) Planned path(green) between low trees highlighted in red ellipses (b) Replanned(green path) as more observations are made, marked in red ellipse, (c) Long range planning horizon. The point cloud shows the noisy measurement but even noisy information allows to infer occupancy at long distances.

completely visible in the current disparity image, they are still a part of obstacles as they were seen at previous robot positions and hence stored in the pose graph. Without the pose graph these trees would have been invisible to the robot. Thus the pose graph helps in keeping memory of obstacles which were seen previously but can't be observed as they exceed the limit of maximum possible disparity after robot motion.

Figure 27(b) shows the previous path was replanned and pushed up as more observations of the bushes/trees are made at long range are marked as obstacles at approximately $30m$ distance from the robot. This was possible due to fusion of occupancy using several disparity images in the pose graph.

Figure 27(c) emphasises the advantage of planning in disparity space at long distances. At greater distances the point cloud is very noisy but we are able to get some information about occupancy by using all the sensor data. While occupancy grids would have huge impact, both memory wise and computationally to use all this data, our approach is able to incorporate all the information using minimalistic image space representation and do better occupancy inference.

Figure 28 shows the reactive nature of our approach. For this experiment the robot was allowed to find a plan outside the sensor's field of view and was given a goal point in right direction. As the robot follows the plan and turns right, an obstacle obstructing its path is detected and a new plan avoiding it is generated. This happened at a speed of $4m/s$ hence implying our approach quickly reacts to newly seen obstacles. Figure **??** shows the third person view of the same run.

Using the parameters specified in Table 4, if the robot moves $15m$ maintaining 10 nodes and assuming a maximum of only $100m$ depth ($1.79pixel$ disparity) per image our approach uses approximately 38% of the memory required by a gridmap of cell size $1m^3$ covering the same volume. This is the case when using a gridmap of large cell size meaning a very coarse resolution. For a better resolution gridmap will require even more memory.

More than 100 runs were executed with approximately 1.6 hours in autonomous mode, covering a cumulative distance of approximately $1.5Km$. The maximum speed was capped at $4m/s$. Our approach allowed us to plan to distances greater than $100m$ as shown in Figure 27(c). Average distance to goal was $36m$. The standard deviation of length of planned paths from straight line paths was on average of $1.38m$ with a maximum of $30m$. This shows that in most cases planned paths were close to a straight path but with slight deviation to avoid obstacles.

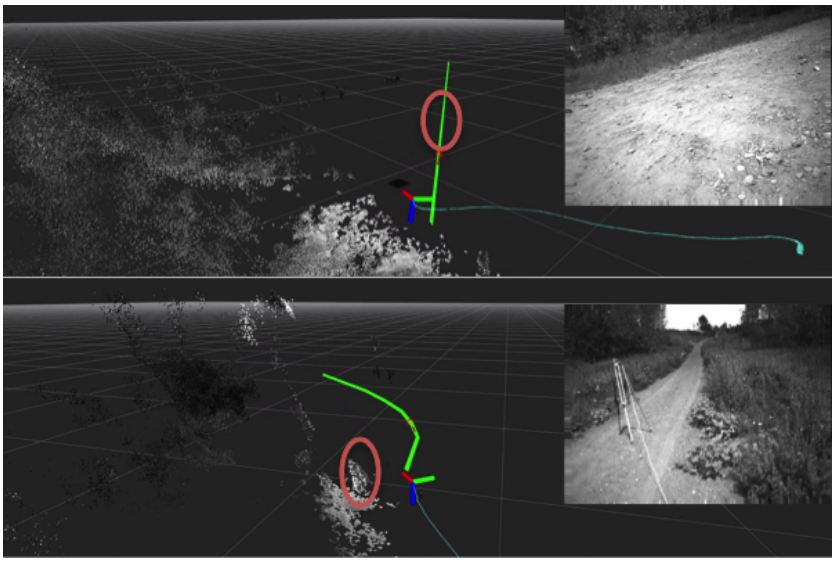

Figure 28: Reactive Planning at $4m/s$: Top image shows the robot has planned to go right with unseen obstacle marked in red ellipse. Bottom image: after banking right an obstacle obstructs the previous plan and a new plan avoiding it is generated.

CONCLUSIONS & FUTURE WORK

We have presented an approach and a system design that allows high speed, non-myopic obstacle avoidance. We demonstrated the system flying at $4m/s$ in dense foliage while relying on stereo image data for modeling the world. To our knowledge it is the first one to do so.

The key factor that enables our system to perform safely at high speeds in highly cluttered environments is integrating multiple stereo sensor frames in real time while reasoning about the related highly non-linear noise model to generate a world representation in inverse depth space. Using FPGA hardware for disparity calculation, combined with fast disparity expansion allowed us to limit our computational burden. This allowed the system to share computation with classification and state estimation tasks.

The current disparity space representation lacks an explicit model of unknown space, rendering the system vulnerable to collision whilst operating in environments with complex geometries. We are currently working on this issue by performing disparity space expansion over multiple layers whilst guaranteeing vehicle safety using emergency maneuver libraries [86] in conjunction with active control of heading along the lines of the sensor planning approach suggested in [87].

[86] Sankalp Arora, Sanjiban Chowdhury, Daniel Althoff, and Sebastian Scherer. Emergency maneuver library - ensuring safe navigation in partially known environments. In *IEEE, International Conference on Robotics and Automation*, 2015

[87] Sankalp Arora and Sebastian Scherer. Pasp: Policy based approach for sensor planning. In *IEEE, International Conference on Robotics and Automatio*, 2015

Mapping

Given a semantically classified image, we want to find the position of objects detected in the image, as well as model regions for which the information in measurements is uncertain. Since, this mapping has to be performed on board the vehicle, the driving requirement of the application is computation time. Given a global pose by state estimation filter, each pixel in the labeled image defines a ray originating at the camera center and passing through the pixel center. To perform the mapping operation we use the images with soft pixelwise predictions, together with the robot's global pose estimate and a pre-existing digital elevation map (DEM). We exploit the semantic knowledge of the world (every object rests on the ground) and use the digital elevation map to infer the 3D structure of the environment.

Minimal computation cost has allowed occupancy grid based mapping algorithms to be successfully deployed on-board robots. The reduced the computational complexity is achieved by reducing the dimensionality of the mapping problem by assuming cells in a grid are independent binary random variables and measurements are independent, given a cell's true occupancy value. These assumptions have been shown to work effectively with sensors that provide both range and bearing.

