Information gain-based autonomous exploration of 3D environments using an unmanned aerial vehicle

Milas, Ana

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ana Milas

INFORMATION GAIN-BASED AUTONOMOUS EXPLORATION OF 3D ENVIRONMENTS USING AN UNMANNED AERIAL VEHICLE

DOCTORAL THESIS

Zagreb, 2024

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ana Milas

INFORMATION GAIN-BASED AUTONOMOUS EXPLORATION OF 3D ENVIRONMENTS USING AN UNMANNED AERIAL VEHICLE

DOCTORAL THESIS

Supervisor: Associate Professor Tamara Petrović, PhD

Zagreb, 2024

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Ana Milas

AUTONOMNO ISTRAŽIVANJE 3D PROSTORA S POMOĆU BESPILOTNE LETJELICE ZASNOVANO NA INFORMACIJSKOJ DOBITI

DOKTORSKI RAD

Mentor: izv. prof. dr. sc. Tamara Petrović

Zagreb, 2024.

Doctoral thesis is written at the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering.

Supervisor: Associate Professor Tamara Petrović, PhD

Doctoral thesis has 106 pages

Dissertation No.: _____

About the Supervisor

Tamara Petrović obtained her master's degree from the University of Zagreb Faculty of Electrical Engineering and Computing (UNIZG-FER) in 2007 and defended her Ph.D. thesis entitled "Centralized control of variable structure multi-vehicle systems based on resource allocation“ in 2014.

Since 2008 she has been employed at the UNIZG-FER at the Laboratory for Robotics and Intelligent Control Systems (LARICS) as a research fellow and since 2024 as an associate professor. During her studies, she was the recipient of the bronze plaque "Josip Lončar" (2007) and three "Josip Lončar" awards (2003, 2004, 2005) for outstanding academic success. She has participated in the work of several international and domestic scientific research projects, such as the Horizon Europe project AeroSTREAM, the H2020 project subCULTron and the FP7 project EC-SAFEMOBIL, in the field of multi-agent systems control. In 2013, as part of the ACROSS project, she was a visiting researcher at the Technical University of Crete, Greece. She published 2 book chapters, 15 journal papers, and 22 conference papers. She participates as a reviewer in several international scientific journals and conferences. Her main areas of scientific interest are multi-robot systems and discrete event systems. She participates in teaching activities at the UNIZG-FER, Department of Control and Computer Engineering, which include exercises and lectures, co-mentoring of students, and improvement of teaching content and materials.

Since the academic year of 2016/2017, she has been carrying out the duties of the Faculty Erasmus+ Coordinator for traineeships and the mobility of control engineering and automation students. She also co-coordinates the Erasmus Mundus study IFROS (Intelligent Field Robotic Systems).

O mentoru

Tamara Petrović diplomirala je na Sveučilištu u Zagrebu na Fakultetu elektrotehnike i računarstva (UNIZG-FER) 2007. godine, a doktorirala 2014. godine radom „Centralizirano upravljanje strukturno promjenjivim sustavima s više vozila zasnovano na pridjeljivanju resursa“.

Od 2008. godine zaposlena je na UNIZG-FER, u Laboratoriju za robotiku i inteligentne sustave upravljanja (LARICS) kao znanstveni novak, a od 2024. godine kao izvanredna profesorica. Tijekom studija bila je dobitnica brončane plakete Josip Lončar (2007.) te priznanja Josip Lončar (2003., 2004., 2005.) za izvrstan uspjeh tijekom studija. Sudjelovala je u radu više međunarodnih i domaćih znanstveno-istraživačkih projekata, pri čemu se mogu izdvojiti Horizon Europe projekt AeroSTREAM, H2020 projekt subCULTron i FP7 projekt EC-SAFEMOBIL, iz područja upravljanja multi-agentskim sustavima. Tijekom 2013. godine u sklopu projekta ACROSS kao gostujući istraživač boravila je na Tehničkom sveučilištu na Kreti, Grčka. Objavila je 2 poglavlja u knjizi, 15 znanstvenih radova u časopisima i 22 znanstvenih radova na međunarodnim konferencijama. Sudjeluje kao recenzent u više međunarodnih znanstvenih časopisa te konferencija. Područje znanstvenog interesa primarno obuhvaća upravljanje višerobotskim sustavima te autonomnog istraživanja prostora. Sudjeluje u nastavnim aktivnostima na UNIZG-FER na Zavodu za automatiku i računalno inženjerstvo, što uključuje održavanje vježbi i predavanja, mentoriranje studenata, unapređivanje nastavnog sadržaja i materijala.

Od ak. god. 2016./2017. obavlja dužnost Erasmus+ koordinatora UNIZG-FER za područje stručne prakse te studijskog boravka. Također su-koordinira Erasmus Mundus studij IFROS (Inteligentni terenski robotski sustavi).

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Assistant Professor Tamara, for her invaluable suggestions, ideas, help, and guidance throughout this journey. Being her first PhD student has been an honor and a unique learning experience that has significantly shaped my research and both personal and professional growth. I would also like to thank Professor Stjepan Bogdan for introducing me to robotics and recognizing my love for robotics and guiding me on this path that has brought so much joy and fulfillment to my life. His exemplary dedication to his field and insightful perspectives have profoundly influenced my academic goals and aspirations. Special thanks go to Profesor Zdenko Kovačić, whose unwavering support and optimism have been a constant source of encouragement.

I would like to express my gratitude to all the people in the LARICS. They are not only colleagues, but also friends. I have enjoyed being with Lovro and Ivo in Jazbina, with Antun, Jurica, Marijana, Robi and Jakob in Gaj, and with Filip, Marko, Antonella and Žorž in Vukovarska. They made my time as a doctoral student something special. A big thank you also goes to Antun (again, but he deserves it) and Frano, who gave me ideas, read papers and accompanied me on this journey.

I am very grateful to my friends outside of work for accompanying me on my journey, especially *nesparene čarape* Matea and Marina for listening to me the whole time and sharing both joy and frustration with me. I would also especially like to thank my *nuline* Nina, Maja and Ana for always supporting me and listening to my complaints. I would like to thank Martina, Mate, Sanjin and Tajana for being there for me and trying to keep track of the current status of my PhD.

A special thanks to my family, whose love and support have been my rock, even when they have asked too many times about my PhD completion. They cared wherever I am and whatever I do. I would like to thank my sisters Marija, Vinka and Daria for being with me through it all. Special thanks to Daria, who not only shared a room with me, but also my feelings and for always being there for me, together with my brother Toni. I hope that I am some kind of source of inspiration to them.

Last but not least, I would like to thank my best friend, husband and darling Krešo. I have no words to describe his unconditional love and support (even with my scientific work). Thank you for traveling with me and keeping me company late into the night as deadlines approach. You always push me to do my best no matter what I do. Thank you for all the cups of coffee.

Finally, I would like to end with the words that can describe my journey.

I know what it is to be in need, and I know what it is to have plenty. I have learned the secret of being content in any and every situation, whether well fed or hungry, whether living in plenty or in want. I can do all this through him who gives me strength. (Phil 4: 12-13)

Abstract

This thesis focuses on autonomous exploration using aerial vehicles. An autonomous exploration and mapping process is one of the fundamental tasks of robotics. Exploration methods can be used in both 2D and 3D space. In contrast to 2D exploration and mapping strategies, mapping large environments in 3D requires a large amount of memory and computational effort. Therefore the fastest possible generation of a complete 3D map and autonomous navigation of a robot through the map is a challenging task. This task is used in many applications, such as civil infrastructure or search and rescue scenarios. Additionally, a great challenge is running the exploration and mapping onboard an unmanned aerial vehicle (UAV).

The main objective of this thesis is to develop 3D autonomous exploration strategies capable of meeting the above challenges, and at the same time be adequate for both large and cluttered unknown environments. In those environments, a UAV should navigate autonomously without any a priori knowledge of the environment. The goal of autonomous exploration is to explore an unknown environment trying to optimize time, distance or energy consumption, to name a few. The environment is initially unknown and bounded. The exploration is considered done when the whole environment is explored and the map of the environment is created. The exploration strategies are designed with a focus on enabling a UAV to navigate autonomously and make real-time decisions.

As already mentioned, the primary objective of this research is to develop innovative methods for autonomous 3D exploration in unknown environments, specifically tailored to operate online and onboard robots with limited computational and energy resources. While current methods in the field show promise in scaling to various environment sizes and complexities, a significant gap remains in their applicability for 3D exploration, particularly when constrained to onboard processing capabilities. Addressing this gap, this research introduces three novel methods designed to efficiently manage the dense datasets produced by advanced sensors, applicable across diverse 3D environments, and executable in real time on robotic platforms.

The first method involves using submaps to find frontiers, along with refining those frontiers at multiple resolutions. This technique is engineered to identify and prioritize unexplored areas (frontiers) within a 3D space, adapting its resolution to effectively balance details of the environment and computational load. By dividing the environment into manageable submaps, the system can swiftly update and refine its understanding of the surroundings, guiding the robot to areas of interest.

The second method employs a sampling-based approach, leveraging a Recursive Shadowcasting algorithm. This algorithm is pivotal for the efficient estimation of information gain, a crucial factor in determining whether it is worth exploring a specific area. By recursively analyzing the environment for unobserved regions, this method directs the robot towards locations

that maximize the acquisition of new information, ensuring an effective exploration process.

In addition to these, semantic information about the environment is integrated into the exploration process. This integration enhances the understanding of the environment, allowing robots to make more informed decisions about where to explore next. The semantic layer adds context to the raw spatial data, enabling the robot to recognize and prioritize areas of potential interest or importance.

Each method focuses on rapidly calculating information gain and selecting the most beneficial next exploration goal, thereby optimizing the exploration process. The methods are not only novel in their approach but also in their efficiency, ensuring that even robots with limited resources can execute them effectively.

To verify the effectiveness and robustness of these methods, a series of tests are conducted, both in controlled simulations and in real-world scenarios. These include experiments in simulations to test specific functionalities, as well as in indoor and outdoor environments to test the performance of the system under more complex conditions. It is important to emphasize that the tests in real-world scenarios are carried out directly on the UAV with limited resources. The results of this thesis contribute to advancing the field of autonomous aerial exploration, offering robust, adaptable, and efficient methodologies for 3D mapping and exploration in a variety of challenging environments.

The contribution of this thesis consists of three main elements:

- A method for planning of autonomous 3D exploration based on multi-resolution frontier clustering
- A method for planning of autonomous 3D exploration based on Recursive Shadowcasting algorithm for information gain estimation
- A method for planning of autonomous 3D exploration based on semantic features of the environment

Keywords: Unmanned Aerial Vehicle, Autonomous Navigation, Mapping, Path Planning

Autonomno istraživanje 3D prostora s pomoću bespilotne letjelice zasnovano na informacijskoj dobiti

Ova disertacija fokusira se na autonomno istraživanje prostora pomoću zračnih bespilotnih letjelica. U sklopu ove disertacije, ciljana podskupina bespilotnih letjelica su višerotorske letjelice. Proces autonomnog istraživanja i kartiranja prostora jedan je od osnovnih zadataka robotike. Metode istraživanja prostora mogu se koristiti u dvodimenzionalnom (2D) i trodimenzionalnom (3D) prostoru. Za razliku od istraživanja prostora i strategija kartiranja prostora u 2D okruženju, izrada 3D karte prostora velikih okruženja zahtijeva veliku količinu memorije i računalnih resursa. Stoga je brza izrada potpune 3D karte prostora i autonomna navigacija robota kroz kartu izazovan zadatak. Ovaj zadatak koristi se u mnogim primjenama, kao u inspekciji infrastrukture ili u scenarijima potrage i spašavanja. Dodatni izazov je izvođenje autonomnog istraživanja i izgradnja karte prostora na bespilotnoj letjelici s ograničenim resursima.

Glavni cilj ove doktorske disertacije je razvoj strategija autonomnog 3D istraživanja prostora sposobnih za suočavanje s navedenim izazovima, koje su istovremeno prikladne za velika i složena okruženja. U takvim okruženjima, bespilotna letjelica bi se trebala navigirati autonomno ne znajući unaprijed informacije o okruženju. Cilj autonomnog istraživanja je istražiti nepoznato okruženje pokušavajući optimizirati na primjer vrijeme istraživanja, prijeđeni put ili potrošnju energije. Okruženje je na početku nepoznato, a veličina okruženja je unaprijed definirana. Istraživanje prostora smatra se potpunim u trenutku kada je istraženo cjelokupno okruženje i kada je izrađena karta prostora. Strategije istraživanja prostora implementirane su s ciljem da omoguće bespilotnoj letjelici autonomnu navigaciju i donošenje odluka u stvarnom vremenu.

Kao što je spomenuto, cilj ovog istraživanja je razvoj metoda za autonomno 3D istraživanje nepoznatih prostora koje može izvoditi robot s ograničenim resursima. Dosad razvijene metode pokazuju zadovoljavajuće performanse s obzirom na zadanu veličinu prostora te mogu istražiti kako okruženja velikih tako i malih dimenzija. Međutim, neke od njih nisu prikladne za istraživanje 3D prostora. Može se reći da dosad razvijene metode u ovom području pokazuju potencijal te se odlikuju skalabilnošću. To je posebno istaknuto u prilagodbi istih složenim i velikim prostorima. Međutim, značajan nedostatak je u primjeni postojećih metoda u istraživanju 3D prostora u stvarnom vremenu koristeći bespilotne letjelice. Stoga se predlažu tri metode za autonomno istraživanje prostora koje mogu obrađivati guste i velike skupove podatakaa, koje su prikladne za sve vrste 3D prostora i mogu se izvršavati na sklopovlju robota. Metode uključuju otkrivanje fronte iz podkarte prostora, filtriranje fronte koristeći višerezolucijsku metodu, pristup temeljen na uzorkovanju koji se oslanja na rekurzivni algoritam bacanja sjene za učinkovitu procjenu informacijske dobiti te integraciju semantičkih značajki prostora u proces istraživanja prostora. Predložene metode postupno istražuju prostor uz računanje in-

formacijske dobiti i planiranje putanje. Metoda temeljena na fronti kao i metoda uzorkovanja temelje se na novoj i učinkovitoj procjeni informacijske dobiti i odabiru sljedeće najbolje ciljne točke. Metoda za autonomno istraživanje prostora korištenjem semantičkih značajki okoline integrirana je u metodu temeljenu na fronti čime je postignuta pristranost istraživanja područja u blizini objekata od interesa.

Metode su detaljno opisane u nastavku.

Prva metoda temelji se na detekciji fronte na osnovu podkarte prostora u kombinaciji s filtriranjem točaka fronte višerezolucijskim pristupom. Ova tehnika osmišljena je za identificiranje granice istraženog i neistraženog 3D prostora (fronte) i prilagodbu rezolucije točaka fronte za učinkovitu ravnotežu između razine detalja objekata u okolini prikazanih u karti prostora i računalnog opterećenja. Dijeljenjem okruženja na podkarte željene veličine, sustav može brzo osvježiti cjelokupnu kartu prostora te time ubrzati detekciju fronte.

Druga metoda koristi pristup temeljen na uzorkovanju u kojem se pomoću algoritma brzorastućih slučajnih stabala generiraju čvorovi u 3D prostoru. Generirani čvorovi evaluiraju se koristeći rekurzivni algoritam bacanja sjene. Ovaj algoritam ključan je za učinkovitu procjenu informacijske dobiti. Predložena metoda usmjerava robota prema pozicijama koje maksimiziraju otkrivanje novog prostora.

Treća metoda temelji se na integraciji semantičkih značajki okoline u proces istraživanja prostora. Predložena integracija osigurava razumijevanje značajki iz okoline, omogućujući robotu da donose informiranije odluke o tome kamo se dalje kretati dok istražuje prostor. Drugim riječima, semantički sloj dodaje kontekst geometrijskim podatcima prostora zapisanim u karti prostora. Na taj način daje se prednost područjima u čijoj se blizini nalaze objekti od interesa.

Sve metode u središte pozornosti stavljaju brzo izračunavanje informacijske dobiti i odabir najbolje sljedeće ciljne točke, optimizirajući time proces istraživanja prostora. Metode su inovativne te osiguravaju učinkovito istraživanje 3D prostora robotima s ograničenim resursima. Kako bi se potvrdila učinkovitost i robustnost ovih metoda, provodi se niz ispitivanja, kako u kontroliranom simulacijskom okruženju tako i u stvarnim okruženjima. To uključuje ispitivanje u simulacijskom okruženju za testiranje specifičnih funkcionalnosti i u vanjskim ili unutarnjim stvarnim okruženjima za procjenu izvedbe sustava u složenijim uvjetima. Testiranje strategija na stvarnom robotu s ograničenim resursima je u isto vrijeme jedan od najvažnijih dijelova disertacije. Rezultati ove disertacije doprinose napretku u području autonomnog zračnog istraživanja prostora, nudeći robustne, prilagodljive i učinkovite metode za 3D kartiranje i istraživanje prostora u različitim izazovnim okruženjima.

Doktorska disertacija sastoji se od osam poglavlja.

U prvom poglavlju opisani su autonomni sustavi u području robotike te primjena istih na stvarnim primjerima. Ovo poglavlje ujedno najavljuje temu doktorske disertacije, odnosno

autonomno istraživanje pomoću bespilotne letjelice. Opisani su osnovni pojmovi kao što su lokalizacija, kartiranje prostora, istraživanje prostora i planiranje putanje. Uz to, predstavljeni su izazovi koje ova tema uključuje, motivacija i ciljevi doktorske disertacije. Ukratko su opisani znanstveni doprinosi doktorske disertacije i dan je kratki pregled strukture rada.

U drugom poglavlju objašnjeni su osnovni koncepti i tehnike korištene u implementaciji metoda za autonomno istraživanje prostora. Opisana je arhitektura robota korištenog u doktorskoj disertaciji, senzori kao što su LiDAR i RGB-D kamera i načini prikazivanja 3D prostora (kartiranje prostora). Osim toga, dan je uvod u relevantne koncepte u teoriji informacija (poput entropije i izračuna informacijske dobiti), te pregled metoda za planiranje putnje i autonomnu navigaciju. Objašnjeni su i osnovni algoritmi računanja informacijske dobiti.

U trećem poglavlju prikazan je pregled područja od interesa za ovu disertaciju. Istraživanje prostora razloženo je na metode temeljene na fronti, metode temeljene na uzorkovanju i metode temeljene na semantičkim značajkama okoline.

U četvrtom poglavlju dan je opis metoda autonomnog istraživanja prostora u 3D okruženju. Na samom početku predstavljen je sustav autonomnog istraživanja prostora sa svojim komponentama. Najveći fokus stavljen je na strategije istraživanja prostora. Stoga, u poglavlju su predstavljene tri strategije autonomnog istraživanja prostora koje su ujedno i znanstveni doprinosi ovoga rada. Strategije su najavljene prethodno u ovom poglavlju. Za svaku strategiju objašnjeni su osnovni pojmovi i detaljnije je opisan znanstveni doprinos.

U petom poglavlju dan je zaključak ove disertacije koji sažima znanstvene doprinose i cjelokupni rad. Zaključak obuhvaća raspravu o rezultatima istraživanja i predlaže smjernice za buduće istraživanje. Ovaj dio objedinjuje doprinose doktorske disertacije i prikazuje njezin značaj u širem kontekstu robotike.

U šestom poglavlju dan je popis objavljenih radova koji čine disertaciju.

U sedmom poglavlju opisan je doprinos autora na svakome od objavljenih radova.

U osmom poglavlju priloženi su radovi na kojima se disertacija zasniva.

Nakon toga izložen je popis literature korištene u disertaciji. Disertacija je izrađena po skandinavskom modelu te ju čine tri časopisna članka. Glavni doprinosi disertacije su izloženi i opisani u nastavku poglavlja.

Tri su glavna doprinosa ove disertacije:

- **Metoda za planiranje autonomnog 3D istraživanja temeljena na grupiranju točaka fronte koristeći višerezolucijski pristup**

  Metoda je zasnovana na detekciji točaka fronte, točaka na granici istraženog i neistraženog prostora. Jedna od većih prednosti razvijene metode je skalabilnost. Metoda zahtijeva manje vremena obrade za istu veličinu prostora, dok održava usporedivo vrijeme istraživanja s postojećim metodama. Ovo poboljšano izvođenje, između ostalog,

postiže se korištenjem ulaznih podataka iz algoritma za istovremenu lokalizaciju i kartiranje prostora, umjesto izravno iz 3D senzora. Rad uključuje znanstvene doprinose kao što su: razvoj metode za detekciju fronte zasnovane na podkartama, implementacija višerezolucijskog pristupa za poboljšanu detekciju fronte, te razvoj algoritma za odabir najbolje točke fronte uključujući aproksimaciju senzora na bazi kocke za brzi izračun informacijske dobiti. U usporedbi s trenutno razvijenim metodama, ovaj pristup nudi brže i učinkovitije izvođenje kako u računalnom vremenu tako i cjelokupnom vremenu potrebnom za istraživanje prostora. Performanse predloženog pristupa potvrđene su kroz detaljno simulacijsko i eksperimentalno ispitivanje.

- **Metoda za planiranje autonomnog 3D istraživanja temeljena na rekurzivnom algoritmu bacanja sjene za procjenu informacijske dobiti**
  Metoda se temelji na nasumičnom uzorkovanju točaka, potencijalnih ciljnih točaka istraživanja prostora i evaluaciji istih u svrhu bržeg istraživanja prostora. Točke su nasumično generirane korištenjem algoritma brzorastućih slučajnih stabala. Evaluacija točaka je implementirana pomoću nove i učinkovite metode za izračunavanje informacijske dobiti, temeljene na algoritmu rekurzivnog bacanja sjene. Za određivanje sljedeće najbolje ciljne točke koristi se metoda evaluacije putanje zasnovana na kvadrima, što značajno smanjuje vrijeme računanja informacijske dobiti cijele putanje. Osim toga, metoda uključuje strategiju za rješavanje zaglavljenja, povećavajući sposobnost brzog oporavka robota u složenim okruženjima, čime se minimizira ukupno vrijeme istraživanja. Usporedna ispitivanja u simulacijama pokazuju da ovaj pristup nadmašuje trenutno razvijene metode u vidu računalne učinkovitosti i ukupnog vremena istraživanja. Doprinosi rada su: razvoj algoritma za procjenu informacijske dobiti temeljenog na algoritmu rekurzivnog bacanja sjene, implementacija metode zasnovane na kvadrima za procjenu informacijske dobiti za svaku vezu brzorastućeg slučajnog stabla, te metoda praćenja posjećenih čvorova u svrhu rješavanja zaglavljenja u složenim prostorima tijekom autonomnog istraživanja.

- **Metoda za planiranje autonomnog 3D istraživanja temeljena na semantičkim značajkama okoline**
  Metoda se temelji na razvoju strategije istraživanja prostora koja integrira metodu zasnovanu na fronti sa semantičkim informacijama u prostoru, omogućavajući iterativno istraživanje i označavanje objekata od interesa u 3D karti prostora. Metoda je kombinirana s metodom temeljnom na fronti. Ukupna informacijska dobit pojedine točke sastoji se od geometrijskih podataka o točki u prostoru i semantičkih informacija dobivenih iz algoritma za semantičku segmentaciju objekata. Glavni doprinos uključuje nadogradnju funkcije informacijske dobiti kako bi obuhvatila i geometrijske i semantičke informacije o prostoru. Implementirano je označavanje 3D objekata u stvarnom vremenu tijekom

istraživanja prostora, što uključuje izdvajanje semantički segmentiranih objekata iz 2D slike i obradu oblaka točaka iz kamere kako bi se odredili položaji objekata u okruženju. Sustav se sastoji od kartiranja prostora, istraživanja prostora, planiranja putanje i autonomne navigacije u prostoru. Sustav je modularan i prilagodljiv novim okruženjima, te je ispitan na jednostavnom sklopovlju s pristupačnim sezorima (RGB-D kamerom) na bespilotnoj letjelici s ograničenim resursima.

Ključne riječi: bespilotna letjelica, autonomna navigacija, kartiranje prostora, planiranje putanje

Contents

1. Introduction 1
- 1.1. Challenges 4
- 1.2. Motivation and Objectives 5
- 1.3. Problem Formulation 8
- 1.4. Hypotheses 9
- 1.5. Contribution 10
- 1.6. Thesis Outline 10

2. Preliminary 12
- 2.1. Robot Architecture 12
- 2.2. LiDAR and RGB-D Sensors 13
- 2.3. 3D Map Representation 14
- 2.4. Information Theory 17
  - 2.4.1. Entropy 17
  - 2.4.2. Information Gain Calculation 18
  - 2.4.3. Raycasting Algorithm 20
  - 2.4.4. Recursive Shadowcasting Algorithm 21
- 2.5. Path Planning and Navigation 22

3. State-of-the-Art 24
- 3.1. Frontier-Based Exploration 25
- 3.2. Sampling-Based Exploration 26
- 3.3. Semantically-Enhanced Exploration 27
  - 3.3.1. Semantic Image Segmentation 28

4. Autonomous Exploration in 3D Environments 30
- 4.1. System Overview 31

4.2. Frontier-Based Exploration 31
4.2.1. Frontier Definition and Detection 33
4.2.2. Frontier Clustering 35
4.2.3. Frontier Evaluation and Selection 37
4.3. Sampling-Based Exploration 38
4.3.1. Potential Exploration Targets Generation 39
4.3.2. Path Evaluation and Selection 40
4.4. Semantically-Enhanced Exploration 43
4.4.1. 2D Image-Based Sematic Segmentation 44
4.4.2. Labeling Objects in the Map 45
4.4.3. Sematically-Aware Frontier Evaluation 46

5. Conclusion 49

6. List of Publications 52

7. Author's Contributions to the Publications 53

8. Publications 57
Publication 1: A Multi-Resolution Frontier-Based Planner for Autonomous 3D Exploration 57
Publication 2: A Shadowcasting-Based Next-Best-View Planner for Autonomous 3D Exploration 66
Publication 3: ASEP: An Autonomous Semantic Exploration Planner with Object Labeling 75

Bibliography 91

Biography 101

Full List of Publications 103

Životopis 105

CHAPTER 1

Introduction

Autonomous systems in aerial robotics, particularly unmanned aerial vehicles (UAVs), are revolutionizing a number of industries with their ability to operate independently of direct human control. Equipped with advanced sensors and sophisticated algorithms, these aerial robots are ideal for tasks such as agricultural surveillance, search and rescue operations, environmental monitoring and infrastructure inspection. Their flying capability along with computer hardware that enables the execution of algorithms, allows them to navigate complex environments and make real-time decisions on obstacle avoidance and path planning. This technology not only increases efficiency and precision in various applications, but also opens up new possibilities in areas where human access is restricted or dangerous. The transition from basic autonomous functions to more complex tasks and advanced applications, where UAVs are deployed in unknown or inaccessible areas, is a challenging endeavour. In such scenarios, the UAVs utilize navigation capabilities and autonomous decision-making processes to perform comprehensive data collection or map creation.

The navigation and decision-making capabilities create a process of *autonomous exploration*. Autonomous exploration refers to the ability of a robot or unmanned vehicle to navigate through an unknown environment and gather information without human intervention. In general, autonomous exploration consists of four core components: *localization*, *mapping*, *exploration* and *path planning*.

Localization in robotics is the process by which a robot determines its position and orientation within its environment. External positioning systems, such as Global Positioning System (GPS), can be utilized for this purpose; however, depending on the task, such systems might not be viable. For example, in GPS-denied environments or if high accuracy or update rate is required. For a robot to function in any scenario, it is therefore desirable that it can localize using

only onboard sensors and computation. This capability is essential for robots to autonomously navigate and interact effectively with their surroundings. It relies on data from various sensors like cameras, Light Detection and Ranging (LiDAR), Inertial Measurement Unit (IMU), and wheel encoders, which provide information about the surroundings of the robot and its motion. A map of the environment serves as a reference for localization and can be either pre-existing or generated by the robot through Simultaneous Localization and Mapping (SLAM). There are different types of localization, including global localization where the robot estimates its pose without prior knowledge of its initial position, and local localization which refines the pose estimate when an approximate location is known.

Mapping is the process of creating a detailed representation, a map, of the environment from sensor data. This map can incorporate various details that assist the robot in understanding and navigating its surroundings. Commonly included are the locations of physical barriers like walls, objects, and trees, which are crucial for localization and collision avoidance. Additionally, the map can contain diverse data such as temperature readings at specific points, the positions of various items like chairs, doors, vehicles, and the presence of different colors. The ability of a robot to autonomously create such a map significantly enhances its functionality, particularly in environments lacking pre-existing information.

Exploration uses decision-making algorithms to enable a robot to automatically acquire a map of an environment without any a priori information, i.e., when the environment is fully or partially unknown. It involves both decision-making and mapping. The mapping part of exploration accumulates the information gathered by the sensors and the decision-making part deals with where and how the environment should be traversed to uncover it. A common formulation for the decision-making objective is to move the robot such that the unknown parts of the map of the environment are removed. Depending on the task in which exploration is performed, the goal of exploration might be to perform, for example, complete 3D reconstruction or search and rescue. This influences the decision-making strategy.

Path planning is an important aspect of autonomous exploration, involving the determination of a sequence of actions for a robot to reach a goal location while avoiding obstacles. In

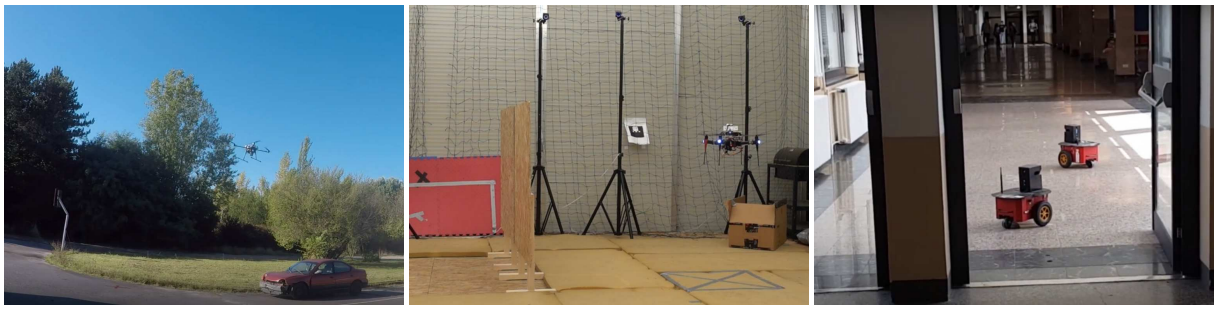


Figure 1.1: Examples of some of the aerial and ground robots autonomously executing exploration of outdoor and indoor environments.

Figure 1.2: UAVs used in outdoor and indoor experiments. On the left is a *Kopterworx* quadcopter, a custom built quadcopter equipped with a Velodyne $VLP-16$ LiDAR sensor. On the right is a *Hexsoon EDU-450* quadcopter equipped with a Realsense $D455$ camera.

other words, path planning involves the robot calculating a path from its current position to its destination, taking into account its understanding of the environment and sensor inputs. It uses map of the environment to plan safe and obstacle-free paths. Optimality criteria, such as shortest distance, quickest time, or energy efficiency, are determined based on the requirements of the application.

Autonomous exploration can be performed by different types of robots, as shown in Fig. 1.1, in various environments and with single or multiple robots. In this thesis, the focus is on aerial robot exploration in static outdoor and indoor environments with a single robot. Autonomous exploration using UAVs has been an active area of research in robotics for several years. It is a rapidly growing field of research that has the potential to revolutionize various industries. The use of UAVs for exploration has several advantages, including the ability to cover large areas quickly and the ability to access areas that are difficult to reach by ground vehicles.

Usually, autonomous exploration is done without any a priori knowledge of the environment. UAVs use various sensors, including LiDAR sensors and RGB-D cameras, to perceive information from the environment and allow the UAV to navigate through unknown environments and gather information autonomously. During exploration, sensors are also used to evaluate the environment in which the robot is operating and allow the robot to adjust following actions based on collected data. The system proposed in the thesis utilizes the capabilities of UAVs equipped with advanced sensors such as LiDARs or cameras, as shown in Fig. 1.2. This system is specifically designed to handle the complexity of navigating and mapping unexplored and unstructured environments. The proposed system is expected to have applications in various fields. These include search and rescue missions where rapid and efficient location of people is crucial, surveillance missions where an extensive and detailed area needs to be monitored, and inspection tasks, particularly in dangerous or inaccessible environments.

1.1 Challenges

Autonomous exploration using UAVs is a highly challenging task for a number of reasons. Here are some of the main challenges when doing autonomous exploration with UAVs.
Challenge 1: Unknown environments. Initially, environments are unknown and during autonomous exploration, a robot can not assume prior knowledge of its surroundings. As a consequence, it is required to plan based on the limited information available at each time step. Furthermore, all systems need to be able to operate in diverse and potentially unforeseen environments.

Challenge 2: Time efficient exploration. The goal is to achieve efficiency regarding time to completion for exploration. Both the exploration paradigm, i.e., defining how long an iteration should be, and the exploration heuristic, i.e., determining what action to take, directly affect the exploration performance. If an iteration is too long, the robot may be idle for a moment, or if too short, the computational cost may be too high – it is therefore important to strike a balance. The exploration heuristic should aim to be optimal in the global sense although exploration operates either in frontier-based or sampling-based fashion. Furthermore, UAVs have limited battery life, which restricts the amount of time they can spend exploring an environment. This can be a significant challenge when exploring large or complex environments, as the UAV may need to return to its base station for recharging.