But a semantically classified image provides bearing only measurements through rays originating from camera pose, making the ray independence assumption limiting. To fully exploit the bearing only measurement and the semantic structure knowledge of the world, we need to model ray dependence. Section and section describe how we model dependence amongst observations while still allowing for an on-line mapping algorithm.

Exploiting Semantic Knowledge

We assume that objects of our interest, represented by $\mathcal{L}_{\mathcal{M}} = \{c_1, c_2, ...c_n\}$, rest on the ground and we know the likely height $h_{c_i} \forall c_i \in \mathcal{L}_{\mathcal{M}}$. We model the world as a $2.5D$ grid. In every cell, C_{ij} of the grid at location i, j, we store the heights at which rays pass over the cell for all classes by casting rays originating from the classified image, table 5. We are interested in finding the cells where the height of rays passing over the cell match the height of object we are looking for, while accounting for occlusions and limited field of view. This leads to following cases for a given class in a cell C_{ij}–

- *Case o-* Average probability of rays that pass over cell C_{ij} with a label other than class c is greater than average probability of rays with class c.

Symbol	Description
$C_{ij}.c.h_u$	The highest height at which a ray with label c passes over or intersects the cell C_{ij}.
$C_{ij}.c.h_l$	The lowest height at which a ray with label c passes over or intersects the cell C_{ij}.
$C_{ij}.c.n_f$	The number of rays with label c that pass over or intersect the cell C_{ij} at a height less than h_c.
$C_{ij}.c.n_a$	The number of rays with label other than c pass over or intersect the cell C_{ij} at a height less than h_c.
$C_{ij}.c.p_f$	The cumulative probability of rays with label c that pass over or intersect the cell C_{ij} at a height less than h_c.
$C_{ij}.c.p_a$	The cumulative probability of rays with label other than c that pass over or intersect the cell C_{ij} at a height less than h_c.
$C_{ij}.c.l_o$	Integrated logodds of an object of class c being present in the cell C_{ij}.

Table 5: Data members of grid cell C_{ij} for class c.

- *Case 1*- Rays of some other class pass from below and above the class of concern over the cell C_{ij}.
- *Case 2*- Rays of some other class pass from below and nothing is observed above the class of concern over the cell C_{ij}.
- *Case 3*- Nothing is observed above or below the class of concern over the cell C_{ij}.
- *Case 4*- Nothing is observed below and some other class is observed above the class of concern over the cell C_{ij}.

Case 1 implies that the cell is well-observed. Therefore, $C_{ij}.c.h_u$ and $C_{ij}.c.h_l$ should be close to h_c and ground height respectively. *Case 2* implies that the upper part of the object could not be sensed due sensing geometry or occlusions. Hence, $C_{ij}.c.h_u$ should be less than h_c and $C_{ij}.c.h_l$ should be close to ground. Similarly, *Case 3* implies that $C_{ij}.c.h_u$ and $C_{ij}.c.h_l$ should be less than h_c and *Case 4* implies that $C_{ij}.c.h_u$ should be close to h_c and $C_{ij}.c.h_l$ should be less than h_c. These cases lead to equation 37, that is used to determine whether there is positive, negative or lack of evidence in the current classified frame regarding the presence of object of class c over the cell C_{ij}

$$\phi_{ij}(c) = \begin{cases} 0, & \text{Case o} \\ e^{\alpha_k C_{ij}.c.h_l} e^{\beta_k (C_{ij}.c.h_u - h_c)/h_c} \dfrac{C_{ij}.c.p_f}{C_{ij}.c.n_f}, & \text{Case k} \end{cases} \tag{37}$$

Where, $k \in [1,4]$ and α_k, β_k are negative constants that allow us to change the weights of the measurements according to the cases encountered. We use the following values for these constants, $\alpha_1 = \beta_1 = -10$, $\alpha_2 = -10, \beta_2 = -1$, $\alpha_3 = -1, \beta_3 = -1$, $\alpha_4 = -1, \beta_4 = -10$. Obviously a value of $\phi_{ij}(c)$ close to 0.5 indicates lack of evidence, and $\phi_{ij}(c) < 0.5$ indicates negative evidence and $\phi_{ij}(c) > 0.5$ positive evidence for the presence of class c in cell C_{ij}.

Temporal Evidence Integration

$\phi_{ij}(c)$, allows the algorithm to model the dependence amongst rays, while allowing us to treat the cells independently. We assume at any given cell, the log odds of probability of observing a class c is given by a constant γ. Each class in a cell is represented as an independent binary random variable, as a cell can have objects of multiple classes. Once the nature of evidence $(\phi_{ij}(c))$ is identified,logodds for each class in each cell is updated with equation 38.

$$C_{ij}.c.l_o = \begin{cases} C_{ij}.c.l_o, & |\phi_{ij}(c) - 0.5| \leq \zeta \\ C_{ij}.c.l_o + \gamma C_{ij}.c.(n_f - n_a), & \text{otherwise} \end{cases} \tag{38}$$

Where, ζ is a small positive number less than 0.5. We use $\zeta = 0.2$ and $\gamma = 1$. Each semantically classified image is integrated with the grid and $C_{ij}.c.l_o$ is updated for every cell that needs updating, this process is repeated for every input semantically classified image. Next section presents the hardware system on which we run the semantic mapping system to enable autonomous scouting. Preliminary results for the mapping algorithm are presented in section .

Platform

Figure 29: Aerial Platform.

Our current MAV is depicted in figure 29. The base platform is an off-the-shelf quadrotor DJI vehicle retrofitted with our own sensors and computing payload designed for autonomous scouting.

Sensing

The sensor suite consists of a monochrome stereo camera pair, a monocular color camera, an integrated GPS/INS unit and a barometer. The GPS/INS system and the barometer are used for state estimation.

All cameras are forward-facing, tilted downwards at 15°, an orientation well suited for low-altitude (< 40 m) operation. The horizontal field of view for this camera is approximately 60°, which we considered a good compromise between coverage and object size, given the sensor resolution of 1600×1200 pixels.

Hardware Platform

All computation for autonomous operation is performed on-board. To this end we equip the MAV with two embedded ARM computers. The first is an Nvidia TK1, which features a low-powered GPU which we use for semantic segmentation. This computer also runs other perception-related tasks necessary for autonomous operation. The second is an ODroid XU4, which runs the mapping and planning systems.