Challenge 3: Scalability with respect to the size of the environment. The computational time plays an important role in the autonomous exploration process. It influences on total exploration time and thus performance in general. The extraction of potential waypoints and information gain calculation should be adjusted to run in real time but also to be able to explore both small and large environments.

Challenge 4: Modular exploration system: The exploration planner should fit to different localization, control and path planning algorithms. It should not be restricted to a specific module.

Challenge 5: Adaptation to sensors and mobile platforms. The exploration algorithm should be adaptable to different sensors, as far as they produce input data suitable for map creation. Exploration planners should apply to various types of autonomous robots equipped with LiDARs or other sensors that can be used to build a map.

Challenge 6: Online operation on robots. To operate on a fully autonomous system, the methods used should run on the limited onboard computation available. This is related to the computational time of each component in the exploration system. Similarly, all methods should

function incrementally as the robot moves and perceives more of its surroundings, and in real-time for an interactive system to operate.

Challenge 7: Onboard computational capabilities. UAVs have limited onboard computational capabilities, which can make it challenging to process the data collected by sensors such as LiDARs and cameras. This can lead to delays in processing and decision-making, which can impact the efficiency of the exploration.

Challenge 8: Safe navigation with obstacle avoidance. One of the main reasons why a mapping system should account for all these challenges is to ensure the safety of the robot. UAVs need to be able to identify and avoid obstacles in the environment to ensure safe and efficient exploration. This requires the development of sophisticated sensory systems and algorithms that can detect and avoid obstacles in real time by planning safe paths.

These challenges summarize the issues that one is usually confronted with in autonomous systems. Often, it is needed to make a trade-off between the challenges and choose some that are more important than others. However, they should all be achieved to enable fully autonomous 3D exploration both in simulation and in the real world. The system proposed in the thesis aims to overcome all these challenges by using different input data, scenarios, sensors, UAV platforms, localization algorithms, exploration strategies, path planning and control algorithms to achieve efficient and complete coverage of the environment.

1.2 Motivation and Objectives

The ability to autonomously explore unknown environments is a crucial component to robot autonomy and a prerequisite for a wide range of applications, ranging from consumer and service robotics to industrial inspection, tunnel mapping, warehousing as well as search and rescue. In many of these applications, the goal of exploration is to actively map an unknown environment, such that it is fully covered as fast as possible. Autonomous mapping and exploration have received great research interest, and numerous methods have been proposed. The motivation for doing autonomous exploration with UAVs is deeply rooted in the desire to enhance the scope and effectiveness of exploration in diverse and often challenging environments. UAVs enable exploration in areas that are either too dangerous, remote, or inaccessible for humans, such as disaster zones, dense forests, and rugged terrain. The UAVs in autonomous exploration can safely and efficiently navigate these environments, providing critical data and insights that would be otherwise difficult to obtain. However, to enable the application of such systems in real-world environments, all of the challenges pointed out in Section 1.1 need to be addressed. While great progress has been made by the community, some of them are not focused on apply-

ing the solution in the real world, and some of them still have several limitations.

The main goal of this research is to develop methods for autonomous 3D exploration of unknown environments that can be performed online and onboard a robot with limited resources. Current methods show good scaling characteristics and the ability to explore both large and small environments. However, most of them are not suitable for 3D exploration and onboard processing. Therefore, proposed methods for autonomous exploration can handle dense datasets generated by state-of-the-art sensors, that are suitable for all types of 3D environments and can be executed online and onboard a robot. The implemented methods cover three different exploration strategies, frontier-based, sampling-based and semantically-enhanced strategy, which will be explained in detail in Chapter 4. The methods include submap-based frontier detection and multi-resolution frontier refinement, a sampling-based approach that relies on a Recursive Shadowcasting algorithm for efficient information gain estimation and the integration of semantic map information into the exploration process. The proposed methods gradually explore an area while planning paths and calculating the information gain in a short computational time. Both frontier-based and sampling-based strategies rely on novel and efficient calculations of information gain and selection of the next best goal. The method for autonomous exploration using semantic information of the environment integrates frontier-based exploration and enables directing to the objects of interest to achieve efficient object labeling in 3D exploration. The system is validated in several realistic and challenging simulation experiments as well as in real-world experiments.

The frontier-based exploration method utilizes a multi-resolution frontier planner that deals with large environments and large input data, trying to reduce computational efforts. It is adequate for all types of the environment, especially for large environments. The approach was inspired by that of Zhu et al. [1], an exploration tool called 3D-FBET. This is a frontier-based tool that is performed in three phases, similar to those presented in this thesis. The phases are 3D mapping, frontier detection in combination with a clustering algorithm, and the selection of the best frontier. Through experimental evaluation on different environments, 3D-FBET showed several shortcomings. First, because the frontier detection is based on a subset of altered voxels (generated from the camera point cloud), which is highly variable, the obtained frontiers were noisy and unreliable. Furthermore, the resulting frontier presented only a local view and the clustering was not adapted to the environment. These problems led to a higher total exploration time. The authors provide the source code and the duration analysis for each phase, which facilitates comparison with the new approaches. This approach is extended to recognize not only local but also global frontiers, similar to Mannucci et al. [2]. Mannucci proposed a 3D exploration with two OctoMaps and two frontiers (local and global) with different resolutions. Global frontiers are assigned when the set of local frontiers is empty. Manucci evaluates the best frontier using a cost-utility approach, similar to [3]. Since maintaining two OctoMaps

is a resource-intensive task, the properties of OctoMaps are used and a solution with multiple resolutions in a single OctoMap is implemented. An advantage of an OctoMap is also used in [4] to merge voxels with an equal state, which results in the reduction of obstacle detection calculations. The 3D frontier detection is motivated by a dense 2D frontier method presented by Orsulic et al. [5], which has achieved good results in terms of wall time per frontier update. Together with multi-resolution clustering and appropriate target point selection, a novel 3D exploration planner that accelerates the 3D exploration process is constructed.

Most related works design the exploration algorithm to minimize the total exploration time ([2], [6], [7]). Some of them take into consideration computation time [1], which plays an important role in an exploration process, such that a lower computation time as well as the next best goal selection algorithm lead to a lower total exploration time. However, both the computation time and the total exploration time are considered. The planner is able to run online and on board a robot with limited resources. Results are shown in simulations and experiments while datasets are provided for further use.

Sampling-based methods have gained considerable traction due to their efficiency in navigating complex environments and their ability to compute volumetric information gain from different waypoints. These capabilities facilitate effective local exploration planning and allow these methods to tailor their approach to different targets [6, 8]. Despite these advantages, these methods have few limitations. Major challenges include determining optimal sampling locations, accurately calculating information gains, evaluating path costs, and effectively integrating these factors. They often rely on heuristic approaches that require extensive tuning, which can limit overall performance. Furthermore, when planning in unknown environments, these methods must rely on the limited information available at each time step. This leads to a reliance on assumptions, such as the predictability that unknown areas are observable, which can affect the efficiency and accuracy of the exploration process. Despite the limits outlined, the considerable potential of the sampling-based method for exploring both large and small cluttered environments is recognized. This approach allows for a reasonable number of waypoints to be evaluated, making it feasible to compute the next best goal in real time. To avoid the time-consuming nature typically associated with such methods, an innovative Recursive Shadowcasting algorithm for calculating information gain has been implemented, instead of the more traditional raycasting technique. This adjustment significantly enhances efficiency. Moreover, by evaluating entire paths rather than just individual sampling points within the environment, the limitations imposed by isolated sampling points are effectively bypassed, thereby accelerating the exploration process.

The integration of semantically enhanced exploration in this approach offers a significant advance in understanding and interacting with the environment. By integrating detailed semantic information, the exploration process can be controlled more precisely. This method uses a

3D segmentation or detection model that is capable of running onboard the robot despite its limited computational resources. The task of this model is to identify and categorize objects in the environment, which are then strategically integrated into the exploration planning. As a result, a map is obtained, that not only outlines the physical layout of the environment, but also identifies objects of interest within the area. The motivation for using a semantically-enhanced strategy in exploration is to add a layer of contextual understanding that goes beyond mere spatial awareness. By identifying specific objects and their properties, the robot can make more informed decisions about where it should be and which areas might be particularly relevant or interesting. This is particularly important in environments where specific objects or features are part of the mission objectives, such as search and rescue missions, where the identification of human figures is of paramount importance. The semantic layer therefore not only increases the efficiency of the exploration by directing the focus where it is most needed, but it also enriches the quality of the data collected and provides a more comprehensive understanding of the explored environment.

1.3 Problem Formulation

In this thesis, the problem of a UAV attempting to explore a previously unknown environment is solved. The UAV has at least one 3D imaging sensor, either LiDAR or RGB-D, with a finite resolution and a fixed horizontal and vertical field of view (FOV), mounted in a fixed position. It is assumed that the UAV is building a map of the environment from this sensor as it navigates. Additionally, it is assumed that the environment is static and does not change over time.

A modular exploration planner is designed, which consists of localization, mapping, exploration and path planning modules. It is assumed that the localization module provides accurate position and the drift over time is not considered in other modules. The map of the environment should be capable of integrating various types of data, including geometric and semantic information, to provide a comprehensive understanding of the environment. The efficiency of the mapping process, its accuracy, and its ability to function with limited computational resources are key factors in its effectiveness. Within the exploration module, it is distinguished between waypoint generation and best waypoint calculation. Depending on the strategy (frontier-based or sampling-based), waypoint generation identity boundaries between explored and unexplored areas (frontiers) or samples waypoints in the environment, respectively. The primary challenge here is to develop an algorithm that can quickly and accurately generate waypoints in real-time, even in complex and challenging environments. This requires the algorithm to be adaptive, and capable of handling varying terrain. Once waypoints are generated or detected, the next step is to calculate the best waypoint to explore next. This involves assessing each waypoint based on a set of criteria such as distance, ease of access and potential information gain. The chal-

lenge in this stage is to develop a decision-making process that not only evaluates these factors effectively but also does so in a manner that optimizes the overall exploration strategy. This process should take into account the current state of the map and the capabilities of the robot. Path planning module treats unknown space as occupied and inaccessible. The core problem to address is the development of a complementary exploration algorithm for use in different environments. All parts of the method should be fast enough to be executed online and in real time and onboard the UAV.

The goal of autonomous exploration is to explore an unknown environment while trying to optimize time, distance or energy consumption, to name a few. The environment is initially unknown and bounded. The exploration is considered done when the whole environment is explored and the map of the environment is created.

1.4 Hypotheses

The problems and challenges considered within this thesis arising in the attempt of autonomous exploration of unknown environments are investigated experimentally, defining the hypotheses of the proposed scientific research as follows:

1. It is possible to improve frontier detection using an OctoMap created from Google Cartographer SLAM submaps
2. By using hierarchical multi-resolution frontier refinement, it is possible to obtain faster frontier points clustering
3. It is possible to estimate the information gain and improve the information gain calculation using the Recursive Shadowcasting algorithm
4. Using a history tracking method to resolve dead end states during exploration improves the overall time to explore the unknown environment
5. It is possible to evaluate exploration algorithms both in a simulation environment and in the real world
6. Using a semantic segmentation algorithm to extract semantic features from the environment into the map, it is possible to extend the proposed exploration methods and achieve more efficient exploration

These hypotheses are formulated to address specific challenges in autonomous exploration and are intended to guide the development and validation of novel methods and algorithms. At the same time, these hypotheses aim to contribute to the advancement of autonomous systems in various domains and scenarios that rely on static or manual procedures and conventional platforms that can be unreliable, costly or even dangerous, and where autonomous UAVs save time, cost and labor.

1.5 Contribution

The contribution of this thesis consists of three main elements:

1. A method for planning of autonomous 3D exploration based on multi-resolution frontier clustering
2. A method for planning of autonomous 3D exploration based on Recursive Shadowcasting algorithm for information gain estimation
3. A method for planning of autonomous 3D exploration based on semantic features of the environment

1.6 Thesis Outline

This thesis is organized into eight chapters, discussing the existing state of the art and the contribution of this thesis. The summary of the following chapters is as follows:

Chapter 2: Within this chapter the basic technologies are overviewed, such as robot architecture, sensors and map representations. Furthermore, an introduction to information theory is given. This is followed by the description of the path planning and navigation algorithm.

Chapter 3: This chapter presents the current state of scientific knowledge about autonomous exploration methods, their development over time, their specific applications and the various ways in which they have been adapted and improved in recent research.

Chapter 4: This chapter is the core of the thesis and discusses the contribution of this thesis related to autonomous exploration for robotic applications in outdoor and indoor environments. It introduces frontier-based, sampling-based and semantically-enhanced exploration strategies. The elements of the scientific contribution to the methods for planning autonomous 3D exploration based on multi-resolution frontier clustering and the Recursive Shadowcasting algorithm for estimating information gain are discussed, as well as the incorporation of semantic data into the exploration. Furthermore, the basic concepts for each strategy are discussed, with a focus on the contribution compared to state-of-the-art methods.

Chapter 5: This chapter presents concluding remarks of the thesis, summarizing the key elements of the contribution and implications of the research. It reflects on the objectives set out at the beginning of the thesis, discussing how they were addressed through the research and highlighting the significant advancements made in the field of autonomous exploration with UAVs. Furthermore, this chapter outlines potential future research directions, identifying areas where

further investigation could lead to continued development and refinement of the methodologies and technologies explored in this thesis.

Chapter 6: This chapter lists all publications contributing to the main results of the thesis.

Chapter 7: This chapter states the author's contribution to each of the included publications.

Chapter 8: This chapter includes full versions of publications published in peer-reviewed journals contributing to the main results of the thesis.

A list of referenced bibliographies is given following the main body of this thesis. Finally, a short biography of the author followed by a complete list of publications is given in the end.

CHAPTER 2

Preliminary

According to Yamauchi et al. [9], the central question in exploration is: *Given what you know about the world, where should you move to gain as much new information as possible?* At the start of an exploration, the only available information about the environment is what you can see from where you are standing. The task of exploration is then to create a map that describes as much of the world as possible in a reasonable amount of time. This chapter describes the UAV system used for autonomous exploration and the sensors attached to the UAV for data collection. It also presents the mapping techniques and the key component of the central question in the exploration, namely the methods for calculating the information gain used in decision making during the exploration.

2.1 Robot Architecture

In this thesis, the exploration is performed with a UAV that has no prior knowledge of the environment. More specifically, a quadcopter or quadrotor is used for autonomous exploration. It is a multirotor helicopter that has four vertically oriented propelled motors which are placed in a square formation with an equal distance from the quadcopter center of the mass. It has 6 degrees of freedom including the translational movements in the x, y, and z axes and the rotational movements which are roll, pitch, and yaw. Two of the motors spin clockwise and two of the motors spin counterclockwise to generate lift without getting a constant yaw and spin on the quadcopter. The configuration of the propellers is shown in Fig. 2.1.

In other words, the state of the UAV is described by a state vector that consists of two main parts: the position of the UAV in space and its yaw rotation. This state vector describes where the UAV is located and in which direction it is facing at a certain point in time. Additionally,

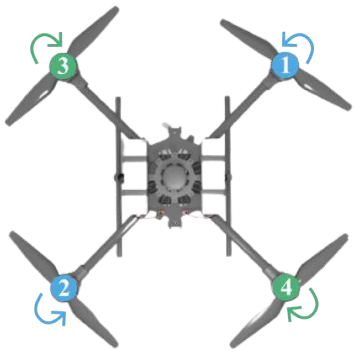


Figure 2.1: The spin direction of the propeller motors on the quadcopter.

within this thesis, the exploration algorithms assume that there is a maximum linear velocity and a maximum angular velocity around z axis. These limits are important to ensure that the movements of the UAV are safe, especially when performing experiments in the real world.

2.2 LiDAR and RGB-D Sensors

To map and explore the environment, a robot is equipped with sensors such as a LiDAR or an RGB-D camera, shown in Fig 2.2. Both have advantages and disadvantages. However, compared to other sensor types, these sensors are suitable for autonomous exploration and mapping because they can capture a large amount of data with each sensor measurement and provide information about the distance to obstacles in sight. The ability to measure the distance to an object is one of the key features as it gives the robot depth perception.

LiDAR uses light waves to measure the distance to a target. To measure the distance, a LiDAR emits a laser light that bounces back to the LiDAR from a physical object. By knowing the constant speed of light in the air and the time it takes for the signal to travel from emission to rebound, the distance can be calculated. RGB-D cameras use two or more lenses to achieve human binocular vision. This allows the robot to capture 3D images. The images provided by the RGB-D cameras contain the distance to each pixel in the camera frame.

The advantages of LiDAR technology are the high measurement range and accuracy, the ability to measure 3D structures and the fast update rate, which gives the robot a real-time view. As it works with light, it can also be used in dark environments. LiDAR technology is independent of extreme weather conditions such as extreme sunlight and other weather scenarios. Another important advantage of LiDAR is that it can have a horizontal field of view of 360°. On the downside, LiDARs are quite expensive, and transparent surfaces can lead to incorrect measurements. The vertical field of view is also limited with a 3D LiDAR, so there are some blind spots above and below the sensor. Furthermore, due to its weight, it cannot be mounted and carried by small UAVs.

The remarkable advantages of RGB-D cameras lie in their high mobility and low cost. How-

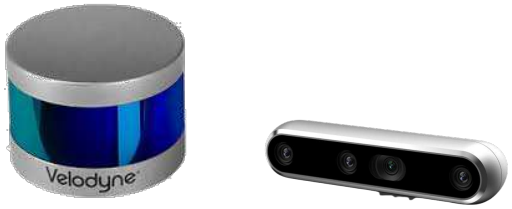


Figure 2.2: A Velodyne $VLP-16$ LiDAR sensor with a maximum range of 100 m and a Realsense $D455$ camera with a maximum range of 6 m, used in the experimental evaluation of the autonomous exploration and mapping in this thesis.

ever, RGB-D sensors require a light source and only allow measurement ranges of a limited distance and a limited field of view.

Depending on the use case, UAV system and requirements for autonomous exploration and mapping, one can choose between LiDAR and RGB-D sensors, as each offers distinct advantages and limitations. The selection ultimately hinges on factors such as the complexity of the environment, the level of detail required in the mapping, budget constraints, and specific mission goals.

2.3 3D Map Representation

A popular approach to model 3D environments is to discretize the world into equal-sized cubic volumes, called voxels. One of the major shortcomings of fixed grid structures is that the size of the area to be mapped has to be known a priori. Voxel hashing [10] is one approach to overcome this shortcoming, as fixed-sized blocks are allocated on demand. Memory requirements can also be a problem when mapping large areas at a high resolution. Hierarchical data structures such as hierarchical voxel hashing [11] and octrees [12] are used for this purpose, where the map can be displayed at different resolutions. One of the most popular mapping frameworks is OctoMap [13]. OctoMap uses an octree-based data structure, as proposed in [12], to do occupancy mapping.

There are also other mapping frameworks in the literature that have good capabilities. The mapping framework Voxblox [14] uses a signed distance field [15] voxel grid, with voxel hashing for dynamic growth, as representation. It was mainly developed for planning or trajectory optimization in the context of micro aerial vehicles (MAVs). The signed distance field representation makes trajectory optimization faster by storing the distance to the closest obstacle in each voxel. Voxblox builds on [10] where they use a spatial hashing scheme and allocates blocks of fixed size when needed. This means that the size of the area to be mapped does not need to be known in advance. In recent years, several mapping frameworks [16, 17, 18, 19, 20, 21] for storing dense semantic information have been proposed. All of these are based on the voxel

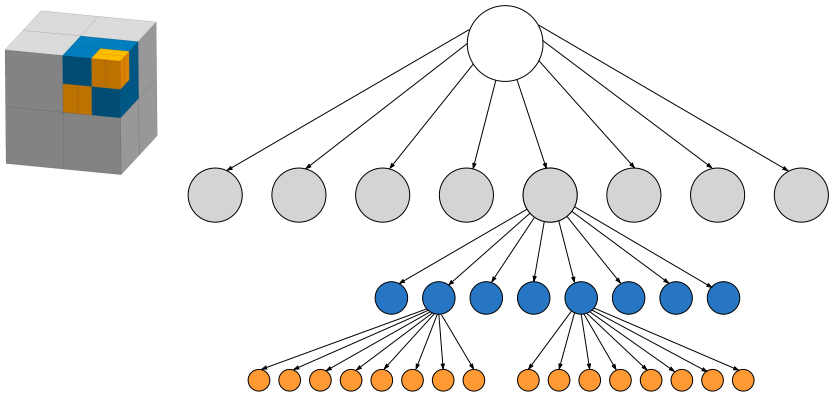


Figure 2.3: An example of octree with space subdivision in octants.

hashing approach of [10].

Within this thesis, the OctoMap framework, developed by Hornung et al. [13], is used to describe the environment. As previously mentioned, it is based on an octree data structure. An octree is a tree data structure in which each node has exactly eight children, as shown on the right side of Fig. 2.3. By using an octree, it is possible to divide a 3D space into eight octants. According to this principle, each node represents a part of the space and its children represent the eight octants of this part, i.e. eight smaller parts of the space (see left side of Fig. 2.3).

The octree structure allows for delaying the initialization of the grid structure. It is also often more memory efficient compared to voxel hashing or fixed-size grid structures since the information can be stored at different resolutions in the octree, without losing any precision. If the inner nodes of the octree are updated correctly, it is possible to do queries at different resolutions. Namely, the model is represented as coarse to fine, with similar voxels compressed into a single node, while volumes containing more details are represented by a larger number of smaller voxels. Fig. 2.4 shows the different layers of an octree-based model. Each image shows a deeper level (higher resolution) of the octree compared to the previous one. Querying at different resolutions can be especially beneficial in systems where multiple algorithms use the same map but have different computational and time requirements.

The OctoMap has several advantages over other approaches, such as probabilistic representation, storage efficiency and flexibility (multi-resolution). The point cloud data is used to create and update the map in each iteration. This can be raw data from sensors (LiDARs or cameras) or processed and accumulated point clouds such as the Google Cartographer submap point cloud [22]. The OctoMaps created from the LiDAR point cloud in the simulator, the camera point cloud and the LiDAR Velodyne $VLP-16$ point cloud are shown in Fig. 2.5, respectively.

As already mentioned, each cube of the OctoMap is called a voxel (cell), which can be *free*, *occupied* or *unknown*. With OctoMap, every voxel has an occupied possibility. The value of the occupied possibility is in the range of 0 to 1.

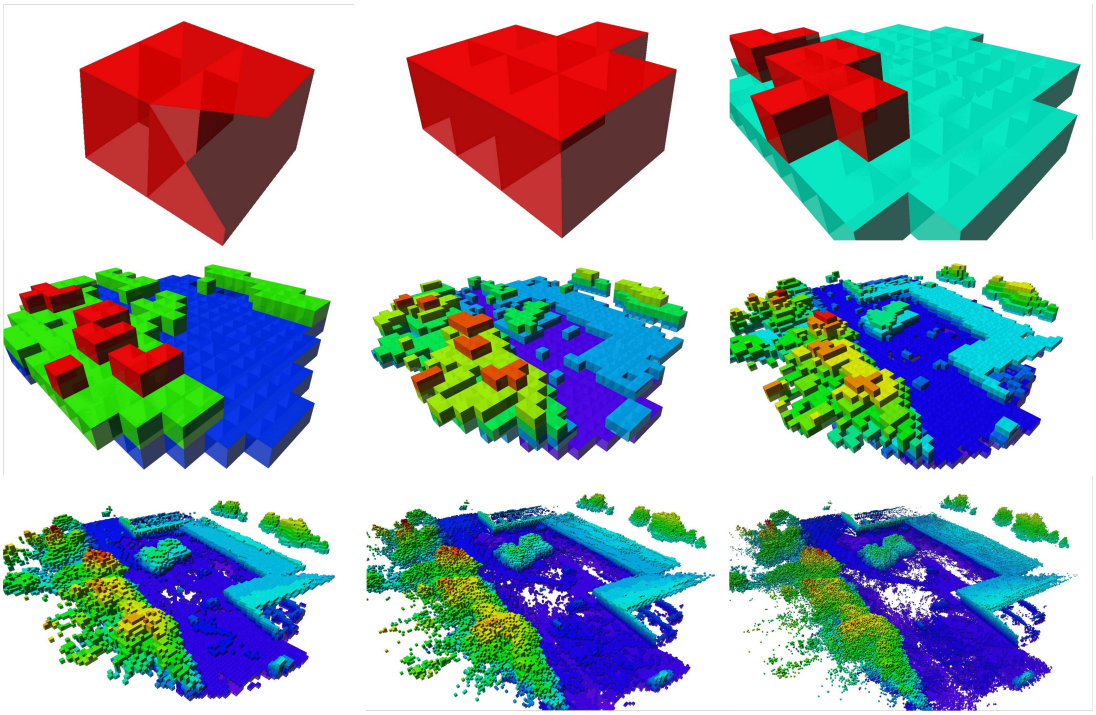


Figure 2.4: Coarse to fine visualization of an octree-based model.

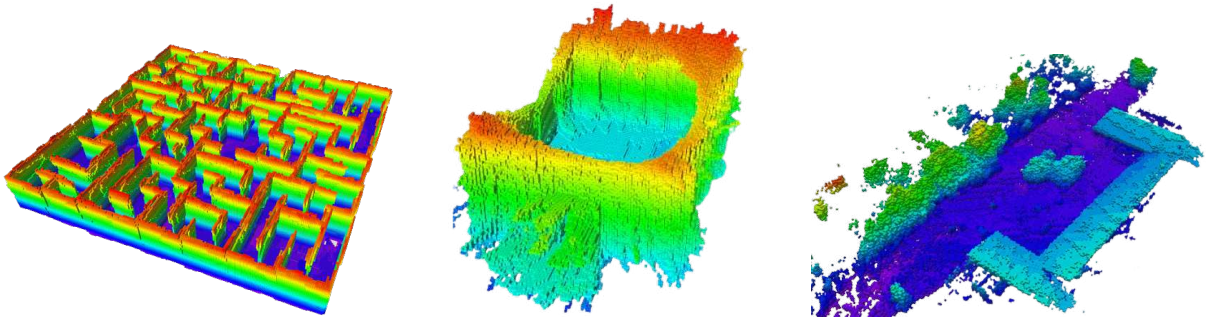


Figure 2.5: Examples of OctoMaps that were created during the exploration in a simulation, an indoor and an outdoor environment as part of this thesis.

The OctoMap plays an important role as it is used for both exploration and path planning. It serves as a key tool for identifying new areas to explore and navigating safely through the environment. For example, in the frontier-based exploration strategy, the OctoMap can be used to easily distinguish between known and unknown areas (frontiers). In this case, the OctoMap influences the extraction of potential waypoints for exploration, their evaluation (calculation of information gain) and finally the selection of the next best goal. To navigate to these new waypoints, OctoMap provides a detailed map of the environment, including obstacles and potentially dangerous areas. The path planner uses this information to calculate a trajectory that avoids collisions, respects the flying capabilities of the UAV, and optimizes factors such as distance. In general, the extraction of the potential waypoints for exploration depends on the applied exploration strategy (frontier-based or sampling-based), which is explained in Chapter 4, while the selection of the next best goal is based on the calculation of the information gain,

which is presented in Subsection 2.4.2.

2.4 Information Theory

The focus of this thesis is the autonomous exploration and mapping of 3D environments. To achieve full autonomy, a fundamental component is the exploration algorithm described in Chapter 4. The core of the exploration algorithms relies on the concept of information gain, a principle rooted in information theory as described in [23]. Information theory is the mathematical study of the coding of information, incorporating several aspects that are important in robotics algorithms. This section provides the basic definitions and concepts within information theory, emphasizing the algorithms used for information gain calculation.

2.4.1 Entropy

Entropy is a central concept in the field of information theory, and it quantifies the amount of uncertainty or randomness associated with a set of probabilities. In simpler terms, it measures the unpredictability of the information content. For a discrete random variable X with possible values $\{x_1, \ldots, x_n\}$, the entropy $H(X)$ is defined as:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \ln p(x_i), \tag{2.1}$$

where $p(x_i)$ is the probabilty that $X = x_i$. Entropy indicates the average amount of information required to represent an outcome of X. A higher entropy indicates greater unpredictability, while a lower entropy indicates greater predictability. Given a second random variable Y, the amount of information required to describe the outcome of X can be defined. If the value of Y is known, the conditional entropy $H(X|Y)$ is as follows:

$$H(X|Y) = \sum_{j=1}^{m} p(y_j) H(X|Y = y_j), \tag{2.2}$$

where $H(X|Y = y_j)$ is defined as:

$$H(X|Y = y_j) = -\sum_{i=1}^{n} p(x_i|y_j) \ln p(x_i|y_j). \tag{2.3}$$

As already mentioned, the random variables X and Y are described as discrete values and the above definitions are based on this assumption. If X and Y are continuous random variables, it is necessary to replace the sum by an integral over all possible values of X or Y. The entropy of a continuous random variable X is therefore described as follows:

$$H(X) = -\int p(x) \ln p(x) dx, \tag{2.4}$$

and the conditional entropy as:

$$H(X|Y) = \int p(y) H(X|Y = y) dy, \tag{2.5}$$

with x and y values of X and Y, respectively.

2.4.2 Information Gain Calculation

In order to be able to make autonomous decisions during exploration, the calculation of information gain is of crucial importance. It is used to quantify how much one learns about the environment by performing a certain action. In general, this quantity is the amount of information obtained about a random variable by observing another random variable. The information gain can be defined using entropy and conditional entropy as follows:

$$I(X,Y) = H(X) - H(X|Y). \tag{2.6}$$

More explicitly, it is defined as:

$$I(X,Y) = \sum_{j=1}^{m} \sum_{i=1}^{n} p(x_i, y_j) \ln \left(\frac{p(x_i, y_j)}{p(x_i) p(y_j)} \right), \tag{2.7}$$

and it continuous case:

$$I(X,Y) = \int \int p(x,y) \ln \left(\frac{p(x,y)}{p(x) p(y)} \right) dxdy. \tag{2.8}$$

The information gain can be calculated using a raycasting algorithm (RC). In autonomous exploration, before making an observation, the robot faces uncertainty about the environment, quantified by the initial entropy $H(X)$. The RC can be employed by the robot to simulate sensor readings. Rays are projected into the environment to detect obstacles or other points of interest. The sensor readings or observations, resulting from this RC represent the variable Y. After performing RC, the robot has new information about the environment that allows it to update its knowledge and reduce uncertainty. This new knowledge is represented as conditional entropy $H(X|Y)$, which represents the remaining uncertainty about the environment given the new sensor readings. The information gain $I(X,Y)$ is then calculated as the difference between the initial entropy and the conditional entropy, indicating by how much the uncertainty about the environment has decreased thanks to the new observations. For a robot, this difference is the improved understanding of its environment. To put this into action with the RC, one would start by discretizing the environment if necessary, simulating how the sensor interacts with the

environment, and then estimating the probability distributions that represent the likelihood of the sensor readings given the state of the environment. This involves calculating how often a simulated ray interacts with certain features of the environment across different grid cells or space partitions. These interactions provide the data to estimate the joint probability distribution $p(x,y)$ and the individual distributions $p(x)$ and $p(y)$. Once these probabilities are estimated, you can compute the entropies and thus the information gain by integrating or summing over these distributions for continuous or discrete spaces respectively.

In this way, the RC algorithm estimates the expected reduction in uncertainty or estimates information gain for each potential action or observation. This process forms the basis for the robot to make decisions that maximize the efficiency of its exploration by steering into areas that promise the highest information gain.

When looking into exploration process, once the potential waypoints for exploration are extracted from the OctoMap, each waypoint is evaluated. The evaluation of the potential waypoint usually considers the information gain of the potential waypoint and the distance from the current position of the robot to the potential waypoint position [24]. The information gain in a probabilistic volumetric map, an OctoMap, is defined as the sum of expected information enclosed in voxels, that are likely to be visible from a particular position (potential waypoint position). As already mentioned, each position in the OctoMap can be evaluated by using a specific algorithm, for example, RC to traverse voxels around candidates. The motion of the robot is formulated as an optimization problem, where the expected entropy change is minimized, or, in other words, the expected information gain is maximized. The goal of autonomous exploration is to determine future poses that maximize the map information gain, or equivalently minimize the map entropy. Since autonomous exploration is conducted in uncertain or unknown environments, a complete path is not known a priori.

In the literature, the information gain calculated from a map is commonly determined with Shannon's entropy [25], an uncertainty metric based on grid cell occupancy probability, such as in [26, 27]. Yamauchi et al. [9] put forward the exploration strategy of the nearest frontier, whose revenue function is inversely proportional to the length of the path. Gonzales-Banos et. al. [24] proposed a revenue function that combines the expected information gain and the length of the expected path. Umari et al. [28] proposed a new information gain estimation method, which only considers the unknown grid inside of the circle centering at the frontier with a predefined radius, and further combined the information gain with the path cost in the benefit function. However, most of them evaluate the candidate frontiers independently and ignore the geometric continuity of obstacles in the environment. In [29], mapping is performed in 3D while motion planning is done in 2D to reduce computational complexity since an MAV typically flies at a constant height. Bissmarck et al. [30] compared various approaches to compute the information gain for candidate views. The proposed frontier-oriented volumetric hierarchical

ray tracing was benchmarked against the hierarchical ray tracing by Vasquez-Gomez et al. [31] in computation time, mapping efficiency and estimation error. The second group of evaluation metrics estimates the amount of unknown volume in the view frustum via RC. This metric is used in [6, 32, 33, 34].

Furthermore, the information gain calculation should be calculated in real time and onboard the robot with limited resources. In general, the information gain represents a volume of the unmapped space that would be observed by robot sensors when the robot is positioned in the potential waypoint for exploration. A common algorithm used for the information gain estimation is the RC algorithm [35], and its results are then weighted by the cost of the robot traveling to the candidate. The described method is used in RH-NBVP [6] approach. The main drawback of the RH-NBVP is the significant computation time required to compute the information gain using the RC. A more efficient algorithm in terms of computation time is the Recursive Shadowcasting (RSC) algorithm, proposed in [36]. Both algorithms will be explained in detail in the following sections.

2.4.3 Raycasting Algorithm

Raycasting (RC) is a fundamental technique in computer graphics that traces rays from a source point in straight lines through a 3D space. This method can be employed to determine the visibility and intersections between objects, often in rendering contexts such as ray tracing [37]. Beyond graphics, RC is useful for many applications, including robotics and environmental sensing. An important use case is the calculation of information gain in the exploration process.

Information gain for each candidate position is calculated using an RC algorithm which traces the path of a series of rays originating from a given candidate. The density and range of rays define the area to be examined and are specified in advance. When one of these rays hits an obstacle (e.g., a wall), all voxels that the ray previously touched are considered occupied voxels. Otherwise, the voxels are considered as free or unknown, depending on the current state of the OctoMap. This results in knowing the number of free and unknown voxels in a predefined area, in each direction from a specific position. Based on this information, a robot can take appropriate actions to move to an unknown area to reduce the total exploration time. In other words, when evaluating potential sensor positions or robot movements in an environment, maximizing information gain (i.e., the amount of new, valuable information obtained) is often crucial. By casting rays from a point source, the amount and quality of information that can be obtained from that position are calculated. For instance, a sensor might obtain more information from an unobstructed viewpoint than one blocked by obstacles. Furthermore, a large number of rays can be cast rapidly, allowing for fine-grained analysis of potential information gain in large environments. However, with limited computational resources available onboard the UAV, it cannot run in real-time, especially if using dense rays.

In general, all algorithms that directly cast rays into the map, cast more rays than necessary because they cast a fixed number of rays regardless of the design of the environment [38]. It is shown in [39] that the computation time of RC algorithm increases as the predefined area increases. This is because the number of rays depends on the predefined area and is not affected by the occupied voxels (obstacles). The problem of computational effort required to calculate the information gain becomes even more apparent when using sensors that produce large or dense point clouds, such as LiDARs or cameras.

2.4.4 Recursive Shadowcasting Algorithm

Recursive Shadowcasting (RSC) was first used in computer games to calculate an FOV from a top-down perspective, where the FOV is defined as a set of locations visible from a specific position in a computer game scene [38]. The original RSC, proposed in [36], considers a 2D FOV grid and initially sets all grid cells to not visible. Then the grid is divided into eight octants centered on the FOV source (**S**) and the cells within each octant are traversed [38]. This traversal occurs within each octant by rows or columns in ascending order of distance from the FOV source. Fig. 2.6 shows the steps of the RSC on an octant. When a cell is traversed, its visibility state is set to visible. However, when an occupied cell (the black cell) is encountered, an octant is recursively split into smaller regions (Fig. 2.6 b) and c)), which are bounded by rays cast from the FOV source to the corners of the occupied cell (blue dashed rays). The cell traversals are then continued within each smaller region. As marked in Fig. 2.6 a) with green arrows, the algorithm first processes rows one through five without encountering any occupied cell. In line six, three occupied cells are encountered, splitting the free region in two and causing the algorithm to call itself recursively. The recursive call then processes the free region on the left (Fig. 2.6 b)), while the main iteration of the algorithm continues processing the free region on the right. Note that even if a ray only grazes the edge of a cell, that entire cell is set to visible. The result of the RSC on an octant in 2D is shown in Fig. 2.6 d), where occupied cells are shown in black, visible cells in yellow and invisible cells in grey. Similarly, the main goal of the RSC in the information gain calculation is to find unknown voxels of the OctoMap among the visible cells. Similar to the FOV grid in 2D computer games, the 3D OctoMap used in this thesis is divided into cube-shaped voxels, allowing us to take advantage of the RSC and calculate the information gain.

Within this thesis, the exploration strategies use sensors that produce large point clouds with each scan, such as LiDARs and dense point clouds from cameras. The RC in the information gain calculation process increases the computation time as the input data increases (shown in [6], [8], [40]). Therefore, using the RSC solves the computational bottleneck and results in a significant computation time reduction during planning iterations.

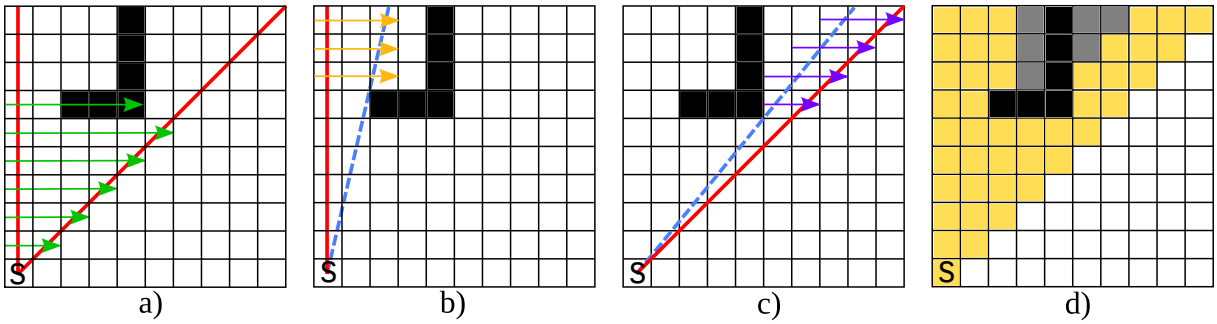


Figure 2.6: Steps of the RSC on a single octant.

2.5 Path Planning and Navigation

Path planning in robotics and autonomous systems is a crucial computational process in which an algorithm determines the most suitable route for a robot to take to get from one point to another. The process begins with an understanding of the environment, which includes mapping the surroundings, identifying obstacles and understanding the terrain. The algorithm must be given a starting point and a goal point. One of the main goals of path planning is obstacle avoidance, i.e. the algorithm must recognize all physical obstacles in the environment and navigate around them. In autonomous exploration path planning is required for the ability to explore the environment, which in the scope of this thesis means moving between selected waypoints. A key aspect of path planning is optimization, which is about finding not just any path, but the optimal one. This can mean choosing the shortest, the fastest or the most energy-efficient route, depending on the specific requirements of the task. Path planning algorithms should be able to adapt dynamically, especially in environments where obstacles or conditions may change, requiring path changes in real time.

Different algorithms are used for path planning, each with unique strengths and suitable for different applications. These include A* [41], Dijkstra's algorithm [42], Rapidly-exploring Random Tree (RRT) [43, 44] and their variants. Finally, the chosen path must be feasible considering the robot's capabilities, including factors such as size, speed, maneuverability, and ability to navigate varying terrain. Path planning is a critical component in autonomous systems, ensuring that they can perform their tasks efficiently while adapting to the complexity and unpredictability of the real-world environment.

In this thesis, path planning and navigation are not the focus, but play an important role in autonomous exploration as they ensure movement throughout the environment. The path planner uses the Rapidly-exploring Random Tree Star (RRT*), an improved version of the original RRT algorithm. The main difference of RRT* lies in its ability to improve the quality of paths. Unlike RRT, which stops after finding a feasible path, RRT* continuously refines the path through rewiring steps during expansion. Over time, this process leads to more efficient

and near-optimal paths. The approach utilized within this thesis has been developed in [45, 46].

The OctoMap is used for path planning and navigation. The path planner strategically generates routes through free voxels that ensure safe navigation. In the practical implementation, the path planner takes a binary representation of the OctoMap as input, which provides an efficient and compact description of the environment. The UAV is represented in the path planner as a rectangular prism with the corresponding dimensions.

CHAPTER 3

State-of-the-Art

Significant progress has been made in the field of autonomous exploration. Various methods have been developed to improve the efficiency and effectiveness of navigation and understanding of unknown environments. This section looks at the extensive research into these methods, focusing on three key areas: frontier-based, sampling-based, and semantically-enhanced exploration.

In the area of frontier-based exploration, the basic concept where the robot navigates to a point on the frontier is examined. The frontier is defined as the boundary between known and unknown areas. This approach, first introduced by Yamauchi [9], has evolved through numerous studies. Various algorithms were proposed to improve this strategy. Elements such as information gain, exploration costs and different functions for the selection of frontier points were considered. Sampling-based exploration is characterized by the fact that it focuses on producing and evaluating a series of points in the area to ensure complete coverage. This method typically involves sampling points near the frontier or randomly in the surrounding area and then evaluating these points for their potential information gain. Sampling-based methods are known for their effectiveness in cluttered spaces, but can face challenges such as local minima that affect the full coverage of the area. Finally, semantically-enhanced exploration represents a significant advance in integrating semantic information from the environment into the exploration process. This approach not only takes into account the geometrical layout of the space, but also incorporates semantic understanding, which significantly improves the efficiency of exploration. By using semantic image segmentation, this strategy improves the ability of the robot to make informed decisions about exploration by utilizing the rich information from the environment. Each of these exploration strategies contributes to the overall goal of autonomous exploration and offers unique solutions and perspectives. This section provides an in-depth look

at these methods, their evolution over time, their specific applications, and the various ways they have been adapted and improved in recent research.

3.1 Frontier-Based Exploration

A characteristic of frontier-based approaches is exploration by approaching a selected point on the frontier between the explored and unexplored portion of the environment. This idea was first introduced by Yamauchi in [9] and subsequently evaluated in more detail in [47]. In each iteration, the next best goal is a frontier point closest to the robot. This method has two shortcomings for the exploration task. First, it treats all frontiers equally and secondly, it is limited to one source of information, finding new areas. The exploration process is complete when no frontiers remain. There are many different frontier point selection algorithms. Simmons et al. [48] and Moorehead et al. [49] present the frontier point selection function by combining information gain and exploration cost to select the target frontier point. Carlone and Lyons [50] use the Mixed-Integer Linear Programming (MILP) model to obtain the optimal frontier point for autonomous exploration. The work by Mei et al. [51] proposes an algorithm to choose the next target frontier point for the robot to explore based on orientation information. The authors in [52] propose a novel exploration strategy that exploits background knowledge by considering previously seen environments to make better exploration decisions. Gautam et al. [53] use the K-means algorithm to cluster frontier points and assign these frontier points to the robots using a Hungarian method.

The simplest approach to 3D exploration is to use 2D frontier-based exploration with 3D maps at different heights (oftentimes called 2.5D approaches) [54]. A complete frontier-based solution for 3D environments is developed in [7], where the next best goal is the frontier that minimizes the velocity change to maintain a consistently high flight speed. It is shown that this approach outperforms the closest frontier method [9]. Zhu et al. [1] introduced an exploration tool called 3D-FBET. This is a frontier-based tool that is performed in three phases. The phases are 3D mapping, frontier detection in combination with a clustering algorithm, and the selection of the best frontier. Through experimental evaluation in different environments, 3D-FBET showed several shortcomings. First, because the frontier detection is based on a subset of altered voxels (generated from the camera point cloud), which is highly variable, the obtained frontiers were noisy and not reliable. Furthermore, the resulting frontier presented only a local view and the clustering was not adapted to the environment. These problems led to a higher total exploration time [55]. Mannucci et al. [2] used not only local but also global frontiers with two OctoMaps and two frontiers (local and global) with different resolutions. Global frontiers are assigned when the set of local frontiers is empty. Manucci evaluates the best frontier using a cost-utility approach, similar to [3]. Fast and efficient exploration performance was achieved

in [56]. The computationally expensive frontier clustering employed in classic frontier-based exploration is avoided by exploiting the implicit grouping of frontier voxels in the underlying octree map representation. Candidate next-views are sampled from the map frontiers and are evaluated using a utility function combining map entropy and travel time, where the former is computed efficiently using sparse raycasting. An advantage of an OctoMap is also used in [4] to merge voxels with an equal state, which results in the reduction of obstacle detection calculations. In the study by Senarathne et al. [57], he augments the traditional frontier-based exploration strategy to include a probabilistic decision step that decides whether further motion on the planned path to the next sensing location is desirable or not. If the motion is not desirable, it is canceled and a new sensing location is selected as the next sensing task. In [58], authors present a strategy based on frontier points optimization and multistep path planning. In order to get the best frontier point, they propose a Random Frontier Points Optimization (RFPO) algorithm. This algorithm optimizes the random frontier points generated by the RRT algorithm. Combining this algorithm with the frontier points evaluation function, they obtain the current optimal frontier point. Stachniss et al. [59] propose an algorithm that uses a highly efficient Rao-Blackwellized particle filter to represent the posterior about maps and poses. It trades off the cost of executing an action with the expected information gain and takes into account possible sensor measurements gathered along the path taken by the robot. Furthermore, frontier-based methods are combined with information theory to design a differentiable utility function for solving optimization problems such as optimal yaw angle and path optimization in [56, 60, 61].

3.2 Sampling-Based Exploration

While frontier-based methods are suitable for large environments due to their ability to identify unexplored spaces in the global map, sampling-based approaches are well-suited for cluttered spaces but can suffer from local minima issues, hindering complete coverage of the target environment [62]. Sampling-based or Next Best View-based (NBV) approaches aim to determine a (minimal) sequence of robot (sensor) viewpoints to visit in the environment until the entire space is explored. Potential viewpoints are typically sampled near the frontier or randomly. Then these viewpoints are evaluated for the potential information gain and the next best viewpoint is assigned. One of the first sampling-based methods is presented in [24] and then extended in [6], [63], [64]. In [6], authors proposed the Receding Horizon Next-Best-View planning (RH-NBVP), which uses a RRT [43], [44] to guide a UAV into the unexplored area. While the method showed good scaling properties and performance in a local exploration, it is not resilient to dead ends, resulting in poor global scene coverage and, thus, a high overall exploration time, as shown in [7], [56], [60]. Improvements of the RH-NBVP are presented in

[32, 65]. The exploration times are later improved in [63]. Often NBV approaches are used to build a 3D object without any a priori information, as in [66] and [67].

Hybrid strategies combine the advantages of both frontier-based and sampling-based approaches and can be combined in various ways as shown in Selin et al. [8] successfully combine the RH-NBVP with conventional frontier reasoning to compensate for a poor performance in global exploration. Respall et al. [40] sample viewpoints in the vicinity of a point of interest near a frontier and additionally memorize nodes that indicate regions of interest in a history graph to reduce the gain calculation time. Furthermore, authors in [68] introduced the Fast UAV ExpLoration (FUEL), a frontier-based exploration approach complemented with a Travelling Salesman Solver to generate minimum-time trajectories for visiting target locations. Namely, FUEL introduced an incrementally updated frontier information structure (FIS), which stores information about the explored space to facilitate high-frequency exploration planning. The FIS is utilized in their proposed hierarchical planner. The hierarchical planner first generates a global tour, similar to [69], that covers all frontier regions. The costs of moving between every pair of frontier regions are cached inside the FIS to make the global tour computationally feasible. Next, a local segment of the tour is refined and a safe minimum-time trajectory, accounting for the dynamics of the robot, is generated for the first part of the refined tour. The FUEL strategy was further extended by [70], adapting the approach for collaborative exploration with a decentralized team of UAVs, thereby reducing the exploration time required for large environments.

3.3 Semantically-Enhanced Exploration

Apart from classification related to candidate extraction and evaluation, exploration algorithms differ in the map used for exploration policy. Besides the volumetric map, such as the OctoMap [13], the environment can also be represented by a topological map with semantic features [71], which can improve the efficiency of the robotic exploration by facilitating the next best goal selection. The nodes on the graph that contain the semantic features are used to guide the exploration. Gomez et al. [72] presented a hybrid mapping approach that combined topological mapping with 3D dense mapping for large indoor 3D environments.

Recently, more and more exploration systems use semantic features from volumetric maps to evaluate candidate locations and select the next best goal. The authors in [73] extend the sampling-based approach from [6] to include the semantic segmentation information in a harbor-like environment. Similarly, Ashour et al. [74] presents an exploration strategy for UAVs that integrates environmental semantics for object mapping. The approach combines semantic information with autonomous exploration techniques to guide the exploration path and enhance object mapping efficiency using the approach from [75]. Instead of mapping objects during the

exploration, objects can be extracted from 2D images and then converted to 3D point types using the point cloud library (PCL). Previously, Wang et al. [76] introduced the extraction of edges. Furthermore, most of the semantic-aware exploration strategies are goal-oriented (search for an object), such as [77, 78, 79]. Authors in [77] introduced a frontier semantic exploration method for visual target navigation. Both frontier detection and semantic segmentation are performed using neural networks.

3.3.1 Semantic Image Segmentation

Semantic image segmentation plays a fundamental role in a variety of computer vision applications as it provides key information for the global understanding of an image. It assigns a specific class or category to each pixel of an image and aims to understand the content and context of an image by assigning semantic labels to different regions.

Several notable approaches have been proposed in the literature to address this problem. Shelhamer et al. [80] presented a fully convolutional network (FCN) that revolutionized semantic segmentation by extending convolutional neural networks (CNNs) to provide dense pixel-wise predictions. The FCN utilizes upsampling and skip connections to obtain spatial information and achieve accurate segmentation results. DeepLab, which was developed by Chen et al. [81], improves FCN by incorporating dilated convolutions that allow to capture of contextual information at multiple levels while maintaining spatial resolution. DeepLab uses various refinements such as Atrous Spatial Pyramid Pooling (ASPP) and post-processing techniques to improve segmentation accuracy.

Ronneberger et al. [82] introduced U-Net, an architecture specifically designed for biomedical image segmentation. U-Net consists of an encoder-decoder structure with skip connections that allows the network to capture both local and global context while preserving fine-grained details. Zhao et al. [83]proposed a Pyramid Scene Parsing Network (PSPNet) that uses a pyramid pooling module to capture context information at different levels. By aggregating features from multiple pyramid levels, PSPNet achieves robust semantic segmentation results. Mask R-CNN, proposed by He et al. [84] extends the popular Faster R-CNN object detection framework [85] to perform instance-level segmentation. Mask R-CNN combines region proposal generation with pixel-wise mask prediction, enabling accurate object segmentation along with object detection. The method developed by Sun et al. [86] proposed HRNet model focuses on obtaining high-resolution details in the entire network architecture. Unlike traditional architectures that downsample the input image at the beginning of the network, HRNet maintains a high-resolution representation by using parallel branches. These branches process the image at different resolutions and exchange information to effectively capture both local and global contexts. HRNet builds a multi-level representation from the high resolution and applies it to the Faster R-CNN, Mask R-CNN and Cascade R-CNN. This work represents a significant ad-

vance in semantic image segmentation as it addresses the challenges associated with context modeling, spatial resolution, and handling object instances. For a detailed literature review on this topic, [87] can be mentioned, which summarizes two decades of research in the field of semantic image segmentation. In this thesis, the HRNet model for semantic segmentation of 2D images is used. It showed enviable performance results [88]. Moreover, it is compact, fast, robust and easy to use, so it is possible to customize the model to run only on the CPU, which makes it suitable for applications on embedded systems (e.g. UAVs).

CHAPTER 4

Autonomous Exploration in 3D Environments

During the last decade, the sensing, planning, and control technologies for UAVs have made great progress. This chapter mainly describes the exploration strategies for UAVs and explains the difference between the current technologies and the contribution of this thesis. Exploration algorithms aim to identify a set of locations that an autonomous vehicle should visit in order to achieve complete coverage of the desired environment. These algorithms rely on local information gathered by perceptual sensors such as RGB-D cameras or LiDAR scanners [89]. Furthermore, exploration algorithms are designed to optimize various utility metrics such as information gain, energy consumption, or time required for exploration. Robotic exploration, which uses mobile robots to map unknown environments, has been studied for years. Some of the works focus on exploring the space quickly [7, 65]. Meanwhile, other methods place more emphasis on accurate reconstruction [69, 90]. Among the various proposed methods, there are two fundamental types of approaches: frontier-based approaches and sampling-based approaches. The frontier-based approaches try to maximize the exploration efficiency by selecting and exploring frontiers between the known and unknown areas of a map, while the sampling-based approaches randomly generate robot states and search the path that can maximize the information gathered in the environment. There are also hybrid strategies that combine the advantages of both frontier-based and sampling-based approaches. In recent years, semantically-enhanced exploration strategies have also been used to solve autonomous exploration problems. They are usually combined with frontier-based or sampling-based strategies. This chapter is divided into four sections. The first section gives a system overview, while the rest of the chapter presents the strategies for autonomous exploration, i.e. frontier-based, sampling-based and semantically-

enhanced exploration.

4.1 System Overview

In general, the proposed autonomous exploration system is modular and each module can be replaced with a custom module, e.g. path planning or mapping. The system relies on odometry data that can be obtained from any localization algorithm. Mapping is done using the OctoMap module, which requires a point cloud from a suitable sensing system, such as a laser scanner or a camera, to create a 3D map. The OctoMap is used for both exploration and collision-free navigation. The key part of this thesis is the exploration strategy that guides the robot in the environment. Each strategy (frontier-based, sampling-based and semantically-enhanced) utilizes the data from the OctoMap and the odometry data to make informed decisions about where to go next, prioritizing unexplored spaces or optimizing routes for efficiency. Finally, the path planner is the component that safely navigate the UAV through the environment. It takes into account the construction of the UAV and any potential obstacles, using the OctoMap to find a safe and efficient path. It results in a trajectory that is forwarded to the controller of the UAV. The system overview is shown in Fig. 4.1.

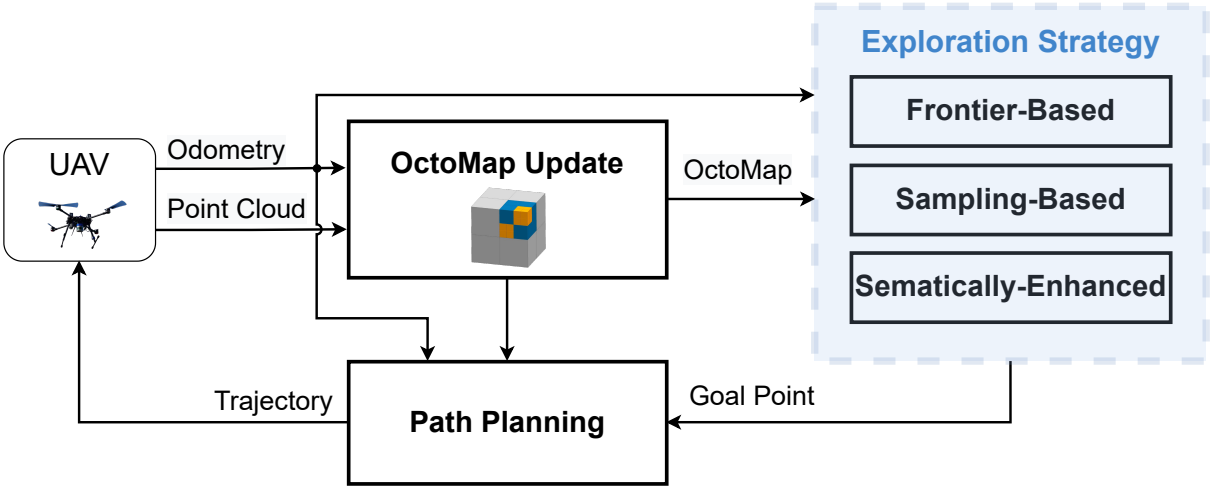


Figure 4.1: Overall schematic diagram of the 3D exploration system. The system input module generates odometry data and point cloud data that produce input data for OctoMap creation. The exploration strategy generates a goal point towards which the robot plans its path and navigates.

4.2 Frontier-Based Exploration

In frontier-based exploration, the central idea is to move a robot to the boundary between known (free) and unknown space to gain as much new information as possible. The points on the border between known and unknown space are called frontier points (frontiers) and they are

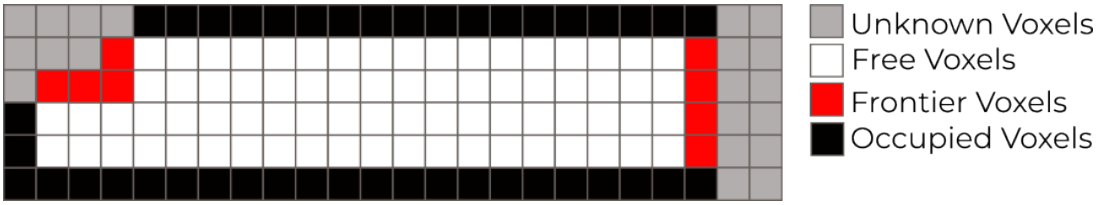


Figure 4.2: Illustration of a 2D frontiers. The frontiers are detected on the border between unknown (gray) and free (white) voxels.

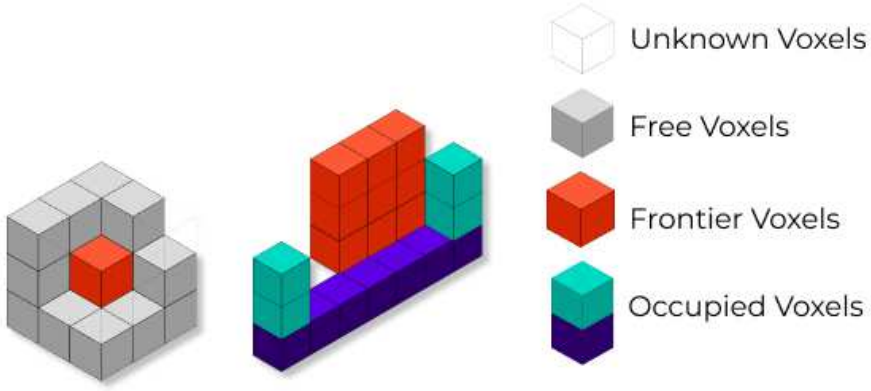


Figure 4.3: Illustation of a 3D frontier detection. The voxels are shown in 3D space which is a more complex, volumetric space where the boundary between free (gray) and unknown (transparent) voxels represents frontier (red) voxels.

shown in Fig. 4.2 for 2D and in Fig. 4.3 for 3D space. Detecting frontiers in 3D space is a much more complex challenge than in 2D space. The additional dimension in 3D requires more sophisticated algorithms to accurately identify and map boundaries between known and unknown areas. This complexity is also reflected in higher computational requirements, as processing 3D data requires more power and memory resources. Furthermore, the intricacies of 3D environments, including the varied terrain and potential obstacles, require advanced spatial awareness and decision-making algorithms to effectively navigate and explore frontiers. As the robot moves to the frontiers, it explores more unknown space and the map is consequently expanded. As the robot continues to navigate to the frontiers, the border between the known and unknown space is pushed back until there are no more frontiers. In this case, the environment is considered fully explored.

As stated by [9], frontier-based exploration would eventually explore all of the accessible space in the world, assuming perfect sensor and motor control. Therefore, frontier-based exploration is suitable in a restricted area as it eventually will reach the end of the exploration. The idea was first implemented and used to build 2D maps. However, lately, 3D maps have been created and used for frontier-based exploration. It is discussed in detail in Section 3.1. 3D maps are required for aerial vehicles to explore safely while allowing for changes in altitude, although it is more complex and requires more data resources.

In this thesis, frontier exploration is divided into three subsystems, frontier detection, fron-

tier clustering and frontier evaluation and selection. Each subsystem is explained in detail in the following text.

4.2.1 Frontier Definition and Detection

Within this thesis, the focus is on frontier detection and evaluation in 3D spaces. The definition of a frontier is that it marks the border between known (free) space and unknown space. An OctoMap M is selected for the map representation. It generates an occupancy grid map that is divided into a 3D grid. The OctoMap structure is explained in detail in Section 2.3. Each cube of the OctoMap, with a predefined resolution r, is denoted as a voxel (cell) $\mathbf{v}$, which can be *free*, *occupied* or *unknown*. Free voxels form the free space $V_{free} \subset V$, occupied voxels form the occupied space $V_{occ} \subset V$ and unknown voxels form the unknown space $V_{un} \subset V$. Initially, the entire bounded space is unknown, $V \equiv V_{un}$, and the unknown space decreases as the exploration advances. The entire space is a union of the three subspaces $V \equiv V_{free} \cup V_{occ} \cup V_{un}$. The exploration problem can be considered fully solved when $V_{occ} \cup V_{free} \equiv V \setminus V_{res}$, where V_{res} is residual space defined as an unexplored space, which remains inaccessible to the sensors.

In this thesis, both a 3D LiDAR and a camera have been used as sensors in frontier-based exploration. The detailed description of both sensors is given in Section 2.2.

A frontier, F, can be defined as a set of voxels $\mathbf{v}_f$ with the following property:

$$F = \{\mathbf{v}_f \in V_{free} : \exists neighbor(\mathbf{v}_f) \in V_{un}\}. \tag{4.1}$$

In other words, a frontier consists of free voxels with at least one unknown neighbor. The center of a frontier voxel is often called the frontier point. Frontiers are shown in Fig. 4.2 and Fig. 4.3.

The OctoMap M used for frontier detection is generated using LiDAR or camera scans S. The OctoMap is in the form of octrees, a format suitable for path planning. During the exploration, the OctoMap M is built iteratively using the method described in [13]. The current OctoMap M^i is created from the current LiDAR or camera scan S^i added to the OctoMap explored so far:

$$M^i = f(M^{i-1}, S^i), M^0 = \emptyset. \tag{4.2}$$

With each new incoming scan, a new OctoMap is created according to Eq. 4.2. At the same time, a frontier detection cycle is performed periodically to ensure that frontiers are constantly updated. Note that the rate of an OctoMap update process is lower than the frontier detection process since the OctoMap update is a computationally demanding process, especially when using dense scans.

The OctoMap used for frontier detection can be generated using raw data (as shown in the equations above) or using Cartographer submaps [22] and their matching point cloud. This

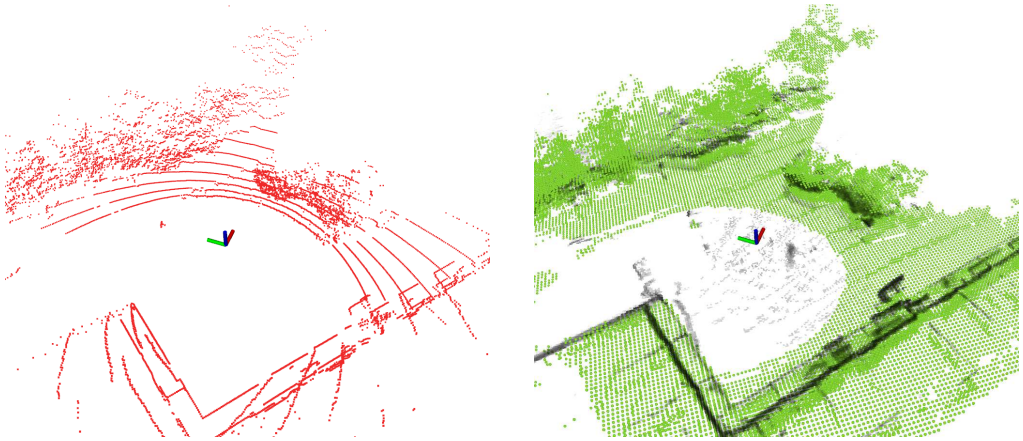


Figure 4.4: Comparison of raw sensor data and submap data. On the left is the raw sensor data, which typically includes a high volume of individual data points captured directly by sensors (in this case, LiDAR). This raw data is often noisy and unstructured, reflecting the immediate readings from the environment without any processing or filtering. On the right side, the submap point cloud is shown, which represents a more refined and organized version of the raw data. The submap point cloud is created from a predefined number of LiDAR scans.

process is described in detail in Publication 1 in Chapter 8. The comparison of raw sensor data and submap data is shown in Fig. 4.4. Generally, in Eq. 4.2, S^i can be replaced by any point cloud, less or more dense. Accordingly, it will be reflected in the density of the created OctoMap.

Once the map M is defined, the frontiers can be extracted from the map. The approach combines local and global frontiers, similar to Manucci [2]. Let V^i_{free} and V^{i-1}_{free} correspond to the free voxels in two consecutive OctoMaps, M^i and M^{i-1}. Then the local frontier F_l contains only newly created frontier points:

$$F_l = \{\mathbf{v}_f \in V^i_{free} \setminus V^{i-1}_{free} : \exists neighbor(\mathbf{v}_f) \in V^i_{un}\}. \tag{4.3}$$

The global frontier F_g is a union of all past local frontiers, which is updated and filtered in each iteration to exclude voxels that do not satisfy Eq. 4.1 anymore. F_g is calculated as follows:

$$\begin{gathered} F^i_g = F^i_l \cup F^i_{gf}, \\ F^i_{gf} = \{\mathbf{v}_f \in F^{i-1}_g : \exists neighbor(\mathbf{v}_f) \in V^i_{un}\}, F^0_g = \emptyset. \end{gathered} \tag{4.4}$$

There is usually a large number of voxels in the global frontier (referred to only as frontier from now on) and their evaluation is expensive in view of the computing effort involved. It is therefore common to cluster these frontier voxels in order to optimize the evaluation process. This clustering strategy culminates in the formation of multiple voxel clusters, where the geometrically central voxels of these clusters are identified as potential exploration targets.