Software Platform

Both computers use ROS on Ubuntu Linux. Our segmentation and mapping methods run concurrently as ROS nodes and communicate through messages. The segmentation node, implemented in Python, uses the Theano [88] library with the Nvidia CuDNN backend to make effective use of the GPU. The mapping algorithm is CPU-only and is implemented in C++.

[88] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *SciPy*, June 2010

Mapping

In this section we demonstrate the effects of exploiting semantic knowledge and modeling ray dependence qualitatively, while measuring the sensitivity of the mapping algorithm to height inaccuracies in the DEM. Figure 30-4 shows a canonical scenario where a car, more than $50m$ away, is detected by the semantic classification algorithm. Exploiting semantic knowledge and modelling dependence allows the mapping algorithm to capture the uncertainty about the presence of a car in the cell occluded by the car, figure 30-1, whereas if we do not reason about ray interdependence, the occluded cell is also inferred to contain cars figure 30-3. If both the semantic knowledge and ray interdependence are not exploited, then a simple projection of classified image to the DEM leads to an inference that multiple cells are occupied by a car figure 30-2. Demonstrating that modelling the ray interdependence and exploiting semantic knowledge leads to better mapping of objects and uncertainties. Figure 30 shows that the algorithm's performance deteriorates in presence of height errors in the DEM. Unsurprisingly the degradation is faster if DEM underestimates the height of the cells due to observation geometry.

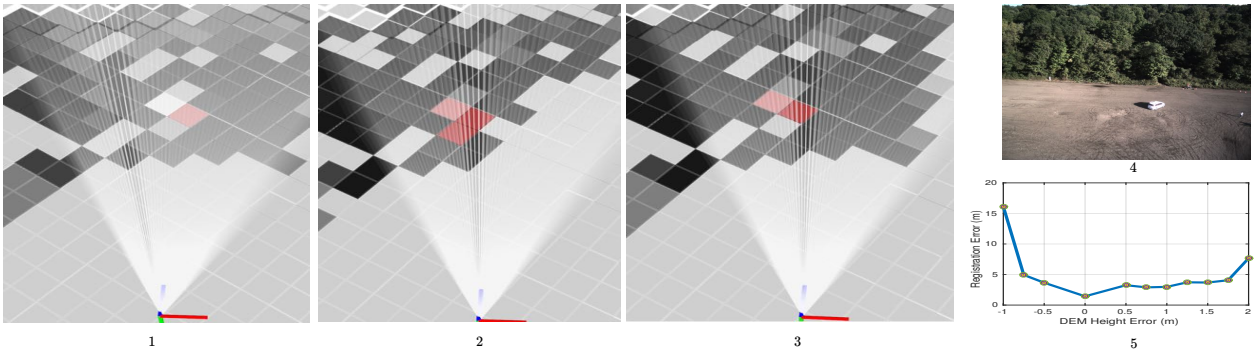

Figure 30: Figures 1 shows the updated map after a classified image 4 is integrated in our current mapping pipeline, Figure 2 shows the updated map if the classified image is projected on the DEM without exploiting semantic knowledge and Figure 3 shows the updated map if the ray interdependence is not modelled.

Conclusions

In this paper we have described a semantic mapping system aimed to support autonomous scouting with MAVs. We evaluated the two main components of the system in isolation and demonstrated an integrated autonomous mission.

We are currently improving this system in several ways. We are labelling a larger dataset, including more semantic classes. With this dataset we hope to get a more accurate picture of the performance limits of our method. We are also evaluating methods from recent work in semantic segmentation aimed to optimize performance.

At the same time, we are planning more field experiments, in order to gather data and evaluate the integrated system quantitatively.

In the future, we are interested in using the image data for dense 3D reconstruction, hoping to avoid the need for an external DEM, or even a GPS.

Proposed Work

Problem

Data acquisition is relatively easy in the virtual world, where the cost of accessing data can be equivalent to accessing a memory block. But the cost of data gathering in physical spaces, where it is impractical to have sensory networks is not as trivial. Currently we rely on humans to carry or drive sensors around to digitize the physical world to collect data.

Gathering data from an oil pipeline to predict future leaks or detect current ones, collecting data from a lake to sample spatial distribution of water contaminants to predict outbreak of diseases, gathering data about flood survivors to aid in search and rescue missions, or to collect data about bridges to predict corrosion rates. All these applications require data gathering in the physical realm at an unprecedented scale. Using humans to do these tedious, boring and often risky tasks is far from ideal. Therefore, the need is urgent to develop autonomous robots to do the job for us.

We need safe data gathering agents that can effectively gather data while reasoning about their physical and sensory constraints. Unmanned Aerial Vehicles (UAVs) are well suited for such data gathering tasks. The primary reason is their ability to reach vantage points unattainable by other systems. UAVs are able to switch from viewing large spaces at a distance to flying in closely to obtain more accurate information. This capability of gaining information at different scales makes UAVs excellent for aforementioned applications.

Consider a situation where a region is affected by floods. There is an urgent need to locate survivors, provide supplies and establish communication with them. We have at our disposal a UAV with a camera pair that can fly up to a limited distance of 1 km. Then the problem becomes that of locating and counting as many survivors as possible while being restricted to fly 1 km. The robot needs to reason about the fact that it can effectively gain information about the area by gaining height and subsequently flying close to survivors to get a more detailed picture. Given limited sensing bandwidth

on the vehicle the same camera pair has to be used for gathering information about the survivors whilst ensuring its safety.

Thus the four salient characteristics of the problem are

1. *Constraint on the total travel distance*. Due to limitations of fuel / battery, the length of the route taken by the robot is limited.

2. *Correlated nature of information*. Equipped with cameras, UAVs can view large areas from a distance to gain information. This leads to reward of visiting different locations being correlated.

3. *Safety Constraint*. The vehicle has to ensure that it will never run into obstacles while gathering data.

4. *Constraint on sensing bandwidth*. Limited payload carrying capacity, limits the number of sensors the vehicle can carry. Leading to limited sensing bandwidth.

The problem of planning routes to gain information is an NP-hard problem . The current state of the art systems attempt to solve this problem using two lines of approach. One approach is to apply a myopic strategy where a set of sensing locations is identified and the system travels to the most promising one. While such strategies are computationally efficient, they fail to effectively account for the constraint on traveling distance. As a result the computed routes can lead to oscillatory behavior.

Another approach is to invoke a long horizon planner . However, these approaches are far from real-time and ignore the need for gathering data for safety of the vehicle or make simplifying assumptions on vehicle dynamics, leading to the vehicle operating at slow speeds.

Generally user explicitly specifies the data that is of relevance for a data gathering mission. For a pipe inspection robot, the user explicitly specifies to gather data about cracks and rust on pipes. We call such a data gathering requirement as *explicit*.

A safe, robust and efficient autonomous data gatherer however needs to not only satisfy the explicit data gathering requirement but also gather data *implicitly* required by the robot to keep itself safe, and improve its assumptions about its environment for better long-term deployment.

We propose to develop online implicit and non-myopic explicit data gathering techniques/pipeline that will enable safe and efficient information gathering agents with the ability of gathering data to improve their assumptions about the world for improved long term deployment.

A decision theoretic approach to such a data gathering problem involves planning sensing actions while reasoning about distribution of the environments. Planning with high uncertainty online is infeasible given the computation constraints on mobile platforms. Section we present the formalization of the data gathering problem. Section **??** briefly describes the challenges presented by the data gathering problem and the completed work that addresses some of those challenges. We discuss open research questions and proposed solutions in the proposed work section, Section .

Proposed Work

Robots have an implicit need to gather data to complete their mission safely and efficiently. While data gathering robots are deployed to explicitly gather data about the environment. The emergency maneuver library combined with policy based sensor planning provides a framework to address the implicit need for data gathering for safety of mobile robots. *RAOr-G* provides a global informative path planning algorithm for robots that are explicitly tasked with gathering information. These methods have taken us closer to efficient data gathering in physical spaces. However the results have uncovered some interesting research questions that need to be tackled to develop a framework for safe, efficient and robust data gathering robots, as described in Figure **??**.

We propose techniques to fill the gaps in developing a safe, efficient data gathering robot. In, Section **??** we discuss the problem of combining implicit information gathering for safety with the explicit requirement of the robot to gather information. We propose a sampling based model predictive planner to efficiently explore the search space.

Section **??** explored the methods to develop faster algorithms for calculating efficient budgeted information data gathering paths. We exploit local lipschitz continuity and local eikonal for improving paths, while scheduling heuristics.

Section **??**, discusses the problem of data gathering for improvement of world models by the robot. We present a method to direct the curiosity of the robot, such that its world model is improved and as a result its long term performance in a given distribution of the environment.

Section describes the case study we will undertake to validate the proposed data gathering pipeline via an unmanned aerial vehicle.

Combining implicit and explicit information gathering

Emergency maneuver library allows the vehicle to stay safe while operating in partially known environments. We have used the emergency maneuver library with a myopic one-step look ahead, independent of path planning. Treating path planing and safety independently allows real-time guaranteed safe planning for systems with highly non-linear dynamics. But the safety planning independence pipeline, potentially leads to sub-optimal system behavior.

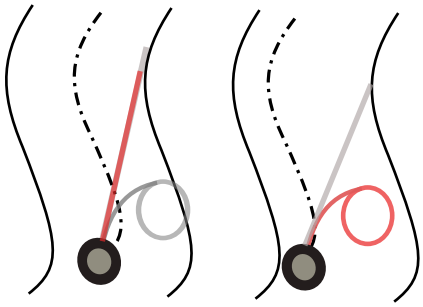

Figure 31: Illustrated here is 10*X*10 grid size benchmark problem with RRO-G run. For all the benchmark problems the nodes are situated in a uniform grid with 1 resolution. For this particular problem RRO-G is able to find a near-optimal solution in 6.9 seconds while the state of the art takes 143.6 seconds for finding same quality of solution.

For example, in figure 31 if the planner and *EML* are operated independently, the planner maximizes the distance from obstacles and finds a path in the middle of the corridor, leading to a slower operation of the vehicle. Whereas if the planner actively optimizes the maximum safe speed while cognizant of the emergency maneuver library takes a path much closer to one of the walls of the corridor, allowing for faster safe navigation.

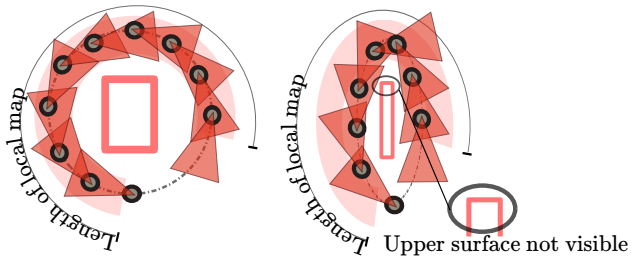

Figure 32: Illustrated here is 10*X*10 grid size benchmark problem with RRO-G run. For all the benchmark problems the nodes are situated in a uniform grid with 1 resolution. For this particular problem RRO-G is able to find a near-optimal solution in 6.9 seconds while the state of the art takes 143.6 seconds for finding same quality of solution.

Efficient data gathering also requires the vehicle to gather relevant data while being safe. The simplest approach to solving this problem is to gather data for safety when required, while focusing sensing bandwidth on explicit data gathering needs as the vehicle flies at a constant velocity. Figure 32, explains a scenario where myopic constant velocity strategy works for a circular path, but the same approach leads to missing pertinent information if the curvature of the path is relatively higher.

The related informative path planning problem gathering can be posed as POMDP. But such a formulation is intractable and we need to reason for the space of maps to find the optimal answer. Kostas et. al, tackled this high dimensional problem by using randomized sampling techniques and expected information to forward simulate sensors. We propose to use randomized trees to address the problem. One of the issues with Rapidly exploring randomized trees operating in high dimensional spacs tend to get stuck in local minima. We will initialize these RRT's using primitives generated offline. *Use the word embed.*

Safe Local Informative Path Planning

The output of the global informative path planner provides a desired position of the data gathering agent x_t, the set of targets from which information needs to sensed X_t^g and the desired velocity v_t^g. The global plan is constructed using a low resolution global map. Hence, it does not take small/unknown obstacles or occlusions into account. A planner is therefore required for safe and efficient data gathering to plan around these artifacts. We propose using a sampling based receding horizon planner to generate safe and efficient local data gathering plans.

Let $\mathcal{X}$ be a bounded connected open subset of $\mathbb{R}^d$, where $d \in \mathcal{N}$. Let, the current state of the robot at time t be given by $x_{init} = x_t$ and the goal region for $\mathcal{X}_G$ be a region around the $x_{t+\tau}$, such that $x_{init} \in \mathcal{X}$, $\mathcal{X}_G \in \mathcal{X}$. Let the belief of the robot at time t be given by b_t. Let $\mathcal{X}_{obs}^{b_t} \subset \mathcal{X}$ be obstacle region according to the belief b_t. Let us denote obstacle free space by $\mathcal{X}_{free}^{b_t} = \mathcal{X} \setminus \mathcal{X}_{obs}^{b_t}$. In the sequel, a path in $\mathcal{X}_{free}^{b_t}$ is said to be collision-free path. Let $s \geq 0$ be path length and the path be defined by $\sigma : [0, s] \to \mathcal{X}_{free}^{b_t}$, where $\sigma(0) = x_{init}$.