4.2.2 Frontier Clustering

In this subsection, two different clustering methods are presented: a novel multi-resolution clustering approach and mean-shift clustering. The proposed novel multi-resolution clustering approach introduces a unique methodology that allows different granularity of the clustering process, which can improve the efficiency and precision of frontier voxel evaluation in complex environments. On the other hand, mean-shift clustering is a technique known for its efficiency in identifying clusters without prior knowledge of the number of clusters. These methods, their implementation and their effects on the efficiency of frontier exploration are thoroughly examined and discussed.

In this thesis, the multi-resolution properties of an OctoMap are utilized for efficient clustering of frontier voxels, an essential part of the proposed exploration strategy. The OctoMap is an advanced framework that uses an octree structure to represent 3D environments. An octree is a tree structure where each node branches into eight children, allowing recursive partitioning of the volume into finer segments (described in Section 2.3).

The OctoMap has a maximum depth d_{max} of 16 levels, with the voxel size at this deepest level defining the level of detail of the map. For frontier detection, the maximum depth is used, but depending on the structure of the environment, other layers may be more suitable.

Detecting the frontier at the maximum depth results in a large number of frontier points, which can consume significant computational resources, especially in expansive outdoor environments. To solve this problem, the number of frontier points is reduced by clustering them at a chosen exploration level, which is called d_{exp}. Both d_{max} and d_{exp} are shown in Fig. 4.5. It shows the OctoMap structure (different layers of an octree-based model), as Fig. 2.4 extended with multi-resolution clustering parameters description.

The level d_{exp} and the corresponding voxel size r_{exp} are selected based on the characteristics of the environment. For more open environments, d_{exp} can be lower, which leads to larger clustered voxels. The process involves determining the parent node at the exploration level d_{exp} which are linked to the known voxels at the maximum depth. For instance, if four depth levels are considered (as shown in Fig. 4.5) for the sake of simplicity, with $d_{max} = 4$ and $d_{exp} = 2$, the goal is to find the parent node at d_{exp} for the frontier voxels of d_{max}.

The proposed multi-resolution clustering algorithm is fast, robust, easy to implement and can be adapted to different map resolutions and exploration depths. It is suitable for both small and large areas. In the proposed methodology, this clustering approach is combined with the mean-shift clustering algorithm so that the robot can be directed to areas where the frontier points are denser, optimizing the exploration process.

Clustering algorithms play an important role in exploration tasks, as evidenced by the use of various methods such as the union-find algorithm, the depth-first algorithm and the flood fill clustering algorithm in the algorithms described in [91], [1] and [2] respectively. Deviating

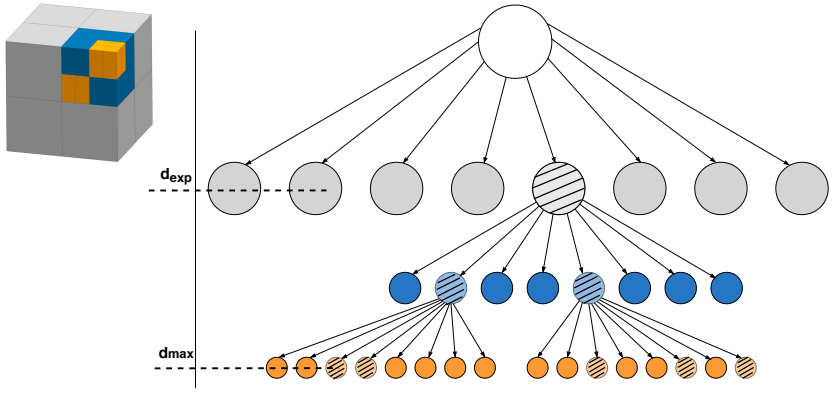


Figure 4.5: The structure of an octree and the cube-shaped space it represents. A representation of frontier voxels at the maximum depth d_{max} and their corresponding parent nodes at the chosen exploration depth d_{exp}.

from these traditional state-of-the-art approaches, the mean-shift clustering algorithm is used in this thesis, which is specifically adapted to 3D point data. This algorithm, originally introduced by Fukunaga and Hostetler [92], is characterized by its flexibility. It works without predefined assumptions about the shape of the distribution or the set of clusters. This is a notable advantage over other algorithms such as *K-means* [93], which require prior knowledge of the number of clusters.

The mean-shift clustering algorithm is particularly known for its robustness to noise and outliers, making it a reliable choice in scenarios where data quality can be inconsistent. Its non-parametric nature means that it does not assume a particular probability distribution for the data, which improves its applicability in various domains where the data distribution is unknown or complex. Through an iterative process, mean-shift moves each data point towards the densest area in its vicinity, which is defined by the bandwidth of the kernel. This process, which continues until convergence is achieved, usually results in well-defined clusters. The computationally most complex component of the mean-shift procedure is the identification of the neighbours of a point in 3D space (as defined by the kernel and its bandwidth). The kernel represents a weighting function and applying it to 3D points generates a probability surface (e.g., a density function). The kernel bandwidth regulates the size of the "window" over which the mean is calculated. In this thesis, the Gaussian kernel is used. Although a fixed bandwidth is typically used, the algorithm can also be implemented with variable bandwidths to adapt to different data densities, which can improve performance in datasets with heterogeneous characteristics. To make the mean-shift algorithm work in real time, along with the reduction in the number of frontiers by multi-resolution frontier clustering, an appropriate bandwidth is carefully selected to balance between computation time and the desired outcome with respect to the size of the environment and r_{exp}. The example of the mean-shift output is given in Fig. 4.6.

In addition to its use in autonomous exploration, mean-shift clustering is also widely used in

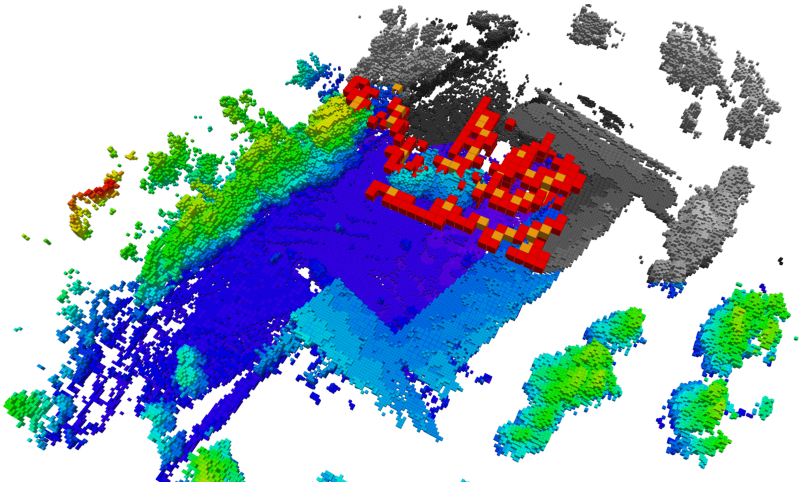


Figure 4.6: Results of mean-shift clustering during real-world experiments within this thesis. Grey and white parts of the OctoMap represent the unexplored and green/blue explored environment. Red voxels represent the frontier and yellow voxels are centroids of frontier point clusters, the output of the mean-shift algorithm.

areas such as image processing, computer vision and pattern recognition thanks to its ability to adapt to the underlying data structure. However, computational intensity remains a challenge, especially for large amounts of data or high-dimensional spaces, but this can be mitigated by various optimizations and efficient implementation techniques. These properties make mean-shift clustering a versatile and valuable tool in the interpretation of complex, real-world data in a wide range of applications.

4.2.3 Frontier Evaluation and Selection

The clustering algorithms result in the potential exploration targets (voxels) F_c. To evaluate which of the potential exploration targets could result in a faster exploration of the environment, *total gain* of every candidate $\mathbf{v}_c \in F_c$ is defined using the following function similar to the one proposed in [24]:

$$G(\mathbf{v}_c) = \frac{I(\mathbf{v}_c)}{e^{\lambda L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}_c})}}, \tag{4.5}$$

where λ is a positive constant, $L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}_c})$ is the distance between the current position of the robot $\mathbf{p}_i$ and the position of the candidate $\mathbf{p}_{\mathbf{v}_c}$, while $I(\mathbf{v}_c)$ is an *information gain* i.e. a measure of the unexplored region of the environment that is potentially visible from $\mathbf{v}_c$. The estimated distance is approximated using the Euclidean distance between the robot position $\mathbf{p}_i$ and the position of the candidate (voxel center) $\mathbf{p}_{\mathbf{v}_c}$, $L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}_c}) = \|\mathbf{p}_i - \mathbf{p}_{\mathbf{v}_c}\|$. The information gain $I(\mathbf{v}_c)$, introduced in Subsection 2.4.2, is defined as the share of unknown voxels in a cube placed around $\mathbf{v}_c$, as shown in Fig. 4.7. The size of the cube is defined with respect to the range of the used sensor.

The constant λ weights the importance of robot motion cost against the expected informa-

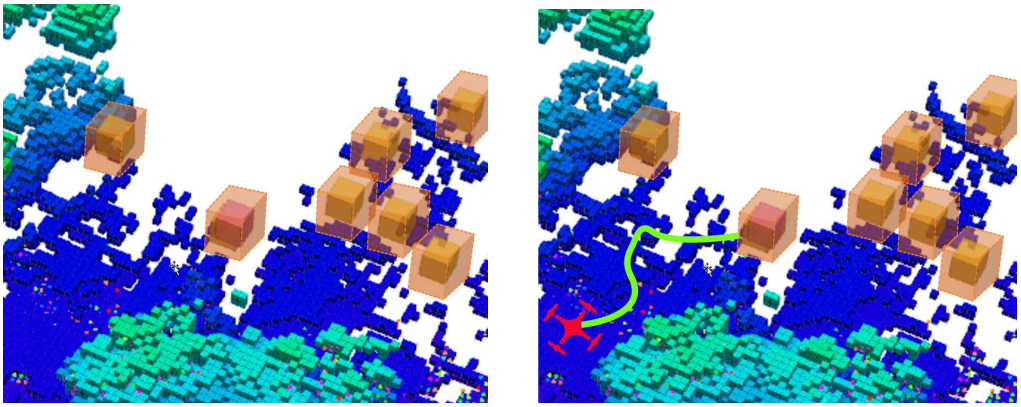


Figure 4.7: Information gain calculation for the potential exploration targets is shown on the left. Cubes are placed around the candidates and the information gain is proportional to the number of unknown voxels within the cubes. The UAV selects the best frontier according to Eq. 4.6, plans the path and navigates toward the target.

tion gain. A small λ gives priority to the information gain, while $\lambda \to \infty$ means that the motion is so expensive that only $\mathbf{v}_c$ near the robot is selected.

Finally, the best frontier voxel $\mathbf{v}_{bf}$ is one that maximizes the total information gain $G(\mathbf{v}_c)$:

$$\mathbf{v}_{bf} = \underset{\mathbf{v}_c \in F_C}{\arg\max}\, G(\mathbf{v}_c). \tag{4.6}$$

Choosing the best frontier will favor actions that explore the unknown environment and gather more information about the unknown space, taking into account the distance between the robot and each candidate $\mathbf{v}_c$.

As soon as the best frontier point is selected, it is forwarded to a path planner as a waypoint. The robot starts to follow the planned path and navigates to the best frontier point $\mathbf{v}_{bf}$, as described in Section 2.5.

4.3 Sampling-Based Exploration

Following the discussion of the frontier-based strategy for autonomous exploration systems in the previous section, another widely used approach, the sampling-based strategy, is introduced. While frontier-based strategies focus on the systematic exploration of the boundary between known and unknown areas, sampling-based strategies offer a different methodology. Sampling-based strategies use random sampling to create a roadmap or tree of possible paths in the environment. This approach is particularly advantageous in complex or high-dimensional spaces where boundary identification can be difficult or computationally intensive. Algorithms such as Probabilistic Roadmaps (PRM) and Rapidly-exploring Random Trees (RRTs) are prime examples of this strategy and provide theoretical guarantees such as probabilistic completeness [94]. PRM and RRT are the classic sampling-based path planning algorithms, which belong to the

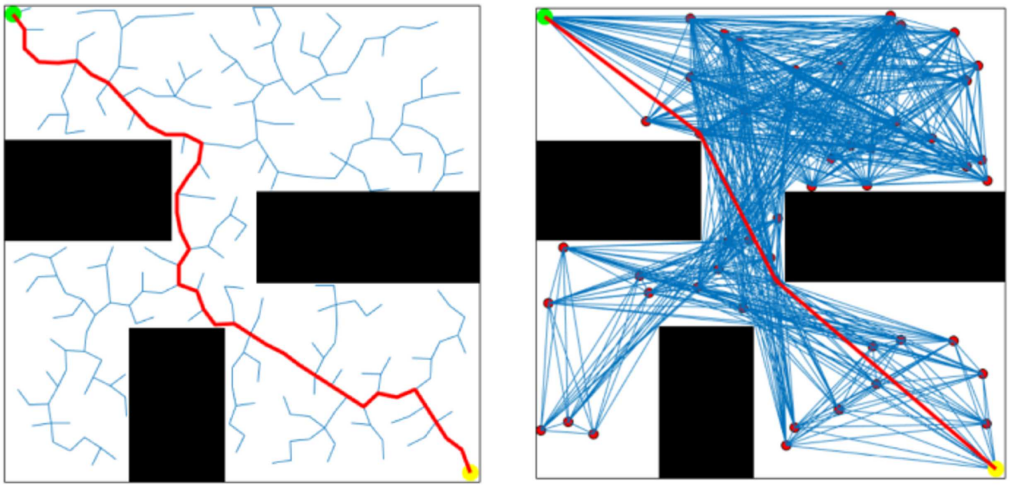


Figure 4.8: Graph search algorithms based on sampling. The green, yellow and red points denote the starting, ending and sampled points, respectively, and the red curve represents the global path. On the left, the performance of the RRT is shown, while on the right, the path planned with the PRM is visualized [95].

graphic search techniques. The comparison of these two algorithms is shown in Fig. 4.8, on the simple example for path planning.

4.3.1 Potential Exploration Targets Generation

A key advantage of sampling-based strategies over frontier-based strategies is their ability to navigate complex environments without the need for constant frontier detection and evaluation. However, this brings with it a number of challenges, particularly in ensuring the quality and optimality of the paths generated. While frontier-based methods provide a more global approach to exploration and focus on expanding the known territory, sampling-based methods provide a more local perspective of the environment. The choice between these two strategies often depends on the specific requirements and constraints of the exploration task, including the complexity of the environment, the available computational resources, and the desired balance between exploration efficiency and path optimization. This section aims to deepen the basics of the sampling-based strategy, the advantages and potential limitations of the sampling-based strategy and to provide a comprehensive understanding of its role in autonomous exploration systems.

As shown in Fig. 4.1, the exploration strategy inside the exploration system can easily be replaced. Instead of a frontier-based strategy, the autonomous exploration system can use the sampling-based strategy, ensuring that the odometry and OctoMap are received. In this thesis, a sampling-based strategy is studied that is based on the state-of-the-art Receding Horizon Next-Best-View planning (RH-NBVP) [6]. The proposed strategy uses a sampling-based approach to generate the potential exploration targets and select the next best viewpoint similar

to the frontier-based strategy (explained in Subsection 4.2.3). This planner is used in combination with RRT [43], [44] to generate traversable paths. The RRT algorithm samples nodes $\mathbf{n} = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$ (highlighted in yellow in Fig. 4.9). For each node in the RRT path, the information gain is calculated as a volume of the unmapped space that would be observed by robot sensors when the robot is positioned in the target node. The algorithm used for the information gain estimation is the Recursive Shadowcasting algorithm, introduced in Subsection 2.4.4. The best RRT path is then determined and executed.

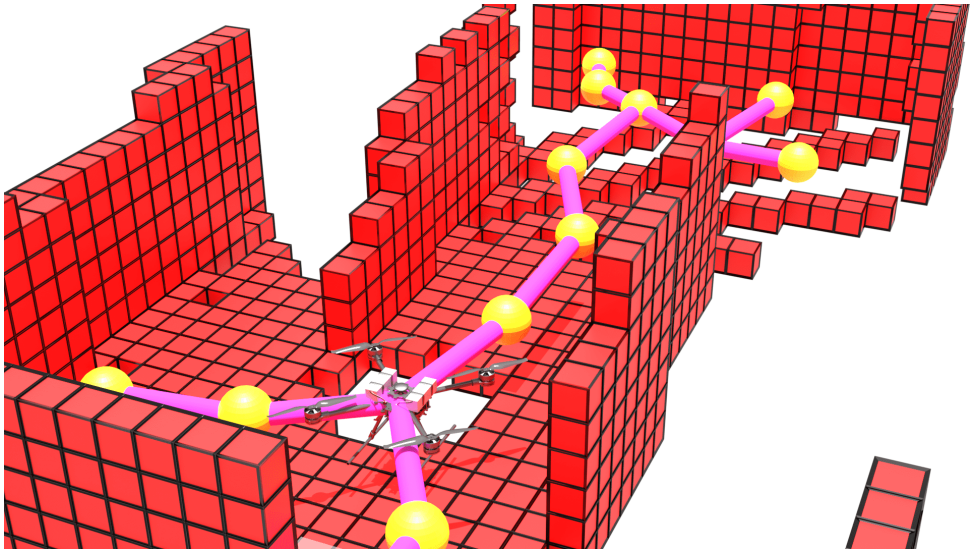


Figure 4.9: An illustration of the RRT node and path generation. Pink paths with yellow waypoints (nodes) are the result of the RRT. The origin of the RRT is at the position of the UAV. The OctoMap is shown in red for better visualization.

4.3.2 Path Evaluation and Selection

As mentioned above, the RRT algorithm samples nodes $\mathbf{n}$. A collision-free RRT path is denoted as $\mu \in M$, where M denotes the set of all RRT paths. Let μ_j, $j \in (1,2,...,N)$ be the path edge between nodes $\mathbf{n}_{k-1}$ and $\mathbf{n}_k$, where $k \in (1,2,...,N+1)$ and N is the number of nodes. For path edge μ_j, the information gain $I(\mu_j) \in \mathbb{R}$ is defined as a measure of an unexplored region of the environment that is potentially visible from the center $\mathbf{c}_k$ of this path edge.

To determine the information gain $I(\mu_j)$ using the RSC, first a cuboid around the edge μ_j is placed. The center and the size of this cuboid depend on the position and length of the edge as well as on the range of the sensors and the size of the environment. The dimensions of the cuboid are important as they define the area within the calculations are performed. The center of the cuboid serves as the focal point for the grid-based analysis and defines the boundaries of the study area. An illustration of the cuboid centered at the path edge is shown in Fig. 4.10. The illustration is simplified showing the performance of the RSC on the first 2D OctoMap slice. The RSC algorithm is performed on each slice and on each path edge. The principle is described

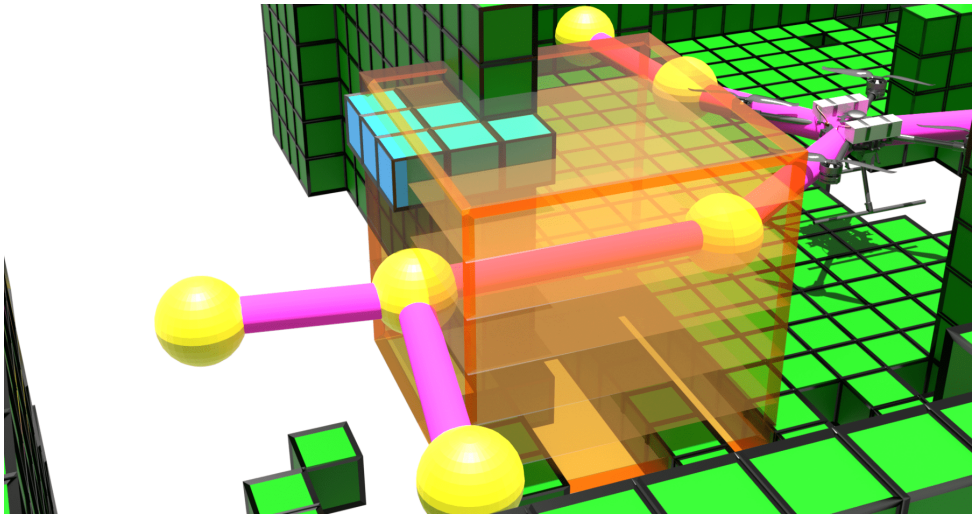


Figure 4.10: An illustration of the proposed Recursive Shadowcasting algorithm in the OctoMap. The algorithm is performed inside the cuboid centered at the path edge, on each 2D slice (planes inside the cuboid). The results of the RSC are shown on the first OctoMap slice, where the cyan voxels represent unknown voxels while the grey voxel is not taken into account for the information gain calculation.

in detail in Publication 2 in Chapter 8. When calculating the information gain, the length of the path is taken into account to avoid missing important data, especially for longer paths. If a path is too long, additional focus points are added to ensure complete coverage. The information gain for each node on the path is calculated by considering both the amount of newly covered area and the travel cost. In doing so, the importance of the new information against the cost of moving to new locations is weighted, giving more weight to the new information or the travel cost, depending on thespecific requirements. Namely, to form the information gain of the node $\mathbf{n}_k$, edge information gain $I(\mu_j)$ is weighted with the negative exponential of the cost to travel along the path up to $\mathbf{n}_k$, similar to the one proposed in [24] and used in [6]:

$$I(\mathbf{n}_k) = I(\mathbf{n}_{k-1}) + \frac{I(\mu_j)}{e^{\lambda L(\mathbf{n}_k, \mathbf{n}_{k-1})}}, \tag{4.7}$$

where λ is a positive constant while $L(\mathbf{n}_k, \mathbf{n}_{k-1})$ is Euclidean distance between nodes $\mathbf{n}_k$ and $\mathbf{n}_{k-1}$. The constant λ weighs the importance of the robot motion cost against the expected information gain. A small λ gives priority to the information gain, while $\lambda \to \infty$ means that the motion is so expensive that the shortest path is selected.

The goal in each step of this process is to find the path that maximizes the information gain. According to Eq. 4.7, path information gain $I(\mu)$ is equal to the information gain of the last node of the path and presents the volume of the unknown space that is covered along the path, combined with the cost of going there. In each iteration, thegoal is to find the best path μ_{bp}, which maximizes the information gain $I(\mu)$:

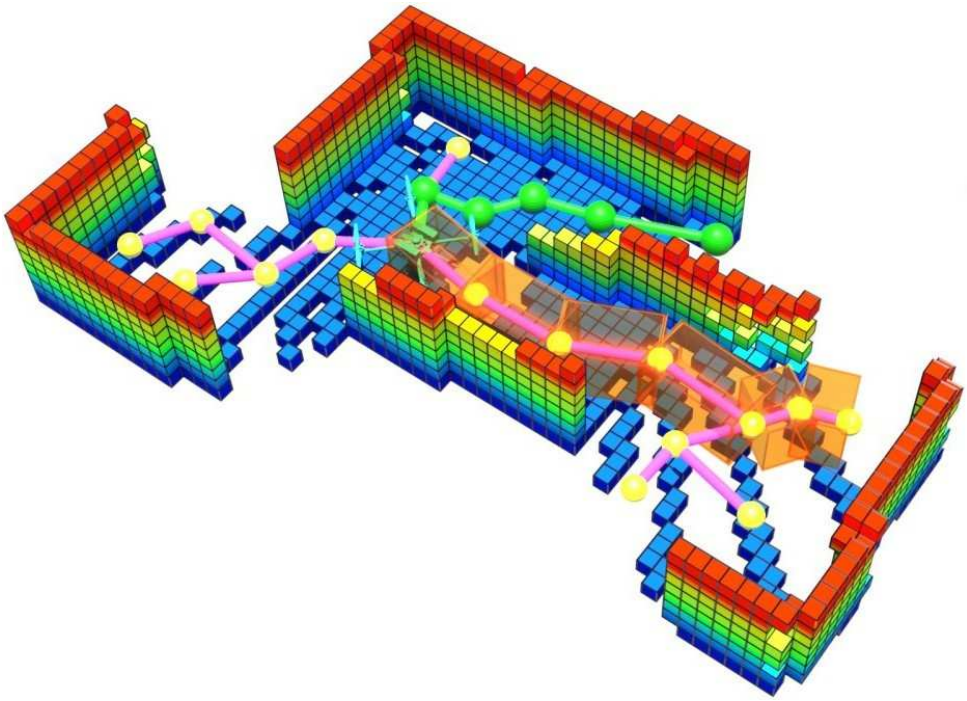


Figure 4.11: The UAV body (cyan color) denotes the first point of the path. Orange cuboids along the path of the RRT illustrate the volume where the information gain is computed. A green path with green waypoints is the best path. Pink paths with yellow waypoints are other paths of the RRT.

$$\mu_{bp} = \arg\max_{\mu \in M} I(\mu). \tag{4.8}$$

The principle of finding the best path is similar to the one used in frontier-based strategy, described in Subsection 4.2.3. The illustration of the path evaluation is shown in Fig. 4.11.

Once the best path is selected, the direction or yaw angle of thesensor is adjusted to ensure that it is aligned with the next point on the path. In the RH-NBVP the yaw angle is randomly sampled during the exploration, which limits the sample efficiency of the exploration. This limitation is briefly addressed in [32], [40]. Unlike other strategies, this strategy implemented in the thesis increases the efficiency of theexploration as it ensures that thesensor is always aligned with the most informative area.

After the path has been augmented with the yaw angle, it is forwarded to the trajectory planner. Within this thesis, the Time Optimal Path Parametrization by Reachability Analysis (TOPP-RA) algorithm, developed by [96], is used. Apart from the waypoints, inputs for the TOPP-RA are also velocity and acceleration constraints, which are maximally set to the UAV physical limitations. The planned trajectory is then executed by the UAV, and a new cycle for determining the best path is started after the UAV stops.

The probabilistic nature of sampling-based strategies not only ensures a thorough exploration but also gives them a degree of flexibility and adaptability that is very valuable in complex or uncertain environments. This makes sampling-based strategies particularly suitable for ap-

plications where the environmental conditions are unknown or not precisely defined in advance. However, it is important to note that while sampling-based strategies are superior in terms of coverage and adaptability, they sometimes lose efficiency compared to more targeted methods such as frontier-based strategies. While the random nature of path generation is advantageous for exploring unknown spaces, it can lead to redundancies, especially in simpler environments.

4.4 Semantically-Enhanced Exploration

Semantically-enhanced exploration refers to an exploration method in which a semantic understanding of the environment is incorporated into the exploration process. This approach goes beyond traditional exploration methods by understanding and interpreting the meaning or context of the objects and features within that environment. For example, a UAV using semantically-enhanced exploration can identify specific objects such as warehouse infrastructure or objects on a construction site, and understand their importance or function within the environment. This advanced understanding allows for more informed decision-making during exploration. The system can prioritize certain areas or objects for closer examination based on their semantic value, leading to more efficient and effective exploration strategies. In practice, this means that a semantically enhanced exploration system not only maps the physical layout of an area, but also creates a context-aware representation of the environment.

Within this thesis, a semantically-enhanced exploration strategy involves object detection using semantic segmentation algorithms in parallel with autonomous exploration. The result is a map of a previously unknown area with labeled objects of interest. By using information about the objects in the environment, theexploration system can effectively identify key regions with a high concentration of relevant objects, resulting in faster labeling of objects of interest. As the UAV navigates through the unknown area, it utilizes advanced algorithms for real-time data processing and decision-making. This allows the UAV to adapt its exploration strategy in flight and focus on the areas most likely to provide valuable insights. Semantic segmentation enables a deeper understanding of the environment, allowing the UAV to distinguish and categorize objects.

Typically, incorporating semantic information into the map requires having a specific goal or application in mind. Thus, within this thesis, the goal of the semantically-enhanced exploration strategy is to direct the exploration to quickly label all objects of interest on the map in the warehouse scenario and the construction site. The algorithm should enable real-time mapping, exploration, navigation, object detection and labeling in GPS-denied indoor environments. By leveraging onboard sensors and processing capabilities, a UAV can detect and identify objects of interest, such as equipment, products, or inventory, in real time, providing valuable information for the exploration algorithm. The integration of real-time localization, mapping and semantic

segmentation onboard a UAV equipped with an RGB-D camera is provided. The semantically segmented object from the image frame is projected on a 3D map of the environment. In this thesis, the proposed strategy takes advantage of the semantic information extracted in 3D so that a new utility function is introduced to guide the UAV toward the objects in the environment.

This section provides an overview of the segmentation of 2D images and the extraction of object poses, which is important for labeling objects in the 3D map. The chapter concludes with a focus on a novel utility function for frontier evaluation and selection.

4.4.1 2D Image-Based Sematic Segmentation

An input image from an RGB-D camera can be represented as a 2D array of pixel values. Semantic segmentation is a computer vision technique that involves labeling each pixel of the image with a specific class or category. The objective of semantic segmentation is to predict the segmentation map for the input image, but instead of containing pixel values, it contains the predicted semantic labels for each pixel. Each object of interest o_i corresponds to a collection of pixels that form a distinct entity that can be visually identified and distinguished from the background or other elements in the image. Then, using a semantic segmentation algorithm, the semantic labels s_i for each object o_i can be determined. In other words, the goal is to find a function $f : O \rightarrow S$ that maps each object to its corresponding semantic label. Given a number of semantic labels N_{labels}, the determination of semantic labels for each object can be expressed as:

$$f(o_i) = s_j \text{ for } 1 \leq i \leq N_{obj}, 1 \leq j \leq N_{labels}, \tag{4.9}$$

where o_i is the object, and s_j is a semantic label, from the set S and for the given object.

By utilizing deep learning models, this approach can accurately segment objects and regions of interest in 2D images. In this thesis, for semantic segmentation, the HRNet [86] is used, which is a recently proposed model that retains high-resolution representations throughout the model, without the traditional bottleneck design.

The HRNet model for the 2D image semantic segmentation is used since it showed enviable performance results [88]. Furthermore, it is compact, fast, robust and easy to use, enabling the model adaptation to work on CPU only, making it suitable for applications running on UAVs with limited computational resources. The model is trained on the ADE20K dataset with 150 objects and stuff classes included. ADE20K is a large open-source dataset for semantic segmentation and scene parsing, released by the MIT Computer Vision team [97], [98]. The ADE20K dataset is selected since some datasets have a limited number of objects (e.g., COCO [99], Pascal [100]) and in many cases those objects are not the most common objects encountered in indoor environments, or the datasets only cover a limited set of scenes (e.g., Cityscapes [101]). Additionally, objects of interest are extracted from the semantically segmented image.

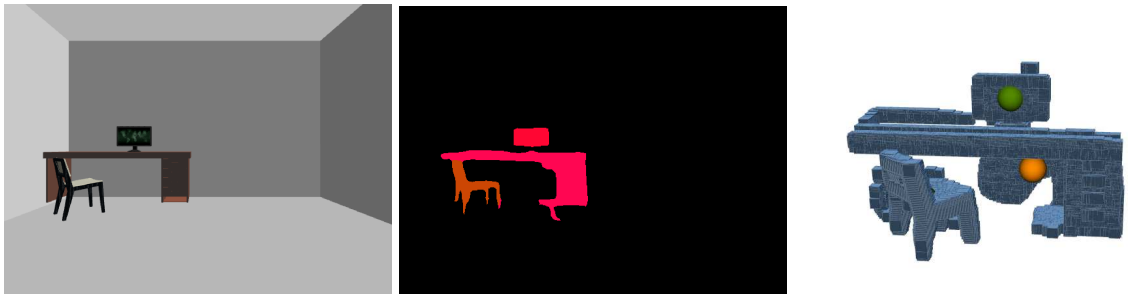


Figure 4.12: Semantic segmentation of simple features (table, chair and monitor). The semantic segmentation is performed with an HRNet model trained on the ADE20K dataset. Once the objects are semantically segmented, the 3D position of each object is extracted and colored in a specific color matched to the object.

A demonstration of semantic segmentation using the HRNet model trained on the ADE20K dataset is illustrated in Fig. 4.12 using a basic image featuring a table, chair, and monitor.

In addition to the HRNet model, another model used in this thesis is RetinaNet [102], which is known for its effectiveness in object detection. RetinaNet is characterized by its unique *focal loss function*, which addresses the problem of class imbalance in training, a common challenge in object detection. This feature makes RetinaNet particularly suitable for recognizing objects in diverse and complex environments, where the presence of many "easy" background examples can make learning the "harder" object examples more difficult.

In scenarios where more computational power is available, such as systems equipped with Nvidia Jetson or similar high-performance computing platforms, the integration of mmdetection or mmsegmentation packages [103] proves to be highly beneficial. Mmdetection provides an object detection toolbox that supports a wide range of detection models, while mmsegmentation offers a comprehensive suite for semantic segmentation tasks. Their high performance and flexibility make them ideal for demanding applications that require real-time processing and precise object recognition. Therefore, in this thesis, the HRNet model trained on ADE20K dataset for warehouse scenarios has been integrated, as well as RetinaNet model with a custom dataset for detecting objects on the construction site (Fig. 4.13). Testing two models with different approaches shows the adaptability of the system in using and exchanging detection algorithms to meet different exploration requirements.

4.4.2 Labeling Objects in the Map

Once the objects in a 2D image have been semantically segmented, the next step is to determine their 3D positions in the global frame for 3D exploration. To do this, both the object mask from the image and the point cloud data from the camera are used. The object mask provides a binary representation of the objects in the image and distinguishes the objects from the background. In addition, the point cloud from a camera, a collection of 3D points that represent the surface

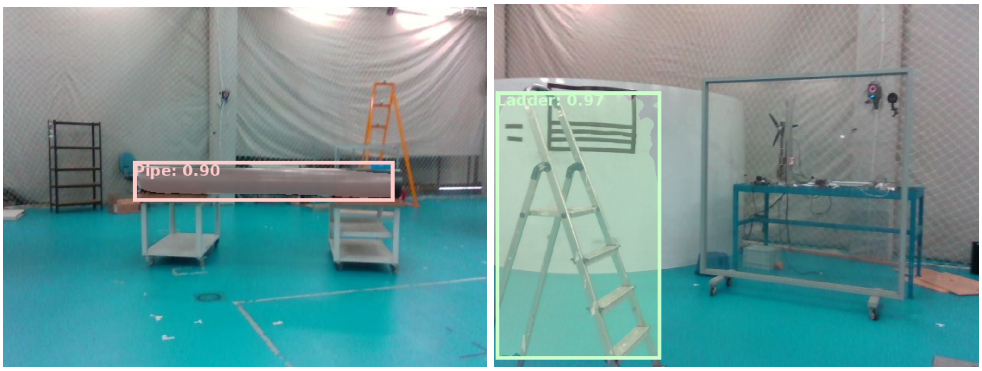


Figure 4.13: Results of the mmdetection algorithm using RetinaNet. The model, trained specifically for construction site environments, effectively detects pipes and ladders. The detection is performed in the real-world scenario.

geometry of the objects, is used.

In this process, the segmented objects in the image mask are matched with the corresponding points in the point cloud. For each object identified in the 2D image, a corresponding set of 3D points is identified from the RGB-D data of the camera that are aligned with the image mask. The 3D position of each object is then calculated using the centroid technique, which estimates the position based on the points in the point cloud, expressed in the coordinate system of the camera.

This matching of the 3D points with the 2D image coordinates helps to determine the position of each object in the coordinate system of the camera. Taking into account the state of the UAV in the global frame, the position of the object can be mapped onto the global frame. For this purpose, transformation matrices are used that define the position and orientation of the UAV as well as the fixed transformation between the UAV body and the camera. The detailed approach is described in Publication 3 in Chapter 8. This method effectively maps 2D image objects to their 3D positions in the world and enables accurate and robust extraction of object positions for 3D exploration.

4.4.3 Sematically-Aware Frontier Evaluation

In this thesis, the frontier-based strategy, described in Section 4.2, is combined with semantic information from the environment. The frontiers are evaluated by taking into account the information from the semantic segmentation algorithm.

Labeled objects of interest are included in the exploration policy, assuming that this leads to faster object labeling. To evaluate each voxel in F_c, the *total gain* of every candidate $\mathbf{v}_c \in F_c$ is defined using the following function:

$$G(\mathbf{v}_c) = \alpha I_{gg}(\mathbf{v}_c) + \beta I_{sg}(\mathbf{v}_c), \tag{4.10}$$

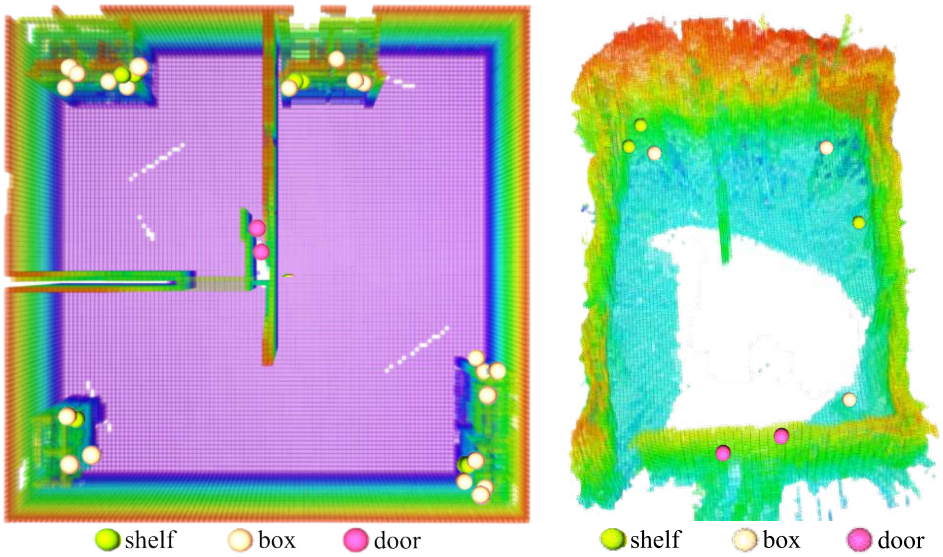


Figure 4.14: The OctoMap of the warehouse environment in the simulation and in the real-world experiment is shown on the left and right respectively. The transparency effect is applied on the OctoMap and detected objects during the exploration are shown. Shelves, boxes and doors are shown with different colors on the map.

where α and β are positive constants, while $I_{gg}(\mathbf{v}_c)$ and $I_{sg}(\mathbf{v}_c)$ represent *geometric information gain* and *semantic information gain* of each candidate $\mathbf{v}_c$, respectively. Therefore, α and β represent the trade-off between the geometric and semantic information gain. The values of α and β are experimentally determined and depend on the environment layout.

The geometric information gain $I_{gg}(\mathbf{v}_c)$ is defined using the function similar to the one proposed in [24]:

$$I_{gg}(\mathbf{v}_c) = \frac{I_{un}(\mathbf{v}_c)}{e^{\lambda L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}_c})}}, \tag{4.11}$$

where λ is positive constant, $L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}_c})$ is the distance between the robot's current position $\mathbf{p}_i$ and the position of the candidate $\mathbf{p}_{\mathbf{v}_c}$, while $I_{un}(\mathbf{v}_c)$ is information gain (explained in Subsection 2.4.2).

Similar to the frontier-based strategy, the estimated distance is approximated using the Euclidean distance between the robot position $\mathbf{p}_i$ and the position of the candidate (voxel center) $\mathbf{p}_{\mathbf{v}_c}$, $L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}_c}) = \|\mathbf{p}_i - \mathbf{p}_{\mathbf{v}_c}\|$. The constant λ weights the importance of robot motion cost against the expected information gain. A small λ gives priority to the information gain, while $\lambda \to \infty$ means that the motion is so expensive that only $\mathbf{v}_c$ near the robot is selected. To include semantically segmented objects from the environment in the exploration policy, $I_{sg}(\mathbf{v}_c)$ is introduced, as shown in Eq. 4.10. $I_{sg}(\mathbf{v}_c)$ represents the semantic information gain of each candidate. Let n_{obj} be the number of currently semantically segmented objects in the environment, then $I_{sg}(\mathbf{v}_c)$

is defined as:

$$I_{obj}(\mathbf{v}_c) = \begin{cases} \frac{1}{L(\mathbf{p}_{obj}, \mathbf{p}_{\mathbf{v}_c})} & \text{if } L(\mathbf{p}_{obj}, \mathbf{p}_{\mathbf{v}_c}) \leq I_{\text{range}}, \\ 0 & \text{otherwise,} \end{cases} \tag{4.12}$$

$$I_{sg}(\mathbf{v}_c) = \sum_{obj=1}^{n_{obj}} I_{obj}(\mathbf{v}_c), \tag{4.13}$$

where $L(\mathbf{p}_{obj}, \mathbf{p}_{\mathbf{v}_c})$ is the distance between the position of the object $\mathbf{p}_{obj}$ and the position of the candidate $\mathbf{p}_{\mathbf{v}_c}$. Position of the object $\mathbf{p}_{obj}$ is calculated as stated in Subsection 4.4.2. In other words, the semantic information gain of each candidate $\mathbf{v}_c$ is the sum of all visible objects from the candidate $\mathbf{v}_c$ inversely proportional to the distance of the object.

Finally, the best frontier voxel is one that maximizes the total information gain $G(\mathbf{v}_c)$:

$$\mathbf{v}_{bf} = \underset{\mathbf{v}_c \in F_C}{\arg\max}\, G(\mathbf{v}_c). \tag{4.14}$$

The best frontier voxel $\mathbf{v}_{bf}$ is forwarded as a target point to the path planner. The result of semantically-enhanced exploration is an OctoMap with labeled objects of interest as shown in Fig. 4.14.

CHAPTER 5

Conclusion

In summary, this thesis focuses on the main concepts for autonomous exploration using unmanned aerial vehicles. The concepts include mapping, exploration, localization and path planning. Mapping involves translating sensor data into a map that contains useful information such as the location of obstacles and the position of objects. Autonomous map creation significantly improves the utility of a robot, enabling it to operate in environments without prior information. Mapping during exploration allows a robot to autonomously create a map of an unknown environment. The exploration aspect is tailored to the specific goal of the exploration task, whether it is 3D reconstruction, search and rescue or other applications. Localization is fundamental to autonomous navigation. It involves determining the position and orientation of a robot in its environment using various sensors and techniques such as SLAM. Finally, path planning is crucial for safe navigation. Taken together, these concepts form effective and efficient autonomous robotic systems that can operate in complex environments. The methods are designed to be adaptable to GPS-denied environments. They can include SLAM algorithms or precise localization techniques as well as various sensors and are therefore suitable for both outdoor and indoor exploration.

The novel elements of the contribution of this thesis are related to the exploration strategies. First and foremost is the development of a method for planning autonomous 3D exploration based on frontier detection. This method presents a novel frontier detection and multi-resolution frontier clustering approach. The approach improves the exploration process by using submap-based frontier detection and refining these frontiers at multiple resolutions. The effectiveness of this method is particularly notable in large or complex environments where efficient exploration is a challenge. The developed multi-resolution clustering algorithm is characterized by fast processing, robust performance and easy implementation and is adaptable to different

map resolutions and exploration depths. Furthermore, the thesis demonstrates the advantages of the multi-resolution frontier clustering approach through various simulations and real-world tests. These tests highlight the effectiveness of the method in optimizing the exploration path and reducing the time and resources required to explore large and complex environments. The adaptability and scalability of the method make it a robust solution for a wide range of applications.

Secondly, the introduction of a sampling-based approach using the Recursive Shadowcasting algorithm to efficiently estimate information gain is a novel approach. This method revolutionizes the process of planning exploration paths and enables a fast and thorough coverage of the environment with minimal computational effort. The ability to calculate information gain quickly and accurately is a key advance for onboard processing in robotic systems. What is more, the inherent probabilistic properties of sampling-based strategies provide comprehensive exploration coverage and give these methods considerable flexibility and adaptability. This is especially beneficial in complex scenarios and in environments where conditions are either uncertain or cannot be explicitly described before exploration.

Third, the thesis includes an integration of semantic map information into the exploration process. This method combines frontier-based exploration with semantic understanding and directs exploration to objects of interest. This approach proves especially advantageous for the efficient labeling of objects in 3D exploration, fostering a more comprehensive and detailed understanding of the environment. Semantically-enhanced autonomous exploration using UAVs represents a significant leap forward in the field of remote sensing and mapping. It combines state-of-the-art technologies for object detection and autonomous navigation to create highly detailed and accurate maps of previously unknown areas. This opens up new possibilities for exploration, analysis and strategic planning in various industries.

Moreover, these methods introduce novel calculations of information gain and the selection of the next best goal. The real-world applicability and effectiveness of these methods have been thoroughly validated through a series of realistic, challenging simulation experiments and real-world tests. These validations not only demonstrate the practical utility of the methods but also highlight their adaptability and reliability in the real world.

All strategies and methods developed and used in the implementation of the autonomous exploration system are modular. This modularity ensures that individual components can be easily expanded or replaced as required and offers a high degree of flexibility and scalability. For example, the localization, mapping, exploration strategy or navigation algorithm can be easily extended or replaced with a different approach if required. Every element of the system has been thoroughly validated, not only in simulated environments but also under real-world conditions. In addition, the entire system and the methods it contains have been made available as open source, so that the research and development community has the opportunity to further

develop and customize the system.

Overall, the contribution presented in this thesis not only addresses the immediate challenges in the field of autonomous robotic exploration, but also sets new standards for future research. The methods developed here open up new avenues for exploration and pave the way for more advanced, efficient and intelligent robotic systems that can navigate autonomously and understand complex 3D environments.

CHAPTER 6

List of Publications

Pub1 Batinović, A., Petrović, T., Ivanović, A., Petric, F., and Bogdan, S. (2021). A Multi-Resolution Frontier-Based Planner for Autonomous 3D Exploration. IEEE Robotics and Automation Letters, 6(3), 4528–4535.

Pub2 Batinović, A., Ivanović, A., Petrović, T., and Bogdan, S. (2022). A Shadowcasting-Based Next-Best-View Planner for Autonomous 3D Exploration. IEEE Robotics and Automation Letters, 7(2), 2969–2976.

Pub3 Milas, A., Ivanovic, A., and Petrovic, T. (2023). ASEP: An Autonomous Semantic Exploration Planner With Object Labeling. IEEE Access, 11, 107169-107183.

CHAPTER 7

Author's Contributions to the Publications

The results presented in this thesis are based on the research carried out in the Laboratory for robotics and intelligent control systems (LARICS), lead by Professors Zdenko Kovacic and Stjepan Bogdan, at the University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia during the period of 2019 - 2023. The thesis includes four publications written in collaboration with co-authors of the published papers. The author's contribution to each paper consists of the method design, software implementation, testing in simulations and real-world experiments, result analysis and written presentation.

Pub1 In the paper entitled *A Multi-Resolution Frontier-Based Planner for Autonomous 3D Exploration* the authors have developed a novel 3D exploration planner, optimized for state-of-the-art 3D sensors like LiDARs. The planner is designed to efficiently process the extensive point clouds generated by these sensors. The core of this planner is an algorithm for detecting frontier points, which represent the boundary between the explored and unexplored areas of an environment. This detection is followed by a novel process for clustering these frontier points and selecting the most suitable point for efficient exploration. One of the main advantages of this planner over existing frontier-based approaches is its scalability. It requires less processing time for the same size of the environment with comparable exploration times. This improved performance is achieved by using data from a mapping algorithm rather than directly from the 3D sensor. The planner takes advantage of the properties of the octree environment representation, enabling efficient analysis at different resolutions. The author's work contains several groundbreaking elements that

make a significant contribution to the field:

- The introduction of a submap-based method for detecting frontiers.
- The implementation of a multi-resolution approach to refine frontiers.
- The development of an innovative technique for selecting the best frontier point that incorporates a cube-based sensor approximation for fast calculation of potential information gain.

Compared to the current state-of-the-art methods, this approach provides faster and more efficient performance in both computation time and environmental coverage speed. The effectiveness and improved performance of the proposed approach are demonstrated by detailed simulations and experimental results. In addition, the author has made great efforts to improve the accessibility and reproducibility of this research. This is reflected in the publication of the datasets used in this study (including source code and maps), which allow direct comparison with other research in this area. This openness not only underlines the practical applicability of the planner, but also invites collaborative further development and validation within the research community.

Pub2 In the paper entitled *A Shadowcasting-Based Next-Best-View Planner for Autonomous 3D Exploration* the authors deal with the challenge of autonomous exploration of unknown environments using an aerial robot equipped with sensors such as LiDARs. The main goal is to efficiently explore an area, plan paths and calculate the information gain in a short time, so that implementation on an onboard computer is possible. An important innovation presented in this paper is a planner that randomly samples viewpoints in the environment. This planner uses a novel and efficient method for computing the information gain based on the Recursive Shadowcasting algorithm. To determine the Next-Best-View (NBV), an evaluation method based on cuboids is used, which significantly reduces the calculation time. In addition, the planner includes a dead-end resolution strategy that improves its ability to recover quickly in complex environments, minimizing overall exploration time. Comparative experiments in simulations show that this approach outperforms current state-of-the-art methods in terms of computational efficiency and total exploration time. The results consistently show that the proposed method achieves full exploration faster and with remarkably less computational time compared to other leading methods. The contributions of this paper are significant and threefold:

- The development of a time-efficient technique for estimating information gain using the Recursive Shadowcasting algorithm.
- The implementation of a cuboid-based method to evaluate the information gain at each RRT edge.
- The introduction of a history tracking method specifically designed to resolve dead-

ends during exploration.

Extensive simulation analyzes and comparisons with state-of-the-art approaches were carried out. To further emphasize the commitment to research transparency and collaboration, the authors have made the source code, simulation datasets and experimental data used in this study openly available. This initiative not only validates the results, but also facilitates future comparative research in the field of autonomous exploration.

Pub3 In the paper entitled *ASEP: An Autonomous Semantic Exploration Planner with Object Labeling* the authors present the Autonomous Semantic Exploration Planner (ASEP), an autonomous 3D exploration planner tailored to GPS-denied environments. ASEP uniquely integrates real-time mapping, exploration, navigation, object detection and object labeling capabilities for a UAV with limited resources. ASEP is based on a frontier exploration strategy that uses semantic information to determine its exploration policy. Policy is enhanced by the integration of geometric and semantic data coming from a deep convolutional neural network (DCNN) used for semantic segmentation. This innovative approach to semantically-enhanced exploration focuses on the rapid identification and labeling of all relevant objects in the environment. In addition, the planner includes an advanced path planning algorithm that continuously checks the feasibility of the path, enabling safe and efficient navigation even in complex environments. An important aspect of ASEP is its modular design, which allows easy expansion or replacement of its components with customized modules. The effectiveness of ASEP has been thoroughly tested both in simulation and in real tests with a UAV. The experimental results confirm the performance of ASEP compared to existing state-of-the-art methods. In particular, the results show an accelerated exploration of objects in the environment, a reduction of the overall exploration time and a consistent computational efficiency, even when considering the requirements of semantic segmentation. The main contributions of this work are as follows:

- Development of an exploration strategy that integrates a frontier-based method with semantic utility functions, enabling iterative exploration and labeling of objects of interest in unknown environments. This includes extending the information gain function to consider both geometric and semantic environment data.
- Implementation of real-time 3D object labeling during exploration, which includes the extraction of semantically segmented objects from 2D images and the processing of camera depth point cloud data to determine object locations in the environment.
- Development of a comprehensive system for mapping, exploration, path planning and navigation that is both modular and adaptable and can operate using low-cost sensors on a UAV with limited resources. This system provides efficient exploration

and object labeling while ensuring the safety and reliability of navigation.

This work represents a significant advance in autonomous exploration planning, offering improvements in exploration efficiency and object labeling accuracy, especially in challenging indoor environments.

CHAPTER 8

Publications

Publication 1: A Multi-Resolution Frontier-Based Planner for Autonomous 3D Exploration

Batinović, A., Petrović, T., Ivanović, A., Petric, F., and Bogdan, S. (2021). A Multi-Resolution Frontier-Based Planner for Autonomous 3D Exploration. IEEE Robotics and Automation Letters, 6(3), 4528–4535.

A Multi-Resolution Frontier-Based Planner for Autonomous 3D Exploration

Ana Batinovic, Tamara Petrovic, *Member, IEEE*, Antun Ivanovic, *Student Member, IEEE*, Frano Petric, *Member, IEEE*, and Stjepan Bogdan, *Senior Member, IEEE*

***Abstract*—In this letter we propose a planner for 3D exploration that is suitable for applications using state-of-the-art 3D sensors, such as LiDARs, that produce large point clouds with each scan. The planner is based on the detection of a frontier - a boundary between the explored and the unknown part of the environment - and consists of the algorithm for detecting frontier points, followed by the clustering of frontier points and the selection of the best frontier point to be explored. Compared to existing frontier-based approaches, the planner is more scalable, i.e., it requires less time for the same environment size while ensuring similar exploration time. The performance is achieved by relying not on data obtained directly from the 3D sensor, but on data obtained by a mapping algorithm. In order to cluster the frontier points, we exploit the properties of the Octree environment representation, which allows easy analysis with different resolutions. The planner is tested in the simulation environment and in an outdoor test area with a UAV equipped with a LiDAR sensor. The results show the advantages of the approach compared to current state-of-the-art approaches.**

***Index Terms*—Aerial systems, perception and autonomy, autonomous agents.**

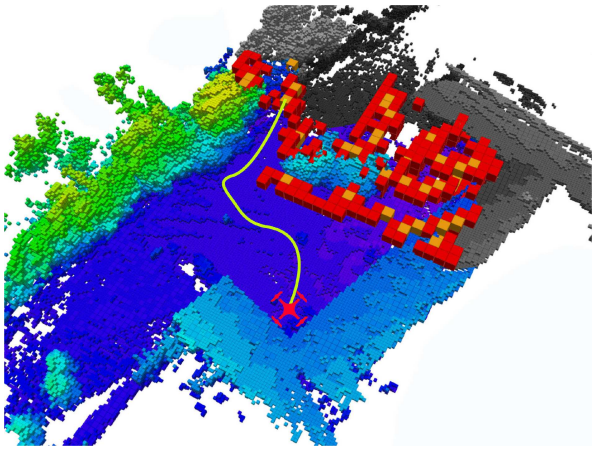


Fig. 1. A snapshot of the proposed method in action. Grey and white parts of the OctoMap represent the unexplored and green/blue explored environment. Red voxels represent the frontier and yellow voxels are centroids of frontier point clusters. A UAV explores the environment by planning a trajectory towards the selected cluster centroid.

I. Introduction

AN AUTONOMOUS exploration and mapping process is one of the fundamental tasks of robotics. Typical exploration methods are based on frontiers [1] and are used in both 2D and 3D space. In contrast to 2D exploration and mapping strategies, mapping large environments in 3D requires a large amount of memory and computational effort. Therefore the fastest possible generation of a complete 3D map and autonomous navigation of a robot through the map is a challenging task. This task is used to inspect civil infrastructure [2] or to search for people or objects, such as in search and rescue scenarios [3]. Furthermore, the fastest possible exploration could be required when operating with an energy-constrained robot [4].

The main objective of this letter is to develop a 3D exploration planner capable of meeting the above challenges. The letter focuses on large unknown environments where a robot should navigate autonomously without any a priori knowledge of the environment. The planner, which consists of a sequence of algorithms, acts as a decision making tool that guides the robot to the next exploration point. A snapshot of the proposed method in action is shown in Fig. 1.

We use the Google Cartographer simultaneous localization and mapping (SLAM) algorithm [5] as the basis for a novel exploration planner. For detection of the frontier, which is the first step of the exploration planning procedure, we use submap point cloud from Cartographer, which has one point for each occupied submap cell. A submap is created from a sequence of sensor scans by scan matching, fusion with IMU and odometry, thus providing a more stable input for the frontier detection algorithm compared to published exploration methods ([6], [7], [8], [9]) which use raw 3D sensor readings.

Manuscript received October 15, 2020; accepted February 24, 2021. Date of publication March 25, 2021; date of current version April 12, 2021. This letter was recommended for publication by Associate Editor I. Sa and Editor P. Pounds upon evaluation of the reviewers' comments. This work was supported in part by the European Union through the European Regional Development Fund - The Competitiveness and Cohesion Operational Programme (KK.01.1.1.04.0041), in part by the project named Heterogeneous autonomous robotic system in viticulture and mariculture (HEKTOR), and in part by EU-H2020 CSA project AeRoTwin - Twinning coordination action for spreading excellence in Aerial Robotics, under Grant Agreement 810321. The work of doctoral student Ana Batinovic was supported in part by the "Young researchers' career development project and training of doctoral students" of the Croatian Science Foundation funded by the European Union from the European Social Fund. *(Corresponding author: Ana Batinovic.)*

The authors are with the University of Zagreb, Faculty of Electrical Engineering and Computing, LARICS Laboratory for Robotics and Intelligent Control Systems, Unska 3, 10000 Zagreb, Croatia (e-mail: ana.batinovic@fer.hr; tamara.petrovic@fer.hr; antun.ivanovic@fer.hr; frano.petric@fer.hr; stjepan.bogdan@fer.hr).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2021.3068923, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3068923

We convert the aforementioned submap point cloud into an OctoMap, which has become the standard in recent years due to its efficient memory and querying properties [10]. For a more efficient frontier detection, we exploit the structure of the OctoMap which allows us to easily query occupied voxels with different spatial resolutions. Points of the frontier are detected at the lowest OctoMap level (corresponding to the smallest Octree voxels), resulting in large number of points which would decrease the overall performance of the proposed algorithm. To mitigate this problem, we efficiently filter frontier points by changing the resolution level of the points in the OctoMap. The final frontier points processing is performed using the mean-shift clustering [11] and results in a significantly reduced number of frontier points to be considered in further steps. The best frontier point to be visited is determined by estimating the benefit of the information gathered by visiting a candidate frontier point. The exploration loop is closed with an autonomous navigation to the selected frontier point, using the map generated through the exploration for trajectory planning and localization.

This exploration procedure encompasses several novel elements that make up the contribution of this letter: a) a submap-based frontier detection; b) multi-resolution frontier refinement; c) the best frontier point selection using a cube-based sensor approximation for fast calculation of the potential information gain. Compared to state-of-the-art approaches, our approach contributes to fast and efficient performance in terms of both computation time and speed of the environment coverage. The validity and increased performance of the proposed approach is demonstrated through extensive analysis of simulation and experimental results. In addition, we focus our efforts on making our datasets (source, maps) available for comparison with other research approaches.

In Section II we give an overview of the state-of-the-art of 3D exploration methods and position our work in relation to them. Section III is the core of the letter and contains the details of the planner. The results of the simulations and experiments performed with a UAV and their analysis, are presented in Sections IV and V. The letter ends with a conclusion in Section VI.

II. Related work

There is a wealth of earlier work related to autonomous exploration, especially for 2D, but more recently also for 3D environments. The approaches can be roughly divided into frontier-based and next best view-based approaches, even though there is a significant overlap between categories. In this section we give an overview of techniques from each category, with a focus on selected frontier-based approaches for 3D environment such as the one proposed in this letter.

Characteristic of frontier-based approaches is exploration by approaching a selected point on the frontier between the explored and unexplored environment. This idea was first introduced by Yamauchi in [1] and tested in a 2D environment. The simplest approach to 3D exploration is to use 2D frontier-based exploration with 3D maps at different heights (sometimes called 2.5D approaches) [12], [13]. A complete frontier-based solution for 3D environments is developed in [6] and [7], and these approaches are described in more detail later in this section.

Next best view-based (NBV) approaches aim to determine a (minimal) sequence of robot (sensor) viewpoints in the environment to be visited until the entire search space is explored. Potential viewpoints are usually sampled, e.g. near the frontier or randomly. Then the viewpoints are checked for the potential information gain and the next best viewpoint is assigned. One of the first NBV methods is presented in [14] and then extended in [8], [9], [15] and others. In [9] the authors use an RRT-based search to direct a UAV to the unexplored region. The method showed good scaling properties and the ability to handle large spaces, but due to the characteristics of the RRT algorithms, the total exploration time could be much higher for some environments, as verified in [16], [17] and [18], and what we additionally confirmed in our experiments. The exploration times are later improved in [8]. Often NBV approaches are used to build a 3D object without any a priori information, as in [19] and [20].

Our approach was inspired by that of Zhu *et al.* [6], an exploration tool called 3D-FBET. This is a frontier-based tool that is performed in three phases, similar to those presented in this letter. The phases are 3D mapping, frontier detection in combination with a clustering algorithm, and the selection of the best frontier. Through experimental evaluation on different environments 3D-FBET showed several shortcomings. First, because the frontier detection is based on a subset of altered voxels (generated from the camera point cloud), which is highly variable, the obtained frontiers were noisy and not reliable. Furthermore, the resulting frontier presented only a local view and the clustering was not adapted to the environment. These problems led to a higher total exploration time. The authors provide the source code and the duration analysis for each phase, which facilitates comparison with the new approaches. We extend this approach to recognize not only local but also global frontiers, similar to Mannucci *et al.* [7]. Mannucci proposed a 3D exploration with two OctoMaps and two frontiers (local and global) with different resolutions. Global frontiers are assigned when the set of local frontiers is empty. Manucci evaluates the best frontier using a cost-utility approach, similar to [21]. Since maintaining two OctoMaps is a resource-intensive task, we use the properties of OctoMaps and implement a solution with multiple resolutions in a single OctoMap. An advantage of an OctoMap is also used in [22] to merge voxels with an equal state, which results in the reduction of obstacle detection calculations. Our 3D frontier detection is motivated by a dense 2D frontier method presented by Orsulic *et al.* [23], which has achieved good results in terms of wall time per frontier update. Together with multi-resolution clustering and appropriate target point selection, we are constructing a novel 3D exploration planner that accelerates the 3D exploration process.

Most related works design the exploration algorithm to minimize the total exploration time ([7], [9], [16]). Some of them take into consideration computation time [6], which plays an important role in an exploration process, such that a lower computation time as well as the next best goal selection algorithm lead to a lower total exploration time. However, we consider both

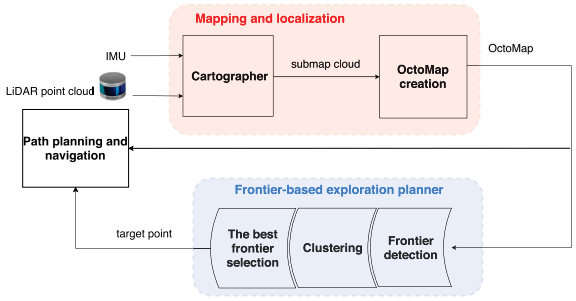


Fig. 2. Overall schematic diagram of autonomous 3D frontier-based exploration. The Cartographer SLAM creates 3D submaps, which are an input for the OctoMap generation module. Frontier detection, clustering and the module for selecting the best frontier voxels form the proposed 3D exploration planner (highlighted in blue). The best frontier voxel represents a target towards which the robot navigates.

the computation time and the total exploration time compared to the state-of-the-art approaches in Section IV-C and V-B. Our planner is able to run online and on board a robot with limited resources. Results are shown in simulations and experiments while datasets are provided for further use.

III. Proposed Approach

An autonomous exploration of either indoor or outdoor unknown 3D space $V \subset \mathbb{R}^3$, among the other objectives, creates a 3D map of the environment.

As a basis for our algorithm, we use an OctoMap, a hierarchical volumetric 3D representation of the environment. Each cube of the OctoMap is called a voxel (cell) $\mathbf{v}$, which can be *free*, *occupied* or *unknown*. Free voxels form the free space $V_{free} \subset V$, occupied voxels form the occupied space $V_{occ} \subset V$ and unknown voxels form the unknown space $V_{un} \subset V$. The entire space is a union of the three subspaces $V \equiv V_{free} \cup V_{occ} \cup V_{un}$.

The approach we propose is a frontier-based exploration, where the goal is to increase the overall knowledge of the environment by directing the robot to the *frontier point* with the best trade-off between benefit and cost. An overview of the proposed system is given in Fig. 2. The Cartographer SLAM algorithm requires an appropriate sensing system, e.g. a laser scanner or a camera to create a 3D map. An OctoMap is generated using the SLAM algorithm and is used for both frontier detection and collision-free navigation. Our planner is applicable to various types of autonomous robots equipped with LiDARs or other sensors that can be used to build an OctoMap. Note that our planner is not restricted to the Cartographer SLAM. As long as a reliable map builder is provided, multi-resolution frontier refinement and the best frontier selection modules can be applied.

A. Frontier Detection

A frontier, F, can be defined as a set of voxels $\mathbf{v}_f$ with the following property:

$$F = \{\mathbf{v}_f \in V_{free} : \exists neighbor(\mathbf{v}_f) \in V_{un}\}. \quad (1)$$

In other words, a frontier consists of free voxels with at least one unknown neighbor. The center of a frontier voxel is often called the frontier point. Since the space V is bounded, once the exploration is over the frontier becomes empty, $F = \emptyset$.

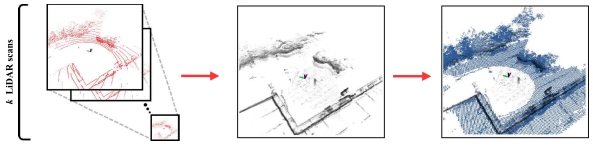


Fig. 3. Creating a submap cloud from n_s LiDAR scans (red) that are matched against a submap M_s (black) and a submap cloud M_{sc} is created at the end (blue).

As already mentioned, the OctoMap used for frontier detection is generated using Cartographer submaps. The submap M_s^i (i^{th} submap) is an occupancy grid map built using the last n_s consecutive LiDAR scans S, and matched with past IMU and odometry data:

$$M_s^i = f(S_{(i-1)n_s}, \ldots, S_{in_s}, IMU, Odometry), \quad (2)$$

where S_k denotes k^{th} LiDAR scan. Function f stands for a nonlinear optimization that aligns each successive scan against a submap being built. When the predetermined fixed number of scans n_s are inserted into a submap, it is marked as completed. Note that the size of a submap is adjustable, which makes the entire exploration process more robust. The map M can be created by joining all past submaps together:

$$M^i = f(M_s^1, \ldots, M_s^i). \quad (3)$$

Both the map M and the submaps M_s are in the form of a 3D occupancy grid. A format much more suitable for path planning and other operations are octrees, so instead of building a 3D occupancy grid map M we build an OctoMap O using the OctoMap generation software [10]. Namely, from each completed submap M_s^i we calculate a submap cloud M_{sc}^i by adding a point in the centre of each occupied voxel of M_s^i. The OctoMap O^i is then created from all the past submap clouds:

$$O^i = f(O^{i-1}, M_{sc}^i), O^0 = \emptyset. \quad (4)$$

The process of creating submap clouds from sensor scans is shown in Fig. 3. Due to optimizing the n_s laser scans to form a submap, submap clouds provide a more stable input for OctoMap generation compared to raw scans from the sensor, which enables more reliable detection of frontier points. Similar to FBET [6], our algorithm extracts the frontiers incrementally. Each time a new submap is created, the OctoMap is updated and a new frontier detection cycle is started. Our approach combines local and global frontiers, similar to Manucci [7]. With each new-coming submap cloud, a new OctoMap is created according to Eq. (4).