A randomized tree $\mathcal{T} = (V, E)$ is constructed as described in algorithm 12. A branch $br \in E$ is defined as valid if the root of the branch is x_{init} and the the branch ends in $\mathcal{X}_G$. The reward function, $R_{local} : E \to \mathbb{R}$, returns the reward value of a valid branch. The objective of tree generation is to find a path that maximizes the reward function. We briefly describe the primitives below and then present the safe local informative path planning algorithm 12.

Sampling: The function $Sample : \mathbb{N} \to \mathcal{X}_{free}$ returns independent identically distributed samples.

Steering: Given two points $x, y \in \mathcal{X}$, the function $Steer : (x, y) \to z$, return a point $z \in \mathbb{R}^d$ such that z is "closer" to y than x is. $Steer(x, y) = \arg\min_{z \in \mathbb{R}^d} \|z - x\|$, where $\|z - y\| \leq \eta$, for $\eta \geq 0$.

Nearest Neighbor: Given a tree $\mathcal{T} = (V, E)$ and a point $x \in \mathbb{X}_{free}$, the function $Nearest : (\mathcal{T}, x) \to v$ returns a vertex in $v \in V$ that

is closest to x in terms of a distance function. We will consider the euclidean distance here.

*SelectBestBranch:*The $SelectBestBranch : E \rightarrow E$ function returns a valid branch $br \in E$, such that it maximizes the reward function R_{local}. Given a point $x \in V$, let the branch connecting the root node x_{init} to x be given by $br^{x}_{x_{init}} \in E$. Let $b^{x}_{x_{init}}$ be the simulated belief of the robot if it reaches x, traveling along $br^{x}_{x_{init}}$, at the maximum safe velocity permissible, given that $b^{x}_{x_{init}}$ is generated using expected sensor measurements. Let, the maximum safe speed at state x and $b^{x}_{x_{init}}$ be given by v^{safe}_{x} and the average speed to reach x by v^{avg}_{x}. The velocity at node x is then given by $min(v^{g}_{t}, v^{safe}_{x})$. A potential reward function is a function of expected information gain along the input branch, the path length of the input branch, and the average velocity of the vehicle. We will have to evaluate multiple combinations of above mentioned aspects, apply such reward functions for myopic informative path planning. The differentiating factor here being that the path optimization is done while taking the safety constaint satisfaction into account.

Algorithm 12: Sampling based Receding Horizon Planner

Input: $\mathcal{X}^{b_t}_{free}, b_{x_{init}}, x_{init}, \mathcal{X}_{\mathcal{G}}$
Output: Optimized local path σ
$V \leftarrow \{x_{init}\}$, $E \leftarrow \emptyset$, $i = 0$
while $i \leq N$ **do**
  $\mathcal{T} \leftarrow (V, E)$
  $x_{rand} \leftarrow Sample(i)$, $i = i + 1$
  $x_{nearest} \leftarrow Nearest(\mathcal{T}, x_{rand})$
  $x_{new} \leftarrow Steer(x_{nearest}, x_{rand})$
  if $ObstacleFree(x_{nearest}, x_{rand})$ **then**
    $V \leftarrow V \cup x_{new}$
    $E \leftarrow E \cup \{(x_{nearest}, x_{new})\}$
  end
end
$\sigma(.) = SelectBestBranch(V, E)$
return $\sigma(.)$

Although sampling based path planning has proven to be effective in high dimensional belief spaces, they are known to suffer from bias, and large run-times. In the next section we propose methods to guide the sampling to reduce the search space and create a set data gathering maneuvers to bias the tree generation along the trajectories generated offline.

Avoiding pitfalls of sampling based planning

PRIMITIVE GENERATION We will present how we used offline generated primitives to seed RRT's, potentially saving on computation

time. This section will contain potential features and metric on which these primitives will be learned on.

Multi-Heuristic Global Path Planning

RAOr has enabled non-myopic information gathering to be run online. Given the run-times of the algorithm we are forced to abstract data gathering tasks for an object to a single node, figure **??**. Although this abstraction helps usa reduce the problem size, hence allowing *RAOr* to run in real-time, the reduced granularity of reasoning about information gathering might lead to sub-optimal behavior.

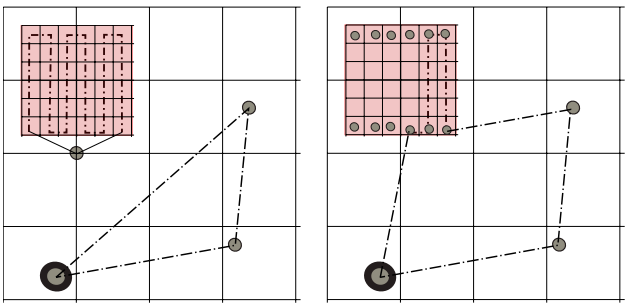

Figure 33: Illustrated here is 10*X*10 grid size benchmark problem with RRO-G run. For all the benchmark problems the nodes are situated in a uniform grid with 1 resolution. For this particular problem RRO-G is able to find a near-optimal solution in 6.9 seconds while the state of the art takes 143.6 seconds for finding same quality of solution.

For example, Figure 33 shows a scenario where the vehicle is scouting for landing zones for a helicopter, if the whole landing zone is abstracted as a single node, the node is not visited as the cost of visiting the node exceeds the traveling budget. A faster global planning algorithm would allow for partial coverage of landing zone, hence better performance on data gathering.

We propose using sets of heuristics to decrease global planning run times for *RAOr*, allowing us to solve bigger informative path planning algorithms on-board the vehicle.

In Section **??** we presented preliminary results for using heuristics to speed up *RAOr*. However, it is hard to design a single heuristic function that works in all environments. Multi-heuristic A^* and our previous work sanjiban et. al developed an ensemble of heuristics to improve the performance of planner operating in multiple environments. However, they fail to take into account the hard run-time constraint on an online algorithm.