Let V_{free}^i and V_{free}^{i-1} correspond to the free voxels in two consecutive OctoMaps, O^i and O^{i-1}. The local frontier F_l, which contains only newly created frontier points can be calculated as follows:

$$F_l = \{\mathbf{v}_f \in V_{free}^i \setminus V_{free}^{i-1} : \exists neighbor(\mathbf{v}_f) \in V_{un}^i\}. \quad (5)$$

The global frontier F_g is a union of all past local frontiers, updated in each iteration and filtered to exclude voxels which

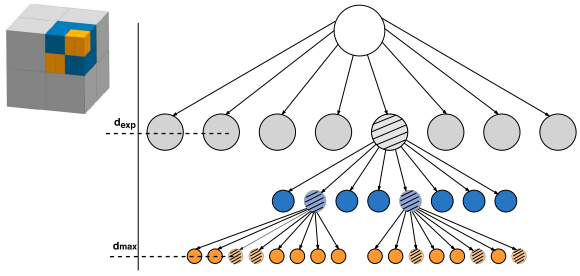


Fig. 4. The structure of an octree and the cube shaped space it represents. An instance of frontier voxels at d_{max} and their parents on the desired exploration depth d_{exp}.

do not satisfy the property Eq. (1) anymore. F_g is calculated as follows:

$$F_g^i = F_l^i \cup F_{gf}^i$$
$$F_{gf}^i = \{\mathbf{v}_f \in F_g^{i-1} : \exists neighbor(\mathbf{v}_f) \in V_{un}^i\}, F_g^0 = \emptyset. \quad (6)$$

There is usually a large number of voxels in the global frontier (referred to only as frontier from now on) and their evaluation is expensive in view of the computing effort involved. Therefore, we cluster the frontier voxels using both multi-resolution frontier and mean-shift clustering algorithms. This procedure leads to multiple clusters whose geometric central voxels are potential exploration targets.

B. Multi-Resolution Frontier Clustering

We exploit the Octree structure of the OctoMap to perform initial clustering of frontier voxels. The Octree recursively divides the volume and is a tree structure consisting of a node, or *OcTreeNode*, which has eight children (Fig. 4), which are also OcTreeNodes. The maximum depth of the OctoMap is $d_{\max} = 16$ [10]. The size of the voxel, expressed in meters, at this maximum level determines the level of detail of the OctoMap and is denoted with r_{max}.

In this work, the frontier is being detected at the OctoMap level d_{max}, but in general we can choose other levels for initial frontier detection. When making this decision, one should consider the expected structure of the environment, as the calculation on lower levels may artificially close corridors or narrow paths through the environment. The trade-off for calculation of the frontier on d_{max} is a large number of frontier points, which can cause unnecessary consumption of computational resources in later stages of the exploration planning procedure, especially if we focus on large outdoor environments. For that reason we aim to decrease the number of frontier points for future consideration while exploiting the multi-resolution properties of OctoMap for efficient frontier clustering.

We define the desired exploration level d_{exp} and the corresponding exploration voxel size r_{exp} based on the characteristics of the environment. If we expect more open areas, d_{exp} can be lower and r_{exp} can be larger. Frontier points clustered on the level d_{exp} are denoted as F_{exp} and determined as follows. Let us consider only four depth levels (as shown in Fig. 4), and let the frontier detection level be $d_{max} = 4$ and the desired exploration level be $d_{exp} = 2$. Then our goal is to find frontier parent OcTreeNodes from depth d_{exp} that are parents to the known frontier voxels $\mathbf{v}_f$ from d_{max}. A general expression for determination of the frontier points clustered at the exploration level in exploration planner iteration j (parent frontier voxels) F_{exp} is:

$$F_{exp}^j = \{\mathbf{v}_{exp}^j : \mathbf{v}_{exp}^j = parent(\mathbf{v}_f^j) \text{ at } d_{exp}\}, \forall \mathbf{v}_f^j \in F_g^j. \quad (7)$$

We use the superscript j in the previous equation to emphasize that the process of clustering is not performed for each newly built OctoMap, but only when the $j-th$ commanded waypoint, generated by the previous $(j-1)$ exploration planner iteration, is reached. The frontier is updated after each n_s LiDAR scans, i.e. when a new submap is created. The described multi-resolution clustering algorithm is fast and robust, easy to implement and suitable for different map resolutions and exploration depths. It can be applied to small and large areas. In our approach we combine it with the mean-shift clustering algorithm to direct a robot into the area where the frontier points are denser.

C. Mean-Shift Frontier Clustering

Clustering algorithms are often used in exploration (union-find algorithm, depth-first algorithm or flood fill clustering algorithm are used in [24], [6] and [7] respectively). In contrast to state-of-the-art approaches, we use the mean-shift clustering algorithm applied to 3D points. The mean-shift was first proposed by Fukunaga and Hostetler [25] and requires no assumptions on the form of the distribution or the number of clusters (compared to for example *K-means* [26]).

The input into the mean-shift clustering are the parent frontier voxels obtained in the previous step, F_{exp}. The output of the mean-shift clustering are frontier voxels which are candidates for becoming a next waypoint for the exploration and are denoted as F_c. The computationally most complex component of the mean-shift procedure is the identification of the neighbours of a point in 3D space (as defined by the kernel and its bandwidth). The kernel represents a weighting function and applying it to 3D points generates a probability surface (e.g., a density function). The kernel bandwidth regulates the size of the "window" over which the mean is calculated. In this letter, the Gaussian kernel is used. To make the mean-shift algorithm work in real time, along with the reduction in the number of global frontiers by multi-resolution frontier clustering, we carefully selected an appropriate bandwidth to balance between computation time and the desired outcome with respect to the size of the environment and r_{exp}.

D. Best Frontier Voxel Selection

To evaluate which of the voxels in F_c could result in a faster exploration of the environment, we define *total gain* of every candidate $\mathbf{v}_c \in F_c$ using the following function similar to the one proposed in [14]:

$$G(\mathbf{v}_c) = \frac{I(\mathbf{v}_c)}{e^{\lambda L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}_c})}}, \quad (8)$$

where λ is a positive constant, $L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}c})$ is the distance between the robot's current position $\mathbf{p}_i$ and the position of the candidate $\mathbf{p}_{\mathbf{v}c}$, while $I(\mathbf{v}_c)$ is a *information gain* i.e. a measure of the unexplored region of the environment that is potentially visible from $\mathbf{v}_c$. The estimated distance is approximated using the Euclidean distance between the robot position $\mathbf{p}_i$ and the position of the candidate (voxel center) $\mathbf{p}_{\mathbf{v}c}$, $L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}c}) = \|\mathbf{p}_i - \mathbf{p}_{\mathbf{v}c}\|$. The information gain $I(\mathbf{v}_c)$ is defined as the share of unknown voxels in a cube placed around $\mathbf{v}_c$. The size of the cube is defined with respect to the range of the used sensor. Often the information gain is estimated using a ray tracing algorithm and a real sensor field of view instead of using a cube-based approximation. By using the proposed simplification, the high calculation effort required by ray tracing is avoided.

The constant λ weights the importance of robot motion cost against the expected information gain. A small λ gives priority to the information gain, while $\lambda \to \infty$ means that the motion is so expensive that only $\mathbf{v}_c$ near the robot is selected. We can calculate the value of λ to be used based on the relative importance we set on the motion cost and the information gain. If we have two candidates $\mathbf{v}_{c1}$, $\mathbf{v}_{c2}$ and their information gains $I(\mathbf{v}_{c1})$, $I(\mathbf{v}_{c2})$, we can choose λ as follows:

$$\lambda = \frac{\ln(\frac{I(\mathbf{v}_{c2})}{I(\mathbf{v}_{c1})})}{L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}c2}) - L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}c1})}. \tag{9}$$

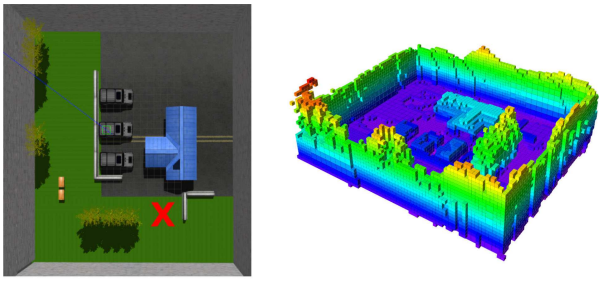
In this way, it is easy to set the ratio between the desired information gain and the distance with respect to the desired behavior of the system. For instance, if we want to prefer twice less information gain only if $L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}c2}) - L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}c1}) > 5$ m, we set λ to 0.1386. Finally, the best frontier voxel is one that maximizes the total information gain $G(\mathbf{v}_c)$:

$$\mathbf{v}_{bf} = \underset{\mathbf{v}_c \in F_C}{\arg\max}\, G(\mathbf{v}_c). \tag{10}$$

As soon as the best frontier point is selected, it is forwarded to a path planner as a waypoint. The robot starts to follow the planned path and navigates to the best frontier point $\mathbf{v}_{bf}$. For UAV control, we use an RRT-based path planner and trajectory following solution [27]. A new cycle of the procedure for determination of the best frontier is started after the previous waypoint is reached by the UAV, that is, the clusters and the candidate frontier voxels F_C are re-calculated. The exploration process is performed until the entire environment is explored and a complete map of the environment is created.

IV. Simulation-Based Evaluation

Simulations are performed in the Gazebo environment using Robot Operating System (ROS) and a model of the *Kopterworx* quadcopter, which is identical to the one used for experiments in the real world. More details about our system and the control structure can be found in our previous work [28]. The quadcopter is equipped with a Velodyne VLP-16 LiDAR sensor, whose maximum range is reduced to 20 m in the simulations. We set the maximum velocity of the UAV to 0.8 m/s, identical to the velocity used in outdoor experiments, which is low for safety reasons, however, it is comparable to the state-of-the-art settings



(a) (b)

Fig. 5. House exploration scenario. (a) Gazebo world. (b) OctoMap generated during exploration.

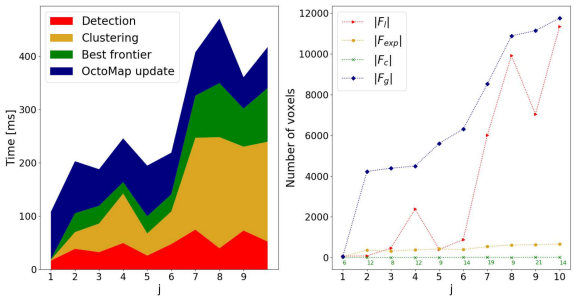


Fig. 6. House scenario metrics for $r_{max} = 0.25$ m, $r_{exp} = 1$ m, bandwidth $= 2$. On the left, the computation times for significant modules in each iteration j are shown. On the right, a graph shows the number of voxels in each iteration.

([7], [9], [18]), especially in the experimental analysis. In simulation and outdoor experiments, $n_s = 90$ while the side length of a cube used for information gain calculation is set to 5 m. We run two scenarios with different sizes and analyze the results. All simulations have been run 10 times on Intel(R)Core(TM) i7-8550U CPU @ 1.80 GHz $\times$ 8.

A. House Exploration Scenario

The first scenario refers to a 30 m $\times$ 40 m $\times$ 5 m space shown in Fig. 5. The vehicle starts from the marked position in the Gazebo world and navigates through the environment to explore the entire space. For the first scenario, the voxel size at the lowest resolution level is $r_{max} = 0.25$ m. For the exploration we use the level $d_{exp} = 14$, that is, voxel size $r_{exp} = 1$ m and mean-shift bandwidth of 2. The results are shown in Fig. 6.

The figure shows, for each iteration j of the waypoint calculation, the number of frontier voxels $|F_g|$, local frontier voxels $|F_l|$, the number of parent frontier voxels at the exploration depth $|F_{exp}|$ and the final target candidates $|F_c|$. Furthermore, the computation times for the significant modules (OctoMap creation, frontier detection, clustering and best frontier selection) are also given. Note that clustering includes both multi-resolution and mean-shift algorithms, while detection takes into account the time required to detect local frontiers and to update global frontiers. These modules take part in the total computation time, also calculated in each iteration. For this scenario, the average computation time is 0.197 s with a standard deviation

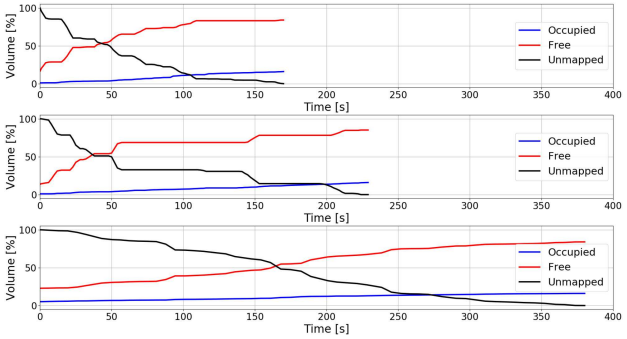


Fig. 7. House scenario - the percentage of free, occupied and unmapped volumes in total exploration time for the proposed planner, closest frontier method [1] and the NBVP [9], from top to bottom.

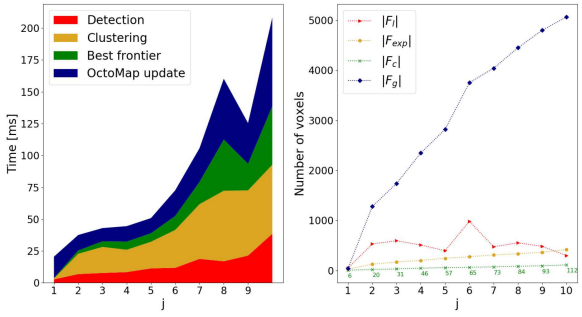


Fig. 8. Large scenario metrics for $r_{max} = 0.5$ m, $r_{exp} = 2$ m, bandwidth $= 2$. On the left, the computation times for significant modules in each iteration j are shown. On the right, a graph shows the number of voxels in each iteration.

0.062 s, which shows that our planner is suitable for real-time performance.

The proposed algorithm is compared with the closest frontier method [1] adapted to our planner as well as with the Next-Best-View planner (NBVP) presented in [9]. Parameters $d_{max}^{planner}$, λ, N_{max}, r and the maximum edge length of the RRT tree refer to the setup of NBVP explained in [9] and are set to 1.5 m, 0.25, 20, 0.25 m and 1.5 m respectively. We adapted the NBVP to our quadcopter, equipped with a LiDAR and to our control system to allow the fairest possible comparison. The percentage of free, occupied and unmapped volumes during exploration for all three planners is shown in Fig. 7. Note that for the proposed planner and the closest frontier method the total exploration time here includes the computation and execution time. For the NBVP, the total exploration time includes only the execution time, while the computation time for the given setup in each iteration was about 10 s. The graph shows that our method performs significantly faster than the closest frontier and the NBVP, taking less than 180 s to explore the house scenario. It is shown that combining the cube-based approximation of the information gain with the distance, instead of considering only the distance to the frontier point, results in faster exploration. The random sampling of the NBVP leads to regions revisiting, resulting in a higher total exploration time (380 s).

B. Large Exploration Scenario

The second scenario refers to a 130 m $\times$ 160 m $\times$ 5 m space, similar to the real world environment, as shown in Fig. 10 (a). In this scenario we set r_{max} to 0.5 m, d_{exp} to 14 and the bandwidth to 2. We can allow a lower resolution and exploration depth, as this scenario is larger than previous one. An instance of the large scenario exploration with the mentioned parameters is shown in Fig. 10 (b). Global frontiers (red) are clustered, resulting in candidates (yellow). The path is computed and the UAV navigates to the best frontier voxel (pink). The number of frontiers and the corresponding computation time during a single run are given in Fig. 8.

Even if the environment is larger, our planner needs on average only 0.095 s for the entire calculation time with a standard deviation of 0.058 s. In other words, our planner is also suitable for larger environments where the resolution may be lower. As shown in Fig. 9, the total exploration time is about 650 s. Note that the slope of the curve is different from that shown in Fig. 7. There are no sudden jumps in the explored volume because the environment is larger, but the LiDAR range is the same.

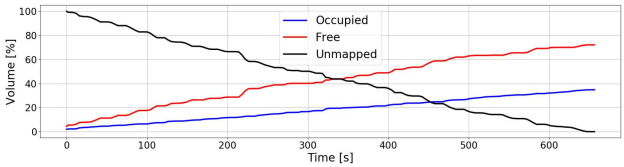
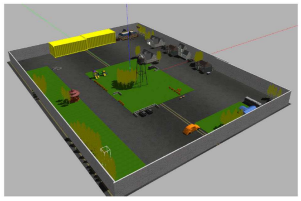
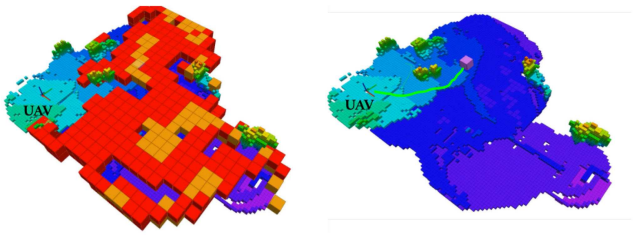


Fig. 9. Large scenario - the percentage of free, occupied and unmapped volumes over total exploration time.



(a)



(b)

Fig. 10. Large exploration scenario. (a) Gazebo world (b) An instance of the exploration process. Global frontiers are marked red, candidates yellow while the best frontier voxel $\mathbf{v}_{bf}$ is marked pink. The UAV planned a path (green) to the best frontier voxel (target).

Fig. 11. A custom built quadcopter equipped with a Velodyne $VLP-16$ LiDAR sensor.

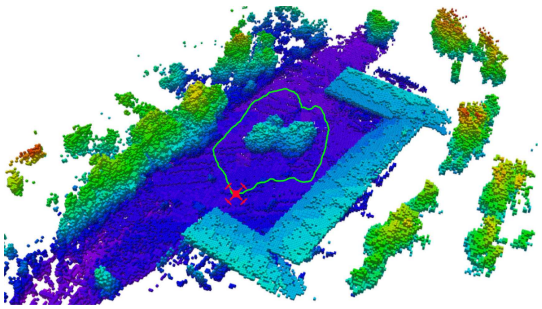


Fig. 12. The OctoMap created during the exploration of the real world scenario with the path traversed by the UAV during exploration.

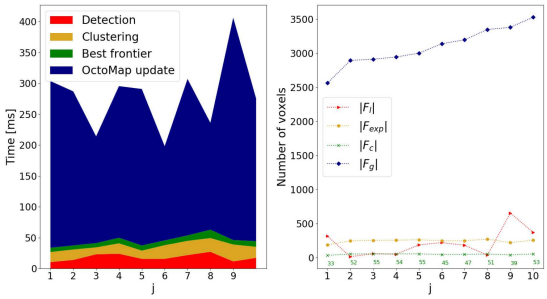


Fig. 13. Outdoor scenario metrics for $r_{max} = 0.5$ m, $r_{exp} = 2$, bandwidth $= 2$. On the left, the computation times for significant modules in each iteration j are shown. On the right, a graph shows the number of voxels in each iteration.

C. Simulation Results Discussion

Based on the simulation results shown in the last subsections, some general conclusions can be drawn which can be used in the design of an exploration system. First, frontier detection computation time and clustering time increase directly with the size of the global frontier, and we can vary the size of the global frontier by setting different values of r_{max}. Next, the computation time for the best frontier point depends directly on the number of candidate voxels F_c, and we can vary the number of candidate voxels for any r_{max} by changing the exploration level d_{exp} and the bandwidth parameter of the mean-shift. As expected, the OctoMap update time does not depend on the size of the frontier. The averages of the total computation time required for one exploration planner iteration (0.197 s, 0.095 s) allow the process to run even more frequently than in the current solution. The total computation time has an approximately linear relation to the size of the frontier, and the absolute values are suitable for the applications under consideration, so we can say that the approach scales well with the increase in the resolution and the number of frontier points. The simulations were performed for different initial UAV positions and similar numerical values were obtained. The comparison showed that the proposed planner is 25% and 53% faster than the closest frontier method and the NBVP, respectively, in terms of total exploration time. We mention the numerical results from [7] where the authors show results for an arena 100 m $\times$ 80 m $\times$ 7 m, stereo camera with a limited field of view and achieve a total exploration time of 1424 s using a single robot. Computation times are not given. Even though these numbers are better in our approach, we need to test solutions in the same setting to make further conclusions.

V. Experimental Evaluation

A. Setup

For our outdoor experimental analysis, we use a *Kopterworx* quadcopter (Fig. 11) which features four *T-motors* P60 KV170 motors attached to a carbon fiber frame. The dimensions of the UAV are 1.2 m $\times$ 1.2 m $\times$ 0.45 m, which makes it a relatively large UAV suitable for outdoor environments. The total flight time of the UAV is around 30 min with a mass of m $= 9.5$ kg, including batteries, electronics and sensory apparatus. The *Pixhawk 2.1* flight controller unit is attached to the center of the UAV body, and it is responsible for the low-level attitude control of the vehicle. Furthermore, we equipped the UAV with an *Intel NUC*, i7-8650U CPU @ 1.90GHz $\times$ 8, on-board computer for collecting and processing sensory data. The on-board computer runs *Linux Ubuntu 18.04* with *ROS Melodic* framework that communicates with the autopilot through a serial interface. The UAV is equipped with a Velodyne $VLP-16$ LiDAR sensor with a maximum range of 100 m. The maximum velocity of the UAV is limited to 0.8 m/s with a maximum acceleration of 0.5 m/s^2.

B. Results and Discussion

In the real world, we used the same parameters as in the large exploration scenario ($r_{max} = 0.5$ m, $d_{exp} = 14$ and the bandwidth $= 2$). Running the planner with the limited onboard resources and in real time, we were able to demonstrate fast exploration processing despite the large number of frontiers (Fig. 13). The OctoMap update time is much higher than in the simulations because the rate of the sensor is higher, however the frontier detection time, which depends on r_{max}, is similar to the times achieved in the simulation. The average calculation time is 0.343 s with a standard deviation of 0.043 s. Fig. 14 shows that the total exploration time is about 350 s. The result of the exploration is the OctoMap of the environment shown in Fig. 12, in which the path traversed by the UAV during the exploration is also shown.

A thorough comparison of the experimental results with other state-of-the-art approaches is not possible, due to the different environments, equipment and setup used. We briefly

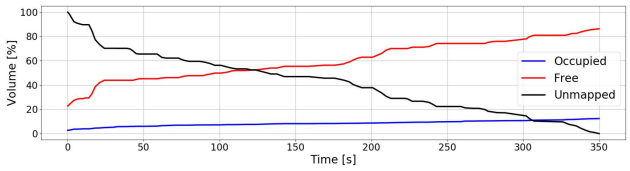


Fig. 14. The percentage of free, occupied and unmapped volumes in total exploration time for the experimental scenario.

state, for completeness, experimental results available in the previously mentioned state-of-the art approaches. In [9] authors experiment in a 9 m $\times$ 7 m $\times$ 2 m indoor arena with a MAV with $v_{max} = 0.25$ m/s and a stereo camera and the exploration finishes approximately after 250 s. Size of our outdoor arena is 50 m $\times$ 100 m $\times$ 4 m, and a UAV with $v_{max} = 0.8$ m/s finishes the exploration in 350 s. Regarding computation time, in [6] the authors use RGBD camera in an area of approximate size 10 m $\times$ 10 m, and obtain frontiers of size up to 200 cells. For these values, frontier detection takes about 18 ms, clustering around 1 ms and OctoMap update 0.5 s. The frontier sizes in our experiments are 1-2 orders of magnitude larger, but the total average computation time is similar, 58 ms. To showcase the reproducibility of our results and facilitate more thorough future comparisons in the exploration field of research, data sets of simulations and experiments carried out for preparation of this letter are available [29].

VI. Conclusion

This letter deals with a novel multi-resolution frontier-based planner. The planner is capable of autonomously exploring a previously unknown area, creating an occupancy grid map using Cartographer SLAM and generating an OctoMap. The results showed an improved behaviour in terms of exploration time compared to a state-of-the-art strategies. A robust frontier detection speeds up the exploration process, while a novel clustering algorithm ensures target evaluation in the real time. This 3D exploration planner has been successfully tested in simulation scenarios, as well as in a real world experiment, using a quadcopter equipped with a LiDAR. Video recordings of frontier-based exploration can be found at YouTube [30].

References

[1] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Automat. CIRA–97*, 1997, pp. 146–151.

[2] W. W. Greenwood, J. P. Lynch, and D. Zekkos, "Applications of uavs in civil infrastructure," *J. Infrastructure Syst.*, vol. 25, no. 2, 2019, Art. no. 04019002.

[3] H. Shakhatreh *et al.*, "Unmanned aerial vehicles (UAVs): A. survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.

[4] W. Bentz, T. Hoang, E. Bayasgalan, and D. Panagou, "Complete 3-D dynamic coverage in energy-constrained multi-UAV sensor networks," *Auton. Robots*, vol. 42, no. 4, pp. 825–851, 2017.

[5] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D lidar slam," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1271–1278.

[6] C. Zhu, R. Ding, M. Lin, and Y. Wu, "A 3D frontier-based exploration tool for mavs," in *Proc. IEEE 27th Int. Conf. Tools Artif. Intell.*, 2015, pp. 348–352.

[7] A. Mannucci, S. Nardi, and L. Pallottino, "Autonomous 3D exploration of large areas: A. cooperative frontier-based approach," in *Modelling Simul. Auton. Syst.*, vol. 10756, pp. 18–39, 2018.

[8] T. Baiming, S. Jicheng, D. Chaofan, and L. Qingbao, "A target point based MAV 3D exploration method," in *Proc. IEEE Int. Conf. Mechatronics Automat.*, 2018, pp. 2406–2413.

[9] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3D exploration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1462–1468.

[10] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: an efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[11] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.

[12] M. Al-khawaldah and A. Nüchter, "Multi-robot exploration and mapping with a rotating 3D scanner," *10th IFAC Symp. Robot Control*, vol. 45, no. 22, pp. 313–318, 2012.

[13] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unknown indoor environments," *Int. J. Micro Air Veh.*, vol. 1, no. 4, pp. 217–228, 2009.

[14] H. H. González-Baños and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *The Int. J. of Robot. Res.*, vol. 21, no. 10–11, pp. 829–848, 2002.

[15] D. Joho, C. Stachniss, P. Pfaff, and W. Burgard, "Autonomous exploration for 3D map learning," in *Proc. Autonome Mobile Syst.*, Springer, 2007, pp. 22–28.

[16] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A. frontier selection method for high speed flight," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 2135–2142.

[17] D. Deng, Z. Xu, W. Zhao, and K. Shimada, "Frontier-based automatic-differentiable information gain measure for robotic exploration of unknown 3D environments," https://arxiv.org/abs/2011.05288, 2020.

[18] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, "Fast frontier-based information-driven autonomous exploration with an MAV," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 9570–9576.

[19] J. I. Vasquez-Gomez, L. E. Sucar, R. Murrieta-Cid, and E. Lopez-Damian, "Volumetric next-best-view planning for 3D object reconstruction with positioning error," *Int. J. Adv. Robot. Syst.*, vol. 11, no. 10, p. 159, 2014.

[20] C. Dornhege and A. Kleiner, "A frontier-void-based approach for autonomous exploration in 3D," *Adv. Robot.*, vol. 27, no. 6, pp. 459–468, 2013.

[21] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 376–386, 2005.

[22] M. Faria, R. Marín, M. Popović, I. Maza, and A. Viguria, "Efficient lazy theta* path planning over a sparse grid to explore large 3D volumes with a multirotor UAV," *Sensors*, vol. 19, no. 1, p. 174, 2019.

[23] J. Oršulić, D. Miklić, and Z. Kovačić, "Efficient dense frontier detection for 2-D graph slam based on occupancy grid submaps," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3569–3576, 2019.

[24] C. Dornhege and A. Kleiner, "A frontier-void-based approach for autonomous exploration in 3D," in *Proc. IEEE Int. Symp. Safety, Security, Rescue Robot.*, 2011, pp. 351–356.

[25] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Inf. Theory*, vol. 21, no. 1, pp. 32–40, 1975.

[26] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley, 2001.

[27] B. Arbanas, A. Ivanovic, M. Car, M. Orsag, T. Petrovic, and S. Bogdan, "Decentralized planning and control for UAV-UGV cooperative teams," *Auton. Robots*, vol. 42, no. 8, pp. 1601–1618, 2018.

[28] R. Milijas, L. Markovic, A. Ivanovic, F. Petric, and S. Bogdan, "A comprehensive lidar-based slam comparison for control of unmanned aerial vehicles," [Online]. Available: https://arxiv.org/abs/2011.02306, 2020.

[29] A. Batinovic, "A Multi-Resolution Frontier-Based Planner for Autonomous 3D Exploration Dataset," [Online]. Available: https://github.com/larics/exploration-datasets

[30] "A Multi-Resolution Frontier-Based Planner for Autonomous 3D Exploration," [Online]. Available: https://www.youtube.com/playlist?list=PLC0C6uwoEQ8a88D6cKfa81Hfo_s_qVZxf

Publication 2: A Shadowcasting-Based Next-Best-View Planner for Autonomous 3D Exploration

Batinović, A., Ivanović, A., Petrović, T., and Bogdan, S. (2022). A Shadowcasting-Based Next-Best-View Planner for Autonomous 3D Exploration. IEEE Robotics and Automation Letters, 7(2), 2969–2976.

IEEE ROBOTICS AND AUTOMATION LETTERS, VOL. 7, NO. 2, APRIL 2022

A Shadowcasting-Based Next-Best-View Planner for Autonomous 3D Exploration

Ana Batinovic, *Graduate Student Member, IEEE*, Antun Ivanovic, *Student Member, IEEE*, Tamara Petrovic, *Member, IEEE*, and Stjepan Bogdan, *Senior Member, IEEE*

***Abstract*—In this letter, we address the problem of autonomous exploration of unknown environments with an aerial robot equipped with a sensory set that produces large point clouds, such as LiDARs. The main goal is to gradually explore an area while planning paths and calculating information gain in short computation time, suitable for implementation on an on-board computer. To this end, we present a planner that randomly samples viewpoints in the environment map. It relies on a novel and efficient gain calculation based on the Recursive Shadowcasting algorithm. To determine the Next-Best-View (NBV), our planner uses a cuboid-based evaluation method that results in an enviably short computation time. To reduce the overall exploration time, we also use a dead end resolving strategy that allows us to quickly recover from dead ends in a challenging environment. Comparative experiments in simulation have shown that our approach outperforms the current state-of-the-art in terms of computational efficiency and total exploration time. The video of our approach can be found at https://www.youtube.com/playlist?list =PLC0C6uwoEQ8ZDhny1VdmFXLeTQOSBibQl.**

***Index Terms*—Aerial systems, perception and autonomy, autonomous agents, mapping.**

I. Introduction

IN THIS letter, we study an autonomous exploration and mapping of a completely unknown 3D environment. We propose a novel method that improves upon the state-of-the-art Receding Horizon Next-Best-View planing (RH-NBVP) [1], which uses a sampling-based approach to select the next best viewpoint [2]. This planner is used in combination with Rapidly-exploring Random Trees (RRT) [3], [4] to generate traversable paths. For each node in the RRT path, the information gain is calculated as a volume of the unmapped space that would be observed by robot sensors when the robot is positioned in the target node. A common algorithm used for the information gain estimation is the Raycasting algorithm (RC) [5], and its results are then weighted by the cost of the robot travelling to the node. The best RRT path is then determined and the first edge is traversed before running a new iteration of the path planner.

The main drawbacks of the RH-NBVP are the significant computation time required to compute the information gain using the Raycasting algorithm and the high probability of ending up in a dead end state during the exploration. To overcome these issues, we propose a new strategy based on a Recursive Shadowcasting (RSC) algorithm, proposed in [6]. Since the RSC allows a much faster computation, we can estimate the information gain not only for each RRT node, but also for each RRT edge. We propose a cuboid-based evaluation for each RRT edge to obtain a more complete information about the unknown space to be discovered. We select the best RRT path and execute a trajectory through the RRT nodes of the best path. We extend the RH-NBVP to deal with sensors that produce large point clouds with each scan, such as LiDARs. Since we use a large point cloud, the RC in the information gain calculation process increases the computation time [1], [7], [8]. On the other hand, using LIDARs in combination with the RSC results in a significant computation time reduction during planning iterations.

The RRT has its root in the current position of the robot and is recomputed in each iteration. In large and narrow environments, as the explored area increases, the RRT algorithm might end up stuck in a dead end characterized by a significant increase in the distance to the node with a non-zero information gain and in the time required to sample valid RRT nodes. To address this drawback, motivated by the previous work on history-aware approaches to the 3D exploration ([7]–[9]), we developed a method to resolve such states by tracking previously visited RRT nodes.