Given *RAOr* can operate with multiple heuristics without the compromising its optimality and completeness properties, we propose to develop an ensemble of heuristics that maximizes *RAOr*'s expected quality of the solution given a distribution of planning problems while treating run-time as a constraint. Section formalizes the heuristic set selection problem and provides a linear-time algorithm

to select a near-optimal set of heuristics. The multi-heuristic *RAOr* algorithm is presented in algorithm 13.

Selecting the set of heuristics

Let, $\xi \in \Xi$ be an informative path planning problem. Let, $P(\xi)$ be the probability of our global planner encountering the problem during run time. Let the total run-time available for global planning be B_T, let $h_i \in \mathcal{H}$ be the i^{th}. Let, the maximum run-time for a heuristic h_i for all planning problems $\xi \in \Xi : P(\xi) \geq 0$ be t_{h_i}. Let $R_{\mathcal{H}} : \mathcal{H}X\Xi \rightarrow \mathbb{R}$ be the reward achieved by applying *RAOr* with a heuristic in $\mathcal{H}$ on a planning problem in Ξ. We want to select a set of heuristics $\mathcal{H}_{online} \subseteq \mathcal{H}$ such that the expected reward achieved by the global planner is maximized, equation 39.

$$\underset{\mathcal{H}_{online} \subset \mathcal{H}}{\arg\max} \underset{P(\xi)}{\mathbb{E}} \left[\max_{\forall h_i \in \mathcal{H}_{online}} R(h_i, \xi) \right] \tag{39}$$

$$\sum_{h_i \in \mathcal{H}_{online}} t_{h_i} \leq B_T$$

The optimization problem in equation 39 is combinatorial optimization problem and is NP-Hard. Since the problem is linear combination of maximization, the function being maximized is monotone and sub-modular with a knapsack constraint . We can find a near-optimal solution to equation 39 in linear time as presented in algorithm **??**.

Algorithm 13: Greedy Heuristic Set Selection

Input: $\xi, P(\xi), \mathcal{H}, t_h \forall h \in \mathcal{H}, B_T, R_{\mathcal{H}}(.)$
Output: Near-optimal set of heuristics in $\mathcal{H}$
$\mathcal{H}_{online} = \emptyset$
$B_{\mathcal{H}} = 0$
while $\mathcal{H} \neq \emptyset$ **do**
  $h^* = \arg\max_{h \in \mathcal{H}} \frac{\mathbb{E}_{P(\xi)} \left[\max_{h \in \mathcal{H}_{online} \cup h} R_{\mathcal{H}}(h_j, \xi) - \max_{h_i \in \mathcal{H}_{online}} R_{\mathcal{H}}(h_i, \xi) \right]}{t_h}$
  if $B_{\mathcal{H}} + t_{h^*} \leq B_T$ **then**
    $\mathcal{H}_{online} = \mathcal{H}_{online} \cup h^*$
    $B_{\mathcal{H}} = B_{\mathcal{H}} + t_{h^*}$
  end
  $\mathcal{H} = \mathcal{H} \setminus h^*$
end
return $\mathcal{H}_{online}$

Multi-heuristic *RAOr* with heuristic selected through list selection will allow global planning to run faster enabling faster informative path planning. The heuristic selection algorithm will allow the set of heuristics to be chosedn according to the planning problems the vehicle will face during its lifetime.

In the last two sections we have proposed methods to improve local and global planning for data gathering systems online. In the next section, we discuss the problem of long term data gathering specifically with a focus on developing algorithms that allow the data gathering agent to collect data to improve its assumptions about the environment, hence improving its performance.

Data Gathering for Improving long term deployment

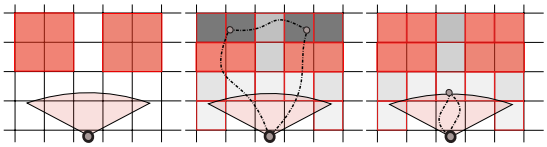

Figure 34: Illustrated here is $10X10$ grid size benchmark problem with RRO-G run. For all the benchmark problems the nodes are situated in a uniform grid with 1 resolution. For this particular problem RRO-G is able to find a near-optimal solution in 6.9 seconds while the state of the art takes 143.6 seconds for finding same quality of solution.

Assumptions are made by data gathering systems for planning and operating effectively under uncertainty. The validity of these assumptions can affect the performance of a data gathering system. Figure 34 presents an example where the robot is using Monocular Semantic Mapping **??** for creating the map of the environment. Wrong assumptions about the environment lead to worse mission performance. If the vehicle is deployed for a long term, the cumulative performance of the vehicle over its deployment might be significantly improved by improving it assumptions about the environment.

In this section we propose a framework to enable implicit data gathering for assumption improvement while exploiting the techniques developed for safe explicit data gathering.

Let the vehicle be deployed for T number of information gathering missions. Let, $a \in \mathcal{A}$ be the vector of assumptions. Let, $b^a \in \mathcal{B}_\mathcal{A}$ be the probability distribution over the assumptions, $b^a : \mathcal{A} \to [0, 1]$. Let $f_\mathcal{A} : \mathcal{B}_\mathcal{A} \to \mathcal{A}$ be the function that maps the probability distribution over assumptions to assumptions used by the robot. The observation model of the robot depends on the assumptions it makes about the environment, let the observation model be given by $P_{b^a}(o|w)$. Let $o^a \in \mathcal{O}^a$ be the observations taken by sensors just after actions $\hat{u}_k = [\{u_k\}, \{u_k^s\}]$ are performed. Let $P_\mathcal{A}(o^a|w)$ be observation model for assumptions. Let $R_\mathcal{A}$ be the reward function. With the weight vector w_a. The optimization is then to find a policy for w_a. let $J_a = \sum_{k=0:N} R(u_k, u_k^s, b_{k-1})$, while equation in intro is maximized be the reward that we actually wanted. $J_a =$

$$\arg\max_{\pi_\mathcal{A}} \sum_{\tau=0:T} J_a \tag{40}$$

$$h(u_k, u_{k-1}, u_k^s, u_{k-1}^s) \leq 0 \forall k = 0 : N$$
$$s_A(b_k, u_k) \leq 0$$
$$b_k(w) = \alpha P(o_k|w) \sum_{\omega \in W} P(w|u_k, u_k^s, \omega) b_{k-1}(\omega)$$
$$\sum_{k=1:N} C(u_k) \leq B$$

Where, α denotes a constant normalizer.
SBO is maximized

MCTS Formulation

As described earlier, MCTS consists for four steps, that are iteratively repeated to construct a search tree until a predefined computational budget b_c is expended.

1. *Root Node:* The root node of the tree is the current belief of the robot about $\mathcal{A}$.

2. *Node:* A node at depth i is defined by the list of $w_{\mathcal{A}}^{1,:i}$ used by the robot to reach that node.

3. *Edges:* The act of using a $w_{\mathcal{A}}^i$ defines an edge.

1. *Select:* An unexpanded node is selected to be expanded and evaluated further.

2. *Expand:* A weight vector, $w_{\mathcal{A}}$ is selected as the next to visit. A new child is added to the unexpanded node.

3. *Playout:* A possible world is sampled from M and from the newly added node a sequence of actions are performed until the depth of T using a default policy. This default policy is described in section **??**.

4. *Backup:* The information gain is evaluated at the end of the playout step and each nodes visit count is incremented by 1 and its reward value is updated according the probability of that trajectory.

MCTS is way to approximate $Q - values$ of the most promising actions. It is obvious that more *Playouts* leads to better $Q - value$ approximation and better decisions under uncertainty. Section **??** describes application of parallelization methods for information gathering,

Case Study

We will use the unmanned aerial vehicle as the platform to test our algorithms. Here we present the system details and scenarios we will test in, with expected improvements.

Bibliography

Nerian SP1 Stereo Vision System. URL nerian.com.

Karim Abdel-Malek, Yingzhou Yang, Denis Blackmore, and Ken Joy. Swept volumes: Foundations, perspectives, and applications. *Int. J. Shape Modeling*, 12(1):87–127, 2006.

F Adolf and J. S. Dittrich. Evaluation of the ARTIS sampling-based path planner using an obstacle field navigation benchmark. In *Proceedings of the American Helicopter Society 68th Annual Forum. Ft. Work, TX*, 2012.

Franz Andert and Florian Adolf. Online world modeling and path planning for an unmanned helicopter. *Autonomous Robots*, 27(3): 147–164, 2009.

Sankalp Arora and Sebastian Scherer. Pasp: Policy based approach for sensor planning. In *IEEE, International Conference on Robotics and Automatio*, 2015.

Sankalp Arora, Sanjiban Choudhury, Sebastian Scherer, and Daniel Althoff. A principled approach to enable safe and high performance maneuvers for autonomous rotorcraft. In *American Helicopter Society 70th Annual Forum*. AHS, 2014.

Sankalp Arora, Sanjiban Chowdhury, Daniel Althoff, and Sebastian Scherer. Emergency maneuver library - ensuring safe navigation in partially known environments. In *IEEE, International Conference on Robotics and Automation*, 2015.

A. J. Barry and R. Tedrake. Pushbroom stereo for high-speed navigation in cluttered environments. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3046–3052, May 2015. DOI: 10.1109/ICRA.2015.7139617.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *SciPy*, June 2010.

Cooper Bills, Joyce Chen, and Ashutosh Saxena. Autonomous mav flight in indoor environments using single image perspective cues. In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 5776–5783. IEEE, 2011.

Jonathan Binney, Andreas Krause, and Gaurav S Sukhatme. Optimizing waypoints for monitoring spatiotemporal phenomena. *The International Journal of Robotics Research*, 32(8):873–888, 2013.

Andreas Bircher, Kostas Alexis, Michael Burri, Philipp Oettershagen, Sammy Omari, Thomas Mantel, and Roland Siegwart. Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6423–6430. IEEE, 2015.

F. Blachini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

Frederic Bourgault, Tomonari Furukawa, and Hugh F Durrant-Whyte. Coordinated decentralized search for a lost target in a bayesian world. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 48–53. IEEE, 2003.

Michael S Branicky, Ross A Knepper, and James J Kuffner. Path and trajectory diversity: Theory and algorithms. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1359–1364. IEEE, 2008.

Markus Buchberger, K-W Jorg, and Ewald von Puttkamer. Laser-radar and sonar based world modeling and motion control for fast obstacle avoidance of the autonomous mobile robot MOBOT-IV. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 534–540. IEEE, 1993.

Alec Cameron and Hugh Durrant-Whyte. A bayesian approach to optimal sensor placement. *The International Journal of Robotics Research*, 9(5):70–88, 1990.

Benjamin Charrow, Sikang Liu, Vijay Kumar, and Nathan Michael. Information-theoretic mapping using cauchy-schwarz quadratic mutual information. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4791–4798. IEEE, 2015.

Chandra Chekuri and Martin Pal. A recursive greedy algorithm for walks in directed graphs. In *Foundations of Computer Science*. IEEE, 2005.

Chandra Chekuri, Nitish Korula, and Martin Pál. Improved algorithms for orienteering and related problems. *ACM Transactions on Algorithms (TALG)*, 8(3):23, 2012.

Chen CHEN, Shih-Fen Cheng, and Hoong Chuin Lau. Multi-agent orienteering problem with time-dependent capacity constraints. *Web Intelligence and Agent Systems*, 2014.

Min Chen, Emilio Frazzoli, David Hsu, and Wee Sun Lee. Pomdp-lite for robust robot planning under uncertainty. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 5427–5433. IEEE, 2016.

Shengyong Chen, Youfu Li, and Ngai Ming Kwok. Active vision in robotic systems: A survey of recent developments. *The International Journal of Robotics Research*, 30(11):1343–1377, 2011.

Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, DTIC Document, 1976a.

Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, DTIC Document, 1976b.

Debadeepta Dey, Tian Yu Liu, Boris Sofman, and Drew Bagnell. Efficient optimization of control libraries. Technical report, 2011.

Debadeepta Dey, Kumar Shaurya Shankar, Sam Zeng, Rupesh Mehta, M Talha Agcayazi, Christopher Eriksen, Shreyansh Daftry, Martial Hebert, and J Andrew Bagnell. Vision and learning for deliberative monocular cluttered flight. In *Field and Service Robotics*, pages 391–409. Springer, 2016.

J. J. Enright, E. Frazzoli, K. Savla, and F. Bullo. On multiple UAV routing with stochastic targets: Performance bounds and algorithms. In *Proceedings AIAA Guidance, Navigation, and Control Conference and Exhibit, Aug. 2005*.

Lawrence H Erickson and Steven M LaValle. Survivability: Measuring and ensuring path diversity. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2068–2073. IEEE, 2009.

William Feller. An introduction to probability theory and its applications. vol. i. 1950.

Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 1997.