We compare our method with the state-of-the-art methods in the simulation. The results show that in all cases our method achieves the complete exploration faster with an enviably low computation time. The contributions of this letter are summarized as follows:

- Time-efficient information gain estimation based on the Recursive Shadowcasting algorithm.
- Cuboid-based estimation of information gain on each RRT edge.

Manuscript received September 9, 2021; accepted January 10, 2022. Date of publication January 27, 2022; date of current version February 3, 2022. This letter was recommended for publication by Associate Editor B. Duncan and Editor P. Pounds upon evaluation of the reviewers' comments. The work of Ana Batinovic was supported by the Young Researchers Career Development Project–Training of Doctoral Students of the Croatian Science Foundation funded by the European Union from the European Social Fund. This work was supported in part by the European Union through the European Regional Development Fund-The Competitiveness and Cohesion Operational Programme through Project Heterogeneous autonomous robotic system in viticulture and mariculture (HEKTOR) under Grant KK.01.1.1.04.0041 and in part by EU-H2020 CSA Project AeRoTwin - Twinning coordination action for spreading excellence in Aerial Robotics under Grant Agreement Number 810321. *(Corresponding author: Ana Batinovic.)*

The authors are with the University of Zagreb, Faculty of Electrical Engineering and Computing, LARICS Laboratory for Robotics and Intelligent Control Systems, Zagreb 10000, Croatia (e-mail: ana.batinovic@fer.hr; antun.ivanovic@fer.hr; tamara.petrovic@fer.hr; stjepan.bogdan@fer.hr).

Digital Object Identifier 10.1109/LRA.2022.3146586

- A history tracking method for resolving dead end states.

To validate our contributions, we performed an extensive simulation analysis and comparison with the state-of-the-art approaches. Furthermore, to encourage the reproduction of our results and facilitate more thorough future comparisons in the exploration field of research, the source code, data sets of simulations and experiments carried out for preparation of this letter are available at [10].

In Section II we give an overview of the state-of-the-art of 3D exploration methods and position our work in relation to them. Section III is the core of the letter and contains details of the proposed planner. Results of simulations performed with a Unmanned Aerial Vehicle (UAV) and their analysis are presented in Section IV. The letter ends with a conclusion in Section V.

II. Related Work

Autonomous exploration and mapping is one of the fundamental tasks of robotics. Typical exploration approaches can be roughly classified into frontier-based, sampling-based, and hybrid strategies.

Characteristic of frontier-based approaches is exploration by approaching a selected point on the frontier between the explored and unexplored environments. This idea was first introduced by Yamauchi in [11] and subsequently evaluated in more detail in [12]. In each iteration, the next best goal is a frontier point closest to the robot. Similarly, in [13], the next best goal is the frontier that minimizes the velocity change to maintain a consistently high flight speed. It is shown that this approach outperforms the closest frontier method [11]. Frontier-based exploration approaches for 3D environments are also researched in [14]–[19].

Sampling-based approaches aim to determine a (minimal) sequence of robot (sensor) viewpoints to visit in the environment until the entire space is explored. Potential viewpoints are typically sampled, e.g., near the frontier or randomly. Then these viewpoints are evaluated for the potential information gain and the next best viewpoint is assigned. One of the first sampling-based methods is presented in [2] and then extended in [1], [20], [21]. In [1], authors proposed the RH-NBVP, which uses an RRT-based search [3], [4] to guide a UAV into the unexplored area. While the method showed good scaling properties and performance in a local exploration, it is not resilient to dead ends, resulting in a poor global scene coverage and thus, a high overall exploration time, as shown in [13], [17], [22], and in our previous work [16]. To address the drawbacks of the RH-NBVP, Witting *et al.* [9] introduced several modifications: memorizing previously visited locations; local gain optimization; and trajectory optimization, resulting in faster exploration. In [23], the authors improve the efficiency of the RH-NBVP by continuously growing a single tree and only sporadically querying feasible paths.

Hybrid strategies combine the advantages of both frontier-based and sampling-based approaches. Selin *et al.* [7] successfully combine the RH-NBVP with conventional frontier reasoning to compensate for a poor performance in the global exploration. In other words, [7] plans global paths towards frontiers and samples paths locally. Meng *et al.* [24] samples viewpoints around frontiers and finds the global shortest tour passing through them. Similarly, Respall *et al.* [8] samples viewpoints in the vicinity of a point of interest near a frontier and additionally memorizes nodes that indicate regions of interest in a history graph to reduce the gain calculation time. Besides, more recent efforts in this domain have focused on multi-objective planning [25], [26], informative path planning [27] or utilizing supervised and reinforcement learning for the problem of exploration planning [28], [29]. Motivated by advances in the literature and the fact that sampling-based algorithms have been shown to provide good behavior in local exploration, but typically require computationally expensive information gain calculation (e.g., the RH-NBVP), this letter presents a novel exploration planner that aims to provide shorter computation times and faster exploration.

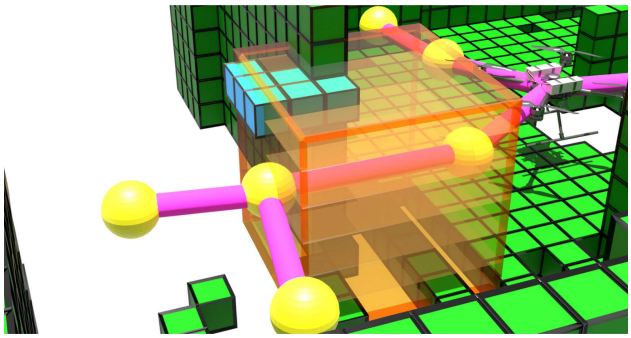


Fig. 1. An illustration of the proposed Recursive Shadowcasting algorithm in the OctoMap. The algorithm is performed inside the cuboid centered at the path edge, on each 2D slice (planes inside the cuboid). The results of the RSC are shown on the first OctoMap slice, where the cyan voxels represents unknown voxels while the grey voxel is not taken into account for the information gain calculation.

III. Proposed Approach

A. System Overview

The main goal of our approach is to explore a bounded and previously unknown 3D space $V \subset \mathbb{R}^3$. As a basis for our approach we use an OctoMap, a hierarchical volumetric 3D representation of the environment [30]. Each cube of the OctoMap is denoted as a voxel (cell), which can be *free*, *occupied* or *unknown*. Free voxels form the free space $V_{free} \subset V$, occupied voxels form the occupied space $V_{occ} \subset V$ and unknown voxels form the unknown space $V_{un} \subset V$. Initially, the entire bounded space is unknown, $V \equiv V_{un}$, and the unknown space decreases as the exploration advances. The entire space is a union of the three subspaces $V \equiv V_{free} \cup V_{occ} \cup V_{un}$. The goal of the exploration process is to explore the environment as soon as possible.

Our proposed approach is a sampling-based exploration where the goal is to increase the overall knowledge of the environment by directing the robot in a way that reduces the overall exploration time. An overview of the proposed system is given in Fig. 2. The OctoMap module requires a suitable sensing system, such as a laser scanner or a camera, to create a 3D map. In our case, a LiDAR point cloud is used to

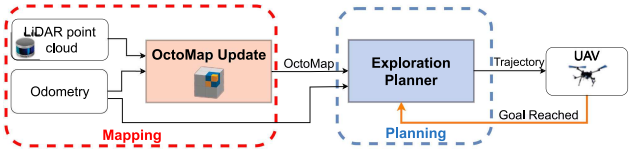


Fig. 2. Overall schematic diagram of the 3D exploration. The LiDAR point cloud and odometry data represent inputs to the OctoMap module. The exploration planner module (highlighted in blue) generates a trajectory to the selected target towards which the robot navigates.

generate an OctoMap, which is used for both exploration and collision-free navigation.

The exploration method is based on a novel information gain computation algorithm that ensures a fast exploration of the environment. We use RSC algorithm to calculate the *information gain* for each RRT edge, evaluating the whole path rather than the subsequent point only, and navigate the robot towards the best path. Our approach leads to an efficient global exploration of the environment.

B. UAV and Sensor Models

In this work, the exploration is performed with a UAV that has no prior knowledge of the environment. Although the concepts are explained with the UAV in mind, the same approach is applicable on other autonomous robots equipped with LiDARs or other sensors that can be used to build an OctoMap.

The UAV is represented with a state vector $\mathbf{x} = \begin{bmatrix}\mathbf{p}^T & \psi\end{bmatrix}^T \in \mathbb{R}^4$ that consists of the position $\mathbf{p} = \begin{bmatrix}x & y & z\end{bmatrix}^T \in \mathbb{R}^3$ and the yaw rotation angle around z axis $\psi \in [-\pi, \pi)$. Furthermore, the algorithm assumes a maximum linear velocity $\mathbf{v}_{max} \in \mathbb{R}^3$ and a maximum angular velocity around z axis $\dot{\psi}_{max}$. For collision checking, it is considered that the UAV is inside a prism centered at $\mathbf{p}$, with adequate length, width and height l, w, h. The algorithm relies on a maximum range of the sensor $R_{max} \in \mathbb{R}$ with horizontal and vertical Field of View (FOV) in range, α_h, $\alpha_v \in (0°, 360°]$, respectively. This allows our algorithm to work with point-cloud-producing sensors with various FOV, such as camera with limited FOV and LiDARs with limited α_v.

C. Overview of the RH-NBVP and Raycasting Algorithms

The RH-NBVP samples nodes from the position of the robot using the RRT algorithm. For each new node, the expected information gain is calculated as the sum of the unknown volume in the sampled camera FOV, exponentially weighted by distance from the node to the current position of the robot. The total gain of a node is the sum of all gains along the RRT path to that node. The growth of the tree is limited with a predetermined number of nodes. When the limit is reached, the node with the highest total information gain defines the best path and the robot executes only the first edge of the path. The described procedure is then repeated. The exploration is considered complete when the best node information gain is below a threshold g_{zero} and the tree reaches the maximum number iterations.

Information gain for each node is calculated using a Raycasting algorithm which traces the path of a series of rays originating from a given node. The density and range of rays define the area to be examined and are specified in advance. When one of these rays hits an obstacle (e.g., a wall), all voxels that the ray previously touched are considered as free voxels. Otherwise, the voxels are considered as free or unknown, depending on the current state of the OctoMap. This results in knowing the number of free and unknown voxels in a predefined area, in each direction from a specific position. Based on this information, a robot can take appropriate actions to move to an unknown area to reduce the total exploration time.

In general, all algorithms that directly cast rays into the map, cast more rays than necessary because they cast a fixed number of rays regardless of the design of the environment [31]. It is shown in [32] that the computation time of Raycasting algorithm increases as the predefined area increases. This is because the number of rays depends on the predefined area and is not affected by the occupied voxels (obstacles). The problem of computational effort required to calculate the information gain becomes even more apparent when using sensors that produce large point clouds, such as LiDARs.

Moreover, in the RH-NBVP, the robot moves to the first node of the best path before performing another planning iteration. This may cause the robot to move back and forth in a small area, changing the best path in each iteration. As the size of the explored area increases, the RRT can easily reach the maximum number of iteration and result in uncovered regions. Moreover, if the distance to the next node is large, the RRT sampling time increases significantly. This limitation is usually noticeable in narrow and large environments.

D. Recursive Shadowcasting Algorithm Overview

Recursive Shadowcasting was first used in computer games to calculate a FOV from a top-down perspective, where the FOV is defined as a set of locations visible from a specific position in a computer game scene [31]. The original RSC, proposed in [6], considers a 2D FOV grid and initially sets all grid cells to not visible. Then the grid is divided into eight octants centred on the FOV source ($\mathbf{S}$) and the cells within each octant are traversed [31]. This traversal occurs within each octant by rows or columns in ascending order of distance from the FOV source. Fig. 3 shows the steps of the RSC on an octant. When a cell is traversed, its visibility state is set to visible. However, when an occupied cell (the black cell) is encountered, an octant is recursively split into smaller regions (Fig. 3 b) and c)), which are bounded by rays cast from the FOV source to the corners of the occupied cell (blue dashed rays). The cell traversals are then continued within each smaller region. As marked in Fig. 3 a) with green arrows, the algorithm first processes rows one through five without encountering any occupied cell. In line six, three occupied cells are encountered, splitting the free region in two and causing the algorithm to call itself recursively. The recursive call then processes the free region on the left (Fig. 3 b)), while the main iteration of the algorithm continues processing

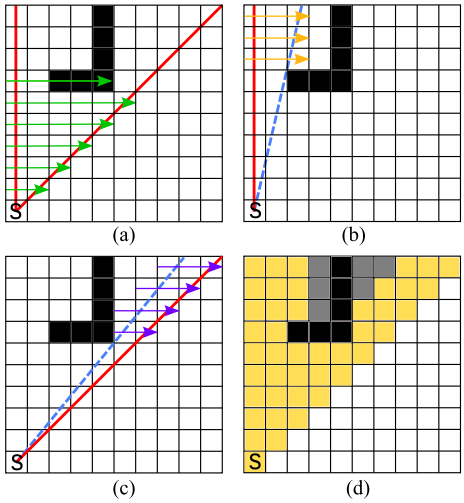


Fig. 3. Steps of the RSC on a single octant.

the free region on the right. Note that even if a ray only grazes the edge of a cell, that entire cell is set to visible.

The result of the RSC on an octant in 2D is shown in Fig. 3 d), where occupied cells are shown in black, visible cells in yellow and invisible cells in grey. Similarly, the main goal of the RSC in the information gain calculation is to find unknown voxels of the OctoMap among the visible cells.

Similar to the FOV grid in 2D computer games, the 3D OctoMap used in this letter is divided into cube-shaped voxels, allowing us to take advantage of the RSC and calculate the information gain in a 2.5D environment. Note that the rays of the RSC in 3D would emanate from a single point, giving us a more accurate estimate of the information gain. Currently, we assume that the 2.5D approximation would lead to a fairly accurate and fast computation of the information gain, so we consider the implementation of the RSC in 3D as future work. To the best of our knowledge, this is the first time that the RSC algorithm is used in the exploration of the environment. In the next section we show how the RSC is used for the evaluation of RRT paths.

E. Cuboid-Based Path Evaluation

The RRT algorithm samples nodes $\mathbf{n} = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$. A collision-free RRT path is denoted as $\mu \in M$, where M denotes the set of all RRT paths. Let $\mu_j, j \in (1, 2, ..., N)$ be the path edge between nodes $\mathbf{n}_{k-1}$ and $\mathbf{n}_k$, where $k \in (1, 2, ..., N + 1)$ and N is the number of nodes. For path edge μ_j, we define information gain $I(\mu_j) \in \mathbb{R}$ as a measure of an unexplored region of the environment that is potentially visible from the center $\mathbf{c}_k$ of this path edge.

To determine the information gain $I(\mu_j)$ using the RSC, we first place a cuboid around the edge μ_j. The cuboid center point is $\mathbf{c}_k$, while the cuboid length is $l_c = \|\mathbf{n}_k - \mathbf{n}_{k-1}\|$. The cuboid width and height are defined by the parameter I_{range}, which depends on the used sensor range and the environment size. The point $\mathbf{c}_k$ is the FOV grid source inside which the RSC is performed, while the cuboid determines the borders of each FOV grid. The FOV grid is obtained as a 2D slice of the OctoMap at point $\mathbf{c}_k$ and the RSC is performed 360° with a horizontal step size r. Additionally, the cuboid is adjusted to the slope of each edge. The voxels considered for computing the information gain should be within the cuboid and the specified boundaries of the environment. An illustration of the cuboid centered at the path edge is shown in Fig. 1. We simplified the illustration showing the performance of the RSC on the first 2D OctoMap slice. As described, the algorithm is performed on each slice and on each path edge.

Note that the maximum cuboid length l_{max} is predefined according to the size of the environment, because calculating the information gain on a large path edge using only one center, that is, one FOV grid for RSC, may result in missing information and poor information gain calculation. In other words, when $l_c > l_{max}$ we add intermediate FOV sources to cover the path edge and to achieve $l_c < l_{max}$.

To form the information gain of the node $\mathbf{n}_k$, edge information gain $I(\mu_j)$ is weighted with the negative exponential of the cost to travel along the path up to $\mathbf{n}_k$, similar to the one proposed in [2] and used in [1]:

$$I(\mathbf{n}_k) = I(\mathbf{n}_{k-1}) + \frac{I(\mu_j)}{e^{\lambda L(\mathbf{n}_k, \mathbf{n}_{k-1})}}, \tag{1}$$

where λ is a positive constant, $L(\mathbf{n}_k, \mathbf{n}_{k-1})$ is Euclidean distance between nodes $\mathbf{n}_k$ and $\mathbf{n}_{k-1}$. The constant λ weighs the importance of the robot motion cost against the expected information gain. A small λ gives the priority to the information gain, while $\lambda \to \infty$ means that the motion is so expensive that the shortest path is selected. The value of λ is experimentally determined.

As for the complexity of algorithms in the information gain calculation, performing a single RC with n horizontal and m vertical rays with the resolution r of the map scales with $\mathcal{O}(\frac{1}{r^4})$ in [1], $\mathcal{O}(nm/r)$ in [7], while our approach with the RSC scales with $\mathcal{O}(\frac{mn \log n}{r})$. By using the proposed algorithm, the high calculation effort required by the RC is avoided. The main reason for the calculation effort reduction is the property of the RSC that ensures each voxel is visited only once.

As presented in [32], [33], RSC has significantly better performance among existing FOV algorithms. However, [31] showed the drawback of RSC algorithm when the grid size increases to tens of thousands of cells. This is because it performs a relatively large number of operations per cell. Nevertheless, we find the RSC suitable for our exploration environments, sensor specifications and OctoMap size.

F. Exploration of the Best Path

According to Eq. 1, path information gain $I(\mu)$ is equal to information gain of the path's last node and presents the volume of the unknown space that is covered along the path, combined with the cost of going there.

In each iteration, our goal is to find the best path μ_{bp}, which maximizes the information gain $I(\mu)$:

$$\mu_{bp} = \arg\max_{\mu \in M} I(\mu). \tag{2}$$

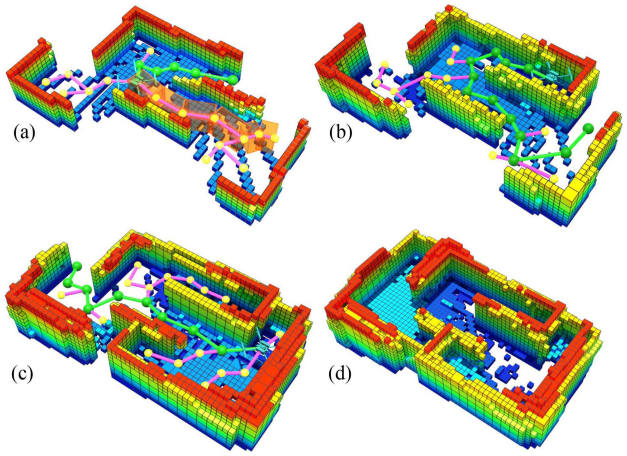


Fig. 4. An illustration of the exploration process. Green path with green waypoints is the best path. Purple paths with yellow waypoints are other paths of the RRT. The UAV body (cyan color) denotes the first point of the path. a) The initial tree with selected path leading towards upper right portion of the environment. Orange cuboids along the right path of the RRT illustrate the volume where the information gain is computed. b) The second iteration with tree leading towards the right of the environment. c) The third iteration leading towards the left portion of the environment. d) Exploration finished and the final map of the environment obtained after third iteration.

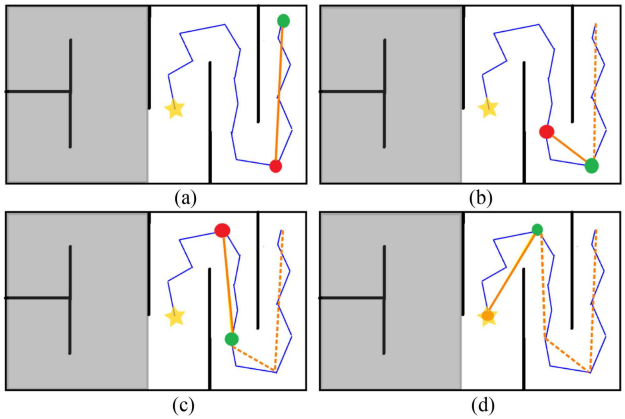


Fig. 5. An illustration of returning from the dead end to the best node inside the proposed recovery strategy. The current node $\mathbf{n}_0$ is marked green, while the yellow star denotes the best node $\mathbf{n}_{bn}$. The blue path is calculated using the RRT and is traversed before reaching the dead end. The orange path connects the nodes of the blue path from $\mathbf{n}_0$ to $\mathbf{n}_{bn}$ to get the shortest possible collision-free path, which is then executed. Grey part of the illustration represents an unknown space.

As soon as the best path μ_{bp} is selected, we address the yaw angle along that path. In the RH-NBVP the yaw angle is randomly sampled during the exploration, which limits the sample efficiency of the exploration. This limitation is briefly addressed in [8], [9], and is not the scope of this letter. Since we use a LiDAR sensor with horizontal FOV $\alpha_h = 360°$, which is attached to the UAV with some pitch angle, our strategy is to align the yaw angle towards the next point on the path.

After the path has been augmented with the yaw angle, it is forwarded to the trajectory planner. Within this letter, we use the Time Optimal Path Parametrization by Reachability Analysis (TOPP-RA) algorithm developed in [34]. Apart from the waypoints, inputs for the TOPP-RA are also velocity and acceleration constraints, which are maximally set to the UAV physical limitations. The planned trajectory is then executed by the UAV, and a new cycle for determining the best path is started after the UAV stops. The exploration process is performed until the entire environment is explored, yielding the environment map. The described process is depicted on Fig. 4.

G. Dead End Resolving Strategy

One of the drawbacks of the RH-NBVP algorithm is the dead end state. It especially occurs in large and narrow environments where the RRT algorithm might end up stuck in a dead end, trying to grow the tree up to the node with non-zero information gain. This results in both higher computation and total exploration time. Inspired by the idea from [9] and [7], we propose a different approach to resolve a dead end state. Both [9] and [7] use RRT planner for local exploration and frontier points for global exploration. In [9], a Breadth First Search (BFS) is used to count reachable frontier points around visited locations, while in [7], frontiers are defined as nodes with a high potential information gain from previously expanded RRTs. However, in [7] the information gain is updated only for the nodes in the previously defined range from the current position of the robot. Our approach mitigates the problems of these approaches (computational requirements of frontier search and search only in a limited range). We say that a dead end state occurs when the information gain of the best path ($I(\mu)$) is smaller than a threshold (I_t) and when the fraction of the known space ($V_{free} \cup V_{occ}$) in the entire space (V) is less than 95 %.

When the robot is in a dead end, the main idea is to return the robot to a previously-visited node that has the highest information gain at the moment, without growing new RRTs, using only previously visited nodes. Information gain is estimated again for each previously visited node by estimating the unknown volume around the nodes using the RSC. Note that the information gain of the path to the node and the corresponding distance (Eq. 1) are not considered. In our observation, the information gain of the previously visited nodes is usually zero or very small. The RSC quickly estimates the best node, $\mathbf{n}_{bn}$, to return to and resolve the dead end.

The previously visited nodes are used to form an obstacle-free path to node $\mathbf{n}_{bn}$. However, instead of backtracking along this path, we use an algorithm to try to find the shortest and collision-free path μ starting from $\mathbf{n}_{bn}$ up to the current node $\mathbf{n}_0$. First, if there is a collision-free path between $\mathbf{n}_{bn}$ and the current node $\mathbf{n}_0$, the path μ is returned. If this is not the case, algorithm tries to find a collision-free path between some other node (denoted as $\mathbf{n}_r$) to the current node $\mathbf{n}_0$. When this is achieved, $\mathbf{n}_r$ becomes a new current node (denoted $\mathbf{n}_{shortest}$). We repeat this procedure until we can connect nodes $\mathbf{n}_{bn}$ and $\mathbf{n}_0$. When the robot returns to the best node, dead-end resolution process is considered finished and the standard exploration process continues. Fig. 5 illustrates the process of dead end resolution by returning to the best node in a simple environment. Note that a simple backtracking along the previous path (denoted with a blue line in Fig. 5) would result in a much longer return path. The main purpose of this dead end

TABLE I
EXPLORATION PARAMETERS

Parameter	Apartment	Maze	Large Maze
r [m]	0.2, 0.4	0.1, 0.2	0.1, 0.2
R_{max} [m]	20.0	20.0	20.0
$\mathbf{v}_{max}^{\{x,y,z\}}$ [m/s]	1.0	1.5	1.5
$\dot{\psi}_{max}$ [rad/s]	0.8	0.8	0.8
I_{range} [m]	5.0	8.0	8.0
λ	0.3	0.6	0.6
I_t	0.1	0.1	0.1

resolving strategy is to shorten the return to the best node and, therefore, avoid unnecessary visits to the previously explored parts of the environment.

IV. SIMULATION ANALYSIS

A. System Setup

Simulations are performed in the Gazebo environment using the Robot Operating System (ROS) and a model of the custom built *Kopterworx* quadcopter. More details about our system and the control structure can be found in our previous work [35]. The quadcopter is equipped with a Velodyne VLP-16 LiDAR sensor featuring a horizontal and vertical FOV $\alpha_h = 360°$, $\alpha_v = 30°$, respectively. For collision checking, dimensions of a prism around the UAV are set to $l = 0.6$ m, $w = 0.6$ m, $h = 0.5$ m. Parameters used in our experiments are shown in Table I. The proposed algorithm is compared with the RH-NBVP and a more recent autonomous exploration planner (AEP) [7]. Parameters $d_{max}^{planner} = 1.5$ m, $\lambda = 0.25$, $N_{max} = 20$ and the maximum RRT tree edge length of 1.5 m refer to the setup of the RH-NBVP explained in [1] and are set to indicated values. The parameters used in the AEP are set to the default values explained in [7], with velocities as given in Table I. We adapted the NBVP and AEP to our quadcopter, equipped with a LiDAR and to our control system to allow the fairest possible comparison. Note that in the AEP the yaw optimization method is neglected, while the horizontal FOV of the sensor is 360°. We run three scenarios with different sizes and resolution r and analyze the results. All simulations have been run 10 times on Intel(R)Core(TM) i7-10750H CPU @ 2.60GHz × 12.

The first scenario refers to a 10 m × 20 m × 3 m relatively simple apartment space used in [1], [7], [17]. The second scenario refers to a 20 m × 20 m × 2.5 m maze environment used in [17], [36]. Finally, the third simulation scenario refers to a 30 m × 30 m × 2 m large maze environment used in [8], [9]. The robot performs a simple trajectory in a close proximity to the initial point, to ensure the planning is performed with some initial information. Additionally, we assume a reliable state estimation and focus on the exploration.

B. Comparison of Raycasting and Shadowcasting

We compared the performance of the RC (used in the RH-NBVP), the sparse RC (used in the AEP) with our RSC-based planner in all three simulation scenarios and at different resolutions. The casting methods in the information gain calculation affect the computation time t_c and thus the total exploration time t_{exp}. The computation time is equal to the time required to select the next best path μ_{bp} in each iteration (growing RRT, path evaluation and μ_{bp} selection). For AEP, t_c includes both the time for gain estimation and collision checking (described in [7]). Computation times and total exploration times for all 10 runs are shown in Table II. It can be observed that the computation times for the RH-NBVP approach are significantly higher than in our approach, especially when using a high resolution map. Furthermore, the use of RC in the RH-NBVP causes the computation time to increase as the complexity of the environment increases. On the other hand, the computation times in the AEP depend mainly on the size of the environment and the number of RRT nodes, rather than on the map resolution (which is especially evident in the large maze scenario). For instance, in the apartment scenario at the resolution $r = 0.2$ m, our planner runs almost thirteen times faster than the AEP, and up to ten times faster for the maze and large maze scenarios. Although the authors in [7] report lower computation times for the apartment scenario than in our simulation runs, we assume that this is caused by using a different sensor FOV and the maximum range of the sensor. The computation time in the proposed planner increases with the finer resolution in all scenarios, but is still low enough not to affect the exploration progress. The results have confirmed that the RC algorithm may cause a bottleneck in larger and more complex scenarios during the exploration. In other words, the robot has to stand still in the air for about 3 s to decide about the next best path. In simulation analysis, we noticed that setting the parameter inside which RC is performed, $d_{max}^{planner}$, to higher values leads higher computation times (up to 10 s).

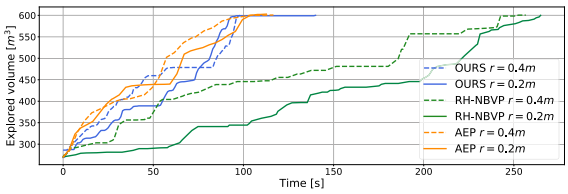


Fig. 6. The explored volume in total exploration time for the apartment scenario.

C. Global Exploration Using Proposed Planner

Simulations were also performed to compare the total exploration time of our exploration planner with the RH-NBVP and the AEP. Fig. 6 shows the explored volume over time for algorithms at a voxel resolution of $r = 0.2$ m and $r = 0.4$ m. It can be observed that our planner and the AEP complete the exploration of the entire area at almost the same time and remarkably faster than the RH-NBVP, especially when using a higher resolution. The graph shows that our method and the AEP need around 100 s to explore the apartment scenario at different map resolutions, while the RH-NBVP needs more than 250 s. Exploring the apartment scenario takes slightly longer in the original setup [7], requiring about 200 s to explore the entire

TABLE II
THE TUPLES OF MEAN AND STANDARD DEVIATION FOR THE TOTAL EXPLORATION TIME t_{exp} AND THE COMPUTATIONAL TIME PER ITERATION t_c

		OURS		RH-NBVP		AEP	
Scenario	**r** [m]	$\mathbf{t_c}$ [ms]	$\mathbf{t_{exp}}$[s]	$\mathbf{t_c}$ [ms]	$\mathbf{t_{exp}}$[s]	$\mathbf{t_c}$ [ms]	$\mathbf{t_{exp}}$[s]
Apartment	0.4	(4.41, 2.39)	(87.82, 13.10)	(15.39, 9.74)	(242.36, 51.63)	(75.96, 99.69)	(92.90, 17.21)
	0.2	(19.63, 10.37)	(113.51, 29.30)	(135.16, 57.86)	(276.84, 70.54)	(248.70, 125.46)	(119.89, 22.69)
Maze	0.2	(25.08, 10.89)	(209.24, 31.03)	(383.33, 124.38)	(504.566, 75.23)	(288.29, 128.48)	(362.01, 17.86)
	0.1	(81.61, 18.84)	(350.05, 87.33)	(1024.19, 297.34)	(832.51, 183.34)	(509.26, 323.56)	(485.79, 90.46)
Large Maze	0.2	(48.67, 19.12)	(1017.23, 271.34)	(744.01, 244.53)	(1847.66, 305.78)	(617.48, 722.61)	-
	0.1	(98.71, 37.52)	(1324.89, 283.22)	(2230.46, 579.43)	(2351.64, 547.52)	(624.08, 583.34)	-

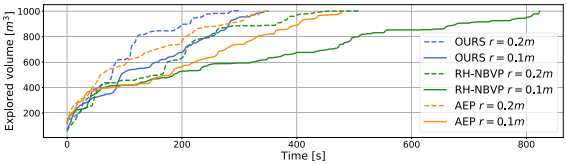


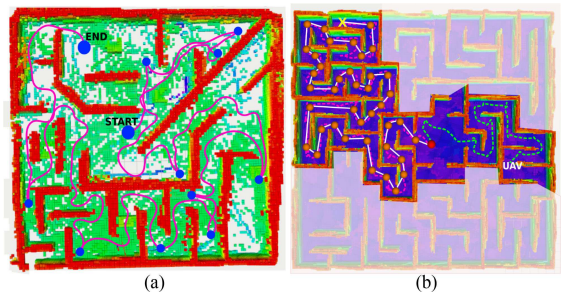
Fig. 7. The explored volume in total exploration time for the maze scenario.

area. However, a method proposed in [17] explores 95 % of the environment in 80 s and 151 s at a resolution of 0.4 m and 0.1 m, respectively.

In the maze scenario, the algorithms were tested using a voxel resolution of $r = 0.1$ m and $r = 0.2$ m. We used higher resolutions because the environment contains some narrow corridors that the UAV cannot navigate through when a coarse resolution is used. The explored volume in time is shown in Fig. 7. Our method explores the entire environment more than twice as fast as the RH-NBVP and about 1.5 times faster than the AEP. All planners behave similarly at the beginning, but over time the RH-NBVP and the AEP show their drawbacks, which affect the total exploration time. Our planner explores the maze environment in 209.24 s with a standard deviation of 31.03 and in 350.05 s with a standard deviation of 87.33 s for a resolution of 0.2 m and 0.1 m, respectively. The total exploration time is comparable to the results of [17], whose method explores 95 % of the environment in 177 s and 330 s for a resolution of 0.2 m and 0.1 m, respectively. The reported results confirm the efficiency of our planner. A thorough comparison of the experimental results with [17] is not possible, due to different equipment and setup used, without the source code provided. The OctoMap of the maze scenario generated by our planner at $r = 0.2$ m is shown in Fig. 8 a) together with the corresponding UAV path and the nodes in which the next best path is calculated. Taking these results into consideration, it is shown that combining the cube-based approximation in the path information gain estimation, instead of considering the nodes only (using either RC or the sparse RC), results in a faster exploration. The random sampling of our, the RH-NBVP and the AEP algorithms leads to revisiting regions, but executing only the first node instead of the whole path results in a higher total exploration time.