T. Fraichard and H. Asama. Inevitable collision states. A step towards safer robots? *Advanced Robotics*, 18(10):1001–1024, 2004a.

Thierry Fraichard and Hajime Asama. Inevitable collision states—a step towards safer robots? *Advanced Robotics*, 18(10):1001–1024, 2004b.

E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *AIAA Journal of Guidance and Control*, 25(1):116–129, 2002.

Paul Furgale and Timothy D Barfoot. Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560, 2010.

Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. Bit*: Batch informed trees for optimal sampling-based planning via dynamic programming on implicit random geometric graphs. *CoRR*, abs/1405.5848, 2014. URL http://arxiv.org/abs/1405.5848.

C Goerzen and M Whalley. Minimal risk motion planning: a new planner for autonomous UAVs in uncertain environment. In *AHS International Specialists Meeting on Unmmaned Rotorcraft, Tempe, Arizona*, 2011.

Pascal Gohl, Dominik Honegger, Sammy Omari, Markus Achtelik, Marc Pollefeys, and Roland Siegwart. Omnidirectional visual obstacle detection using embedded FPGA. *IEEE International Conference on Intelligent Robots and Systems*, 2015-Decem:3938–3943, 2015. ISSN 21530866. DOI: 10.1109/IROS.2015.7353931.

Bruce L Golden, Larry Levy, and Rakesh Vohra. The orienteering problem. *Naval Research Logistics (NRL)*, 34(3):307–318, 1987.

Ruijie He, Emma Brunskill, and Nicholas Roy. Puma: Planning under uncertainty with macro-actions. In *AAAI*, 2010.

Lionel Heng, Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer, and Marc Pollefeys. Autonomous obstacle avoidance and maneuvering on a vision-guided mav using on-board processing. In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 2472–2477. IEEE, 2011.

Gabriel M Hoffmann, Steven L Waslander, and Claire J Tomlin. Quadrotor helicopter trajectory tracking control. In *AIAA guidance, navigation and control conference and exhibit*, pages 1–14, 2008.

Geoffrey Hollinger, Sanjiv Singh, Joseph Djugash, and Athanasios Kehagias. Efficient multi-robot search for a moving target. *The International Journal of Robotics Research*, 28(2):201–219, 2009.

Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3): 189–206, 2013. ISSN 1573-7527. DOI: 10.1007/s10514-012-9321-0. URL http://dx.doi.org/10.1007/s10514-012-9321-0.

Kaijen Hsiao, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Robust belief-based execution of manipulation programs. In *Eighth Intl. Workshop on the Algorithmic Foundations of Robotics*. Citeseer, 2008.

Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *International Symposium on Robotics Research (ISRR)*, pages 1–16, 2011.

Heather L Jones, Uland Wong, Kevin M Peterson, Jason Koenig, Aashish Sheshadri, and William L Red Whittaker. Complementary flyover and rover sensing for superior modeling of planetary features. In *Field and Service Robotics*, pages 415–429. Springer, 2014.

Brian J. Julian, Sertac Karaman, and Daniela Rus. On mutual information-based control of range sensing robots for mapping applications. *The International Journal of Robotics Research*, 33(10): 1375–1392, 2014. DOI: 10.1177/0278364914526288.

Andreas Krause. *Optimizing Sensing: Theory and Applications*. PhD thesis, Carnegie Mellon University, December 2008.

Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI*, volume 7, pages 1650–1654, 2007.

Zhan Wei Lim, David Hsu, and Wee Sun Lee. Adaptive informative path planning in metric spaces. *The International Journal of Robotics Research*, 35(5):585–598, 2016.

L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss. Stereo vision-based obstacle avoidance for micro air vehicles using disparity space. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3242–3249, May 2014. DOI: 10.1109/ICRA.2014.6907325.

H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained systems. *IEEE Transactions on Automatic Control*, 38(11): 1623–1633, 1993.

Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Chapman & Hall/CRC, 2010.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

Joelle Pineau, Geoff Gordon, Sebastian Thrun, et al. Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI*, volume 3, pages 1025–1032, 2003.

Allison Ryan and J Karl Hedrick. Particle filter based information-theoretic active sensing. *Robotics and Autonomous Systems*, 58(5): 574–584, 2010.

Sebastian Scherer, Lyle Chamberlain, and Sanjiv Singh. Autonomous landing at unprepared sites by a full-scale helicopter. *Robotics and Autonomous Systems*, 60(12):1545–1562, 2012a.

Sebastian Scherer, Joern Rehder, Supreeth Achar, Hugh Cover, Andrew Chambers, Stephen Nuske, and Sanjiv Singh. River mapping from a flying robot: state estimation, river detection, and obstacle mapping. *Autonomous Robots*, 33(1-2):189–214, 2012b.

Sebastian Scherer, Joern Rehder, Supreeth Achar, Hugh Cover, Andrew Chambers, Stephen Nuske, and Sanjiv Singh. River mapping from a flying robot: state estimation, river detection, and obstacle mapping. *Autonomous Robots*, 32(5):1–26, 2012c.

Uwe Schöning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 410–414. IEEE, 1999a.

Uwe Schöning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 410–414. IEEE, 1999b.

Tom Schouwenaars, Jonathan How, and Eric Feron. Receding horizon path planning with implicit safety guarantees. In *American Control Conference, 2004. Proceedings of the 2004*, volume 6, pages 5576–5581. IEEE, 2004.

Amarjeet Singh, Andreas Krause, Carlos Guestrin, and William J Kaiser. Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research*, pages 707–755, 2009.

Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. Technical report, 2007.

Sebastian Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2):111–127, 2003.

Pieter Vansteenwegen, Wouter Souffriau, and Dirk Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, 2011.

Abraham Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

David Wettergreen, Greydon Foil, Michael Furlong, and David R Thompson. Science autonomy for rover subsurface exploration of the atacama desert. *AI Magazine*, 35(4):47–60, 2014.

Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151. IEEE, 1997.

Luke Yoder and Sebastian Scherer. Autonomous Exploration for Infrastructure Modeling with a Micro Aerial Vehicle. In *FSR 2015*, 2015.

Luke Yoder and Sebastian Scherer. Autonomous exploration for infrastructure modeling with a micro aerial vehicle. In *Field and service robotics*, pages 427–440. Springer, 2016.

Jingjin Yu, Mac Schwager, and Daniela Rus. Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 342–349. IEEE, 2014.

Haifeng Zhang and Yevgeniy Vorobeychik. Submodular optimization with routing constraints. In *AAAI*, pages 819–826, 2016.