D. Evaluation of the Dead End Resolving Strategy

The performance of the dead end resolving strategy is tested on a challenging large maze scenario with dead ends and narrow



(a) (b)

Fig. 8. The OctoMap of maze and large maze scenarios generated by our exploration planner. In a), the path traversed by the UAV in the maze scenario during exploration is marked in pink. The start and end positions of the UAV are highlighted. The nodes in which the next best path μ_{bp} is calculated are marked with blue circles. In b), the pink dashed line shows the path followed during the exploration of the large maze scenario. When a dead end state occurs (yellow marker), the shortest path (solid white line) to the best node (red circle) is executed. The visited nodes that form the shortest path for return are marked with orange circles. The UAV continues to explore unknown (white) space executing green dashed path.

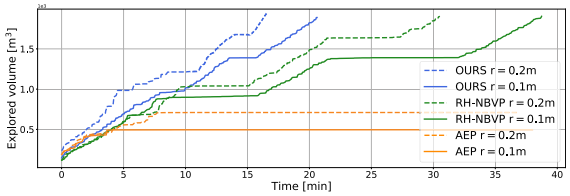


Fig. 9. The explored volume in total exploration time for the large maze scenario.

passages, and further compared to [1] and [7]. The resolution is set to $r = 0.1$ m and $r = 0.2$ m for all planners. The results are shown in Fig. 9. The results demonstrate that our algorithm completes the exploration in 17 minutes and 22 minutes for a resolution of 0.2 m and 0.1 m, respectively. That is more than two times faster on average than the RH-NBVP. The graph shows that the RH-NBVP spends a large amount of time growing the tree when dead end states occur. The authors in [7] use frontiers to deal with dead ends, and a detailed comparison with the proposed method is conducted. Repeated experiments confirm that the AEP (adapted to our control system), at both resolutions, was unable to complete the exploration of such a challenging scenario, despite our best efforts to follow the parameter tuning instructions from [7].

When compared to other state-of-the-art results that use the large maze scenario and history tracking methods, [8] reports

time of 21 minutes at a resolution $r = 0.1$ m and $\mathbf{v}_{max} = 1.0$ m/s, while [9] finished the exploration in 30 minutes ($\mathbf{v}_{max} = 1.2$ m/s). Note that the system setup as well as the maximal exploration velocity are not the same as in our case. The OctoMap of the large maze scenario and an instance of the proposed strategy for resolving dead ends are shown in Fig. 8 b). As can be observed, the path to the best node is successfully shortened, leading to faster exploration.

V. Conclusion and Future Work

This letter presents a novel sampling-based planner for autonomous 3D exploration. The planner is capable of autonomously exploring a previously unknown bounded area and creating an OctoMap of the environment. The results showed an improved behaviour in terms of both computation and total exploration time compared to state-of-the-art strategies. The proposed information gain calculation and path evaluation ensures target evaluation in a short computation time, while a novel dead end recovery algorithm speeds up the exploration process. Even though the proposed information gain calculation method is implemented in 2.5D space, the planner explores the 3D environment and creates a 3D map using a 3D sensor system. This 3D exploration planner has been successfully tested and analysed in simulation scenarios.

For future work we consider testing our planner in an outdoor environment and extending our planner to a hybrid one, combined with the frontier-based approach.

References

[1] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3D exploration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1462–1468.

[2] H. H. González-Baños and J. C. Latombe, "Navigation strategies for exploring indoor environments," *Int. J. Robot. Res.*, vol. 21, no. 10-11, pp. 829–848, 2002.

[3] S. LaValle, "Rapidly-exploring random trees : a new tool for path planning," 1998.

[4] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. ICRA. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia Proc.*, 2000, vol. 2, pp. 995–1001.

[5] T. T. Elvins, "A survey of algorithms for volume visualization," *ACM SIGGRAPH Comput. Graph.*, vol. 26, no. 3, pp. 194–201, 1992.

[6] B. Bergström, "FOV using recursive shadowcasting," Accessed: Sep. 5, 2021. [Online]. Available: http://www.roguebasin.com/index.php?title=FOV_using_recursive_shadowcasting

[7] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt, "Efficient autonomous exploration planning of large-scale 3D environments," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1699–1706, Apr. 2019.

[8] V. Massague Respall, D. Devitt, R. Fedorenko, and A. Klimchik, "Fast sampling-based next-best-view exploration algorithm for a MAV," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 89–95.

[9] C. Witting, M. Fehr, R. Bähnemann, H. Oleynikova, and R. Siegwart, "History-aware autonomous exploration in confined environments using MAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.

[10] A. Batinovic, "A shadowcasting-based next-best-view planner dataset," Accessed: Sep. 5, 2021. [Online]. Available: https://github.com/larics/nbvp_exploration.

[11] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Automat.*, 1997, pp. 146–151.

[12] M. Juliá, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Auton. Robots*, vol. 33, no. 4, pp. 427–444, 2012.

[13] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 2135–2142.

[14] C. Zhu, R. Ding, M. Lin, and Y. Wu, "A 3D frontier-based exploration tool for mavs," in *Proc. IEEE 27th Int. Conf. Tools With Artif. Intell.*, 2015, pp. 348–352.

[15] A. Mannucci, S. Nardi, and L. Pallottino, "Autonomous 3D exploration of large areas: A cooperative frontier-based approach," *Modelling Simul. Auton. Syst.*, vol. 10756, pp. 18–39, 2018.

[16] A. Batinovic, T. Petrovic, A. Ivanovic, F. Petric, and S. Bogdan, "A multi-resolution frontier-based planner for autonomous 3D exploration," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4528–4535, Jul. 2021.

[17] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, "Fast frontier-based information-driven autonomous exploration with an MAV," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 9570–9576.

[18] M. Faria, R. Marín, M. Popović, I. Maza, and A. Viguria, "Efficient lazy theta* path planning over a sparse grid to explore large 3D volumes with a multirotor UAV," *Sensors*, vol. 19, no. 1, p. 174, 2019.

[19] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "FUEL: Fast UAV exploration using incremental frontier structure and hierarchical planning," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 779–786, Apr. 2021.

[20] T. Baiming, S. Jicheng, D. Chaofan, and L. Qingbao, "A target point based MAV 3D exploration method," in *Proc. IEEE Int. Conf. Mechatronics Automat.*, 2018, pp. 2406–2413.

[21] D. Joho, C. Stachniss, P. Pfaff, and W. Burgard, "Autonomous exploration for 3D map learning," in *Autonome Mobile Systeme*. Berlin, Heidelberg: Springer, 2007, pp. 22–28.

[22] D. Deng, Z. Xu, W. Zhao, and K. Shimada, "Frontier-based automatic-differentiable information gain measure for robotic exploration of unknown 3D environments," 2020. [Online]. Available: https://arxiv.org/abs/2011.05288

[23] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1500–1507, Apr. 2020.

[24] Z. Meng *et al.*, "A two-stage optimized next-view planning framework for 3-D unknown environment exploration, and structural reconstruction," *IEEE Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1680–1687, Jul. 2017.

[25] T. Dang, C. Papachristos, and K. Alexis, "Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 2526–2533.

[26] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty-aware receding horizon exploration and mapping using aerial robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 4568–4575.

[27] M. Popović, G. Hitz, J. Nieto, I. Sa, R. Siegwart, and E. Galceran, "Online informative path planning for active classification using uavs," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 5753–5758.

[28] K. Ota, Y. Sasaki, D. K. Jha, Y. Yoshiyasu, and A. Kanezaki, "Efficient exploration in constrained environments with goal-oriented reference path," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 6061–6068.

[29] R. Reinhart, T. Dang, E. Hand, C. Papachristos, and K. Alexis, "Learning-based path planning for autonomous exploration of subterranean environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 1215–1221.

[30] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[31] E. Debenham and R. Solis-Oba, "Efficient field of vision algorithms for large 2D grids," *Int. J. Comput. Sci. Inf. Technol.*, vol. 13, no. 1, pp. 1–20, 2021.

[32] E. Debenham, "New algorithms for computing field of vision over 2D grids," Electronic Thesis and Dissertation Repository. 6552. Univ. Western Ontario, 2019.

[33] Jice, "Comparative study of field of view algorithms for 2D grid based worlds," 2009. Accessed: Sep. 5, 2021. [Online]. Available: http://www.roguebasin.com/index.php?title=Comparative_study_of_field_of_view_algorithms_for_2D_grid_based_worlds

[34] H. Pham and Q. Pham, "A new approach to time-optimal path parameterization based on reachability analysis," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 645–659, Jun. 2018.

[35] R. Milijas, L. Markovic, A. Ivanovic, F. Petric, and S. Bogdan, "A comparison of lidar-based slam systems for control of unmanned aerial vehicles," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2021, pp. 1148–1154.

[36] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D euclidean signed distance fields for on-board MAV planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1366–1373.

Publication 3: ASEP: An Autonomous Semantic Exploration Planner with Object Labeling

Milas, A., Ivanovic, A., and Petrovic, T. (2023). ASEP: An Autonomous Semantic Exploration Planner With Object Labeling. IEEE Access, 11, 107169-107183.

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 August 2023, accepted 24 September 2023, date of publication 28 September 2023, date of current version 4 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3320645

RESEARCH ARTICLE

ASEP: An Autonomous Semantic Exploration Planner With Object Labeling

ANA MILAS, (Member, IEEE), ANTUN IVANOVIC, (Student Member, IEEE), AND TAMARA PETROVIC, (Member, IEEE)
Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia

Corresponding author: Ana Milas (ana.milas@fer.hr)

This work was supported in part by Project VIRTUALUAV—Development of a System of Unmanned Aerial Vehicles (UAVs) Controlled in Virtual Environments funded by the European Regional Development Fund under Grant KK.01.2.1.02.0197, and in part by the Scientific Project Strengthening Research and Innovation Excellence in Autonomous Aerial Systems—AeroSTREAM [1] supported by European Commission HORIZON-WIDERA-2021-ACCESS-05 Program under Grant 101071270. The work of Ana Milas was supported in part by the Young Researchers Career Development Project–Training of Doctoral Students of the Croatian Science Foundation funded by the European Union from the European Social Fund.

ABSTRACT In this paper, we present a novel 3D autonomous exploration planner called the Autonomous Semantic Exploration Planner (ASEP), designed for GPS-denied indoor environments. ASEP combines real-time mapping, exploration, navigation, object detection, and object labeling onboard an Unmanned Aerial Vehicle (UAV) with limited resources. The planner is based on a frontier exploration strategy that utilizes semantic information about the environment in the exploration policy. The policy is extended to incorporate both geometric and semantic information provided by a deep convolutional neural network (DCNN) for semantic segmentation. This semantically-enhanced exploration algorithm directs the exploration toward the quick labeling of all objects of interest in the environment. An extended path planning algorithm continuously checks for path validity, enabling safe navigation in challenging environments. The overall system is designed to be modular and easily extended or replaced with custom modules. The proposed planner is evaluated and analyzed in both simulation and real-world environments using a UAV. Experimental studies demonstrate the effectiveness of the ASEP strategy compared to state-of-the-art methods. Results show that the objects in the environment are explored faster and total exploration time is reduced while the computational time remains consistent regardless of the semantic segmentation processing involved.

INDEX TERMS Autonomous exploration, semantic segmentation, UAV, path planning, object labeling.

I. INTRODUCTION

Autonomus exploration using UAVs has gained significant attention in recent years due to its numerous advantages, including operations in inaccessible or hazardous environments, increased efficiency and cost-effectiveness, mapping and monitoring capabilities, disaster response and search-and-rescue operations. When autonomously exploring indoor environments using a UAV with limited resources, where GPS signals may not be available, it can represent a significant challenge. Industries such as warehousing, logistics, inspection, or maintenance can benefit from the advancements in autonomous UAV exploration to optimize operations, reduce costs, and improve safety.

Using the semantic data from the environment in autonomous exploration is a marginally researched area so far. Generally, it includes simultaneous object detection using semantic segmentation and autonomous exploration. The result is a map of a previously unknown area with labeled objects of interest. Within this work, our objective is to determine the position and semantic label of objects of interest (hereafter referred to only as label objects of interest)

The associate editor coordinating the review of this manuscript and approving it for publication was Heng Wang.

in warehouse environments during exploration. These objects of interest may include shelves, boxes, and doors, among others, while the set of objects of interest is determined prior to exploring the environment. By using the information about the objects in the environment, our exploration system can effectively identify key regions with a high concentration of relevant objects, resulting in faster labeling of objects of interest, and at the same time building a map of the previously unknown environment. The obtained map with labeled objects can be used then for navigation, detailed visual inspections of equipment and infrastructure, counting boxes in warehouses, etc. Exploration algorithms aim to explore completely or partially unknown environments, usually as fast as possible, considering the data from the environment. State-of-the-art methods are focused on extracting data from metric maps, such as the widely used OctoMap [2] and including this data into the exploration policy. The exploration methods are mostly divided into frontier-based [3] and sampling-based [4]. In the literature, most criteria considered by exploration strategies refer only to metric information, i.e., information that can be derived from metric maps that a robot builds. Recently, a few exploration strategies proposed using semantic features to evaluate candidate locations and select the next best goal [5], [6], [7], which will be explained in detail in the following section. Semantic features from the environment can be mapped into the OctoMap so that each voxel is assigned additional information describing its semantic label. This approach is introduced in [8] and used in [5]. On the contrary, to avoid altering the OctoMap structure and make the approach applicable to other map representations, semantic features can be labeled in the 3D environment to have the position and semantic label, as described in this paper. Presented in this way, semantic features can be easily used in the exploration strategy to speed up object labeling. To the best of authors' knowledge, there is no semantically-enhanced exploration algorithm that directs the exploration to quickly label all objects of interest on the map.

Thus, we present an autonomous UAV exploration planner called Autonomous Semantic Exploration Planner (ASEP), that enables real-time mapping, exploration, navigation, object detection and labeling in GPS-denied indoor environments. By leveraging onboard sensors and processing capabilities, a UAV can detect and identify objects of interest, such as equipment, products, or inventory, in real time, providing valuable information for the exploration algorithm. We provide the integration of real-time localization, mapping and semantic segmentation onboard a UAV equipped with an RGB-D camera. The semantically segmented object from the image frame is projected on a 3D map of the environment. The proposed approach takes advantage of the semantic information extracted in 3D so that a new utility function is introduced to guide the UAV toward the objects in the environment. The proposed planner is thoroughly evaluated in both simulation and real-world environments and compared with state-of-the-art methods. The results demonstrate the proposed strategy is capable of exploring unknown environments and labeling objects effectively.

A. CONTRIBUTIONS

The key contributions of this work are summarized as follows:

- An exploration strategy that utilizes the previously introduced frontier-based method [9] to generate candidates, and combine semantic utility functions to iteratively explore the unknown environment and label all the objects of interest. This includes an extension of the information gain function to incorporate not only geometric but also semantic information of the environment.
- A 3D object labeling during exploration. This includes extraction of semantically segmented objects from 2D images and processing camera depth point cloud to estimate the position of the object in the environment.
- An overall system for mapping, exploration, path planning and navigation that is modular and can be extended or replaced by a custom module. The system utilizes a low-cost sensing system and ensures exploration and object labeling onboard a UAV with limited resources.

B. ORGANIZATION

The paper is organized so that in Section II we give an overview of the state-of-the-art of 3D exploration methods and position our work in relation to them. Section III describes problems solved within this paper while Section IV introduces system and sensor models used in the proposed method. Section V is the core of the paper and contains details of the proposed planner. Results of simulations performed with a UAV and their analysis are presented in Section VI while Section VII shows our experiment setup and results in a real-world indoor environment. The paper is concluded in Section VIII.

II. RELATED WORK

In the context of warehousing, UAVs are used to collect data and provide certain information for tasks such as inventory management, monitoring stockpiles, and inspecting hard-to-reach areas within a warehouse facility. UAVs used in warehouse exploration missions should be able to effectively navigate the environment and gather information. To achieve this, UAVs should be able to localize themselves, detect objects, explore and map the environment. This paper mainly focuses on environment exploration and object labeling on the map. Thus, in this section, state-of-the-art methods for autonomous exploration are overviewed.

There is a wealth of earlier work related to autonomous exploration, especially for 2D, but more recently also for 3D environments. Typical exploration approaches can be roughly classified into frontier-based, sampling-based, and hybrid strategies, even though there is a significant overlap between these categories.

A characteristic of frontier-based approaches is exploration by approaching a selected point on the frontier between the explored and unexplored portion of the environment. This idea was first introduced by Yamauchi in [3] and subsequently evaluated in more detail in [10]. In each iteration, the next best goal is a frontier point closest to the robot. The simplest approach to 3D exploration is to use 2D frontier-based exploration with 3D maps at different heights (oftentimes called 2.5D approaches) [11]. A complete frontier-based solution for 3D environments is developed in [12], where the next best goal is the frontier that minimizes the velocity change to maintain a consistently high flight speed. It is shown that this approach outperforms the closest frontier method [3]. Frontier-based exploration approaches for 3D environments are also researched in [9], [13], [14], [15], [16], [17], and [18].

Sampling-based approaches aim to determine a (minimal) sequence of robot (sensor) viewpoints to visit in the environment, until the entire space is explored. Potential viewpoints are typically sampled near the frontier or randomly. Then these viewpoints are evaluated for the potential information gain and the next best viewpoint is assigned. One of the first sampling-based methods is presented in [19] and then extended in [4], [20], and [21]. In [4], authors proposed the Receding Horizon Next-Best-View planning (RH-NBVP), which uses a Rapidly-exploring Random Tree (RRT) [22], [23] to guide a UAV into the unexplored area. While the method showed good scaling properties and performance in a local exploration, it is not resilient to dead ends, resulting in poor global scene coverage and, thus, a high overall exploration time, as shown in [12], [15], and [24], and in our previous work [9]. Improvements of the RH-NBVP are presented in [25], [26], [27].

Hybrid strategies combine the advantages of both frontier-based and sampling-based approaches. Selin et al. [28] successfully combine the RH-NBVP with conventional frontier reasoning to compensate for a poor performance in global exploration. In other words, [28] plans global paths towards frontiers and samples paths locally. Meng et al. [29] samples viewpoints around frontiers and finds the global shortest tour passing through them. Similarly, Respall et al. [30] samples viewpoints in the vicinity of a point of interest near a frontier, and additionally memorizes nodes that indicate regions of interest in a history graph to reduce the gain calculation time.

Apart from classification related to candidate extraction and evaluation, exploration algorithms differ in the map used for exploration policy. Besides the volumetric map, such as the OctoMap [2], the environment can also be represented by a topological map with semantic features [31], which can improve the efficiency of the robotic exploration by facilitating the next best goal selection. The nodes on the graph that contain the semantic features are used to guide the exploration. Gomez et al. [32] presented a hybrid mapping approach that combined topological mapping with 3D dense mapping for large indoor 3D environments.

Recently, more and more exploration systems use semantic features from volumetric maps to evaluate candidate locations and select the next best goal. The authors in [6] extend the sampling-based approach from [4] to include the semantic segmentation information in a harbor-like environment. Similarly, Ashour et al. [5] presents an exploration strategy for UAVs that integrates environmental semantics for the object mapping. The approach combines semantic information with autonomous exploration techniques to guide the exploration path and enhance object mapping efficiency using the approach from [8]. Instead of mapping objects during the exploration, objects can be extracted from 2D images and then converted to 3D point types using the point cloud library (PCL). Previously, Wang et al. [33] introduced the extraction of edges. Furthermore, most of the semantic-aware exploration strategies are goal-oriented (search for an object), such as [7], [34], and [35]. Authors in [7] introduced a frontier semantic exploration method for visual target navigation. Both frontier detection and semantic segmentation are performed using neural networks.

Regarding the navigation and operations in the warehouse environment using UAVs, Campos-Macias et al. [36] presented an autonomous navigation framework for capturing inventory and locating out-of-place items while focusing on the exploration in unknown 3D cluttered environments. They used an RGB-D camera for depth sensing and a tracking camera for the visual-inertial odometry. Kwon et al. [37] demonstrated autonomous navigation for inventory inspection tasks in long and narrow warehouse aisles using a low-cost sensing system. Their system consists of a relatively small number of sensors, including three cameras, a laser scanner and a range sensor.

Even though efforts to improve the efficiency, accuracy, and robustness of autonomous exploration have shown promising results, it is important to note that a semantically-enhanced exploration algorithm for onboard UAV applications, which focuses on fast object labeling, has not yet been developed in the literature. With this in mind, we present a novel autonomous exploration strategy specifically designed for UAVs with limited payload capabilities and computational resources. Our approach integrates real-time mapping, exploration, navigation, object detection, and object labeling capabilities directly onboard the UAV. The key component of our proposed strategy is a frontier exploration planner that incorporates semantic information about the environment into the exploration policy. By leveraging this semantic understanding, our planner enables the UAV to make informed decisions regarding which frontiers to explore and, thus, directs the exploration toward the quick labeling of all objects of interest on the map. By combining real-time mapping, exploration, navigation, object detection, and object labeling, our approach addresses the limitations of

existing algorithms and provides a comprehensive solution for autonomous exploration onboard UAVs.

III. PROBLEM DESCRIPTION

The main goal of the proposed approach is to explore a bounded and previously unknown 3D space $V \subset \mathbb{R}^3$ and to label objects of interest in a 3D map as soon as possible. As a basis for our approach, an OctoMap M is used, a hierarchical volumetric 3D representation of the environment [2]. Each cube of the OctoMap is denoted as a voxel (cell), which can be *free*, *occupied* or *unknown*. Free voxels form the free space $V_{free} \subset V$, occupied voxels form the occupied space $V_{occ} \subset V$ and unknown voxels form the unknown space $V_{un} \subset V$. Initially, the entire bounded space is unknown, $V \equiv V_{un}$, and the unknown space decreases as the exploration advances. The entire space is a union of the three subspaces $V \equiv V_{free} \cup V_{occ} \cup V_{un}$. The exploration problem is considered fully solved when $V_{occ} \cup V_{free} \equiv V \setminus V_{res}$, where V_{res} is residual space defined as an unexplored space, which remains inaccessible to the sensors. Namely, sensors have limitations in perceiving surfaces, leading to an inability to explore hollow spaces or narrow pockets within a given setup.

The object labeling in a map M is executed in parallel with the exploration. Let O be the set of objects of interest present in the map. The set O is defined as:

$$O = \{o_i | i = 1, 2, \ldots, N_{obj}\}, \tag{1}$$

where N_{obj} is the total number of objects in the 3D map. Each object o_i is represented by its 3D position and its semantic label s_i, selected from the set of semantic labels:

$$S = \{s_i | i = 1, 2, \ldots, N_{labels}\}, \tag{2}$$

where N_{labels} is the total number of different semantic labels. It is preset and depends on the elements expected in the environment. In this work, the focus is on static objects and the semantic labels are related to a warehouse scenario. Each object o_i is defined as $o_i = (\mathbf{p}_{obj_i}, s_i)$, where $\mathbf{p}_{obj_i} = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \in \mathbb{R}^3$ is the object position. The semantic labels of objects are obtained by the semantic segmentation algorithm described in IV-C. Given the nature of this problem, it is crucial to employ an algorithm capable of detecting objects and estimating their positions in real-time, while exploring and mapping the unknown envronment. Additionally, a suitable and obstacle-free path should be computed online. The autonomy of the algorithm requires the planner to run onboard with limited computational resources.

IV. SYSTEM OVERVIEW

In the subsequent section, a detailed description of the proposed system is provided, including the sensors used and the methodologies employed for the simultaneous localization and mapping (SLAM), as well as semantic segmentation based on 2D images and object pose estimation from RGB-D data.

An overview of the proposed system is given in Fig. 1. It shows the proposed semantically-aware exploration system architecture, which consists of five main modules: 1) system input, 2) localization and mapping, 3) semantic data extraction, 4) exploration, and 5) path planning and navigation. A detailed description of all modules is given in the following sections.

A. UAV AND SENSOR MODELS

In this work, the exploration is performed with a UAV that has no prior knowledge of the environment. Although the concepts are explained with the UAV in mind, the same approach is applicable to other autonomous robots equipped with a camera or other sensors that can be used to utilize SLAM and build an OctoMap.

The UAV is represented with a state vector $\mathbf{x} = \begin{bmatrix} \mathbf{p}^T & \psi \end{bmatrix}^T \in \mathbb{R}^4$ that consists of the position $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$ and the yaw rotation angle around the body z axis $\psi \in [-\pi, \pi)$. Furthermore, the algorithm requires dynamical constraints in terms of velocity $\dot{\mathbf{x}}_{max} \in \mathbb{R}^4$ and acceleration $\ddot{\mathbf{x}}_{max} \in \mathbb{R}^4$ for each degree of freedom. For collision checking, it is considered that the UAV is inside a rectangular prism centered at $\mathbf{p}$, with adequate length, width and height l, w, h.

The algorithm relies on a maximum range of the sensor $R_{max} \in \mathbb{R}$ with horizontal and vertical Field of View (FOV) in range, α_h, $\alpha_v \in (0°, 360°]$, respectively. This allows our algorithm to work with point-cloud-producing sensors with various FOV, such as cameras with limited FOV, and LiDARs with limited α_v. In this paper, an RGB-D camera that provides rich visual and depth information is used, allowing the UAV to build a detailed and accurate map of the environment and localize itself using the visual SLAM method.

B. LOCALIZATION AND MAPPING

The sensing system described above allows the robot to estimate its pose and capture point clouds of its environment. However, this sensed data is not sufficient by itself to create a consistent global 3D map. As the robot moves around the world, measurements from these sensors must be integrated, taking into account the motion of the robot, to create a coherent global representation of its surroundings, i.e., a map. In many cases, the pose of the robot in the map will be estimated at the same time as the map is built, which is often known as localization; in this case, the task is usually referred to as SLAM.

During autonomous exploration, robots need to navigate in unknown or partially known environments, gradually perceiving the environment through streaming data provided by onboard sensors. A majority of the 3D strategies use a metric map, an OctoMap [2], in order to navigate through 3D space and visualize the environment. The OctoMap is a hierarchical volumetric 3D representation of the environment. The OctoMap can be generated using the input from the SLAM algorithm [39], as shown in [9], or with raw data from a sensor system, such as a laser scanner or a camera, as demonstrated in [27]. In the proposed system, a camera point cloud is used to generate an OctoMap, as shown

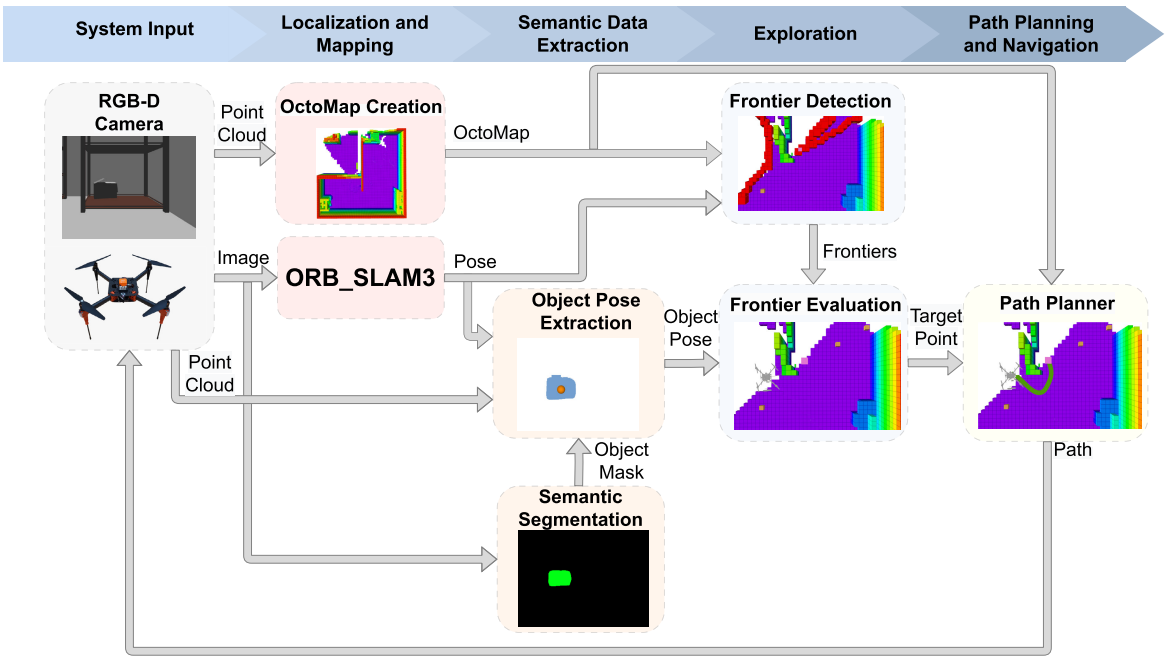


FIGURE 1. Overall schematic diagram of the 3D exploration. The system input module consists of a forward-facing camera that produces input data for both SLAM and OctoMap creation. The semantic segmentation module and object pose extraction module ensure object recognition and object pose estimation in a 3D map, respectively. The semantically-enhanced exploration method generates a target point towards which the robot plans its path and navigates. The navigation system relies on an off-the-shelf SLAM system with loop closing and relocalization capabilities [38], which is fed with RGB-D images.

in Fig. 1. Localization and mapping module results in an OctoMap of the environment and in the pose provided by a SLAM algorithm. An OctoMap is used for both exploration (frontier detection) and path planning and navigation module (collision-free navigation).

For visual SLAM ORB-SLAM3 [38] is utilized, which performs visual, visual-inertial and multimap SLAM with different camera types (monocular, stereo and RGB-D). Within this paper, RGB-D camera is used to perform ORB-SLAM3. It uses depth information to synthesize a stereo coordinate for extracted features on the image. This way SLAM system is agnostic of the input being stereo or RGB-D. As shown in [40] and [41], the RGB-D SLAM outperforms state-of-the-art methods in most sequences on TUM RGB-D dataset.

To avoid possible sudden jumps resulting from the ORB-SLAM3 algorithm loop closures, a multi-sensor fusion method introduced in [42] is used. In this case, the multi-sensor fusion method takes into account the inertial measurement unit (IMU) data obtained from the LPMS-CU2 unit, together with the pose measurements from the ORB-SLAM3.

C. 2D-IMAGE-BASED SEMANTIC SEGMENTATION

An input image from an RGB-D camera can be represented as a 2D array of pixel values. Semantic segmentation is a computer vision technique that involves labeling each pixel of the image with a specific class or category. The objective of the semantic segmentation is to predict the segmentation map for the input image, but instead of containing pixel values, it contains the predicted semantic labels for each pixel. To each object of interest, o_i corresponds a collection of pixels that form a distinct entity that can be visually identified and distinguished from the background or other elements in the image. Then, using a semantic segmentation algorithm, the semantic labels s_i for each object o_i can be determined. In other words, the goal is to find a function $f : O \rightarrow S$ that maps each object to its corresponding semantic label. Given a number of semantic labels N_{labels}, the determination of semantic labels for each object can be expressed as:

$$f(o_i) = s_j \quad \text{for } 1 \leq i \leq N_{obj}, 1 \leq j \leq N_{labels}, \tag{3}$$

where o_i is the object, and s_j is a semantic label, from the set S and for the given object.

By utilizing deep learning models, this approach can accurately segment objects and regions of interest in 2D images. In our work, for semantic segmentation, the HRNet [43] is used, which is a recently proposed model that retains high-resolution representations throughout the model, without the traditional bottleneck design. The model architecture is shown in Fig. 2.

The HRNet model for the 2D image semantic segmentation is used since it showed enviable performance results [45]. Furthermore, it is compact, fast, robust and easy to use, enabling the model adaptation to work on CPU only, making it suitable for applications running on UAVs with

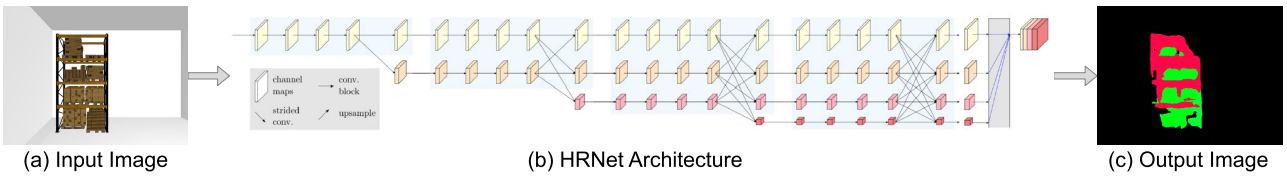


FIGURE 2. HRNet architecture applied to the task of semantic segmentation. The input image from the Gazebo simulator contains a shelf and boxes. The corresponding output image shows that the HRNet has successfully identified and segmented the objects. ©2021 IEEE. Reprinted with permission from [44].

limited computational resources. The model is trained on the ADE20K dataset with 150 objects and stuff classes included. ADE20K is the largest open-source dataset for semantic segmentation and scene parsing, released by the MIT Computer Vision team [46], [47]. The ADE20K dataset is selected since some datasets have a limited number of objects (e.g., COCO [48], Pascal [49]) and in many cases, those objects are not the most common objects one encounters in indoor environments, or the datasets only cover a limited set of scenes (e.g., Cityscapes [50]). Additionally, objects of interest are extracted from the semantically segmented image. In the case of warehouse exploration, the objects of interest are shelves, boxes, doors, etc. Fig. 2 illustrates an example of semantic segmentation performed on an image with a shelf and multiple boxes. As shown in Fig. 2, it is not important that each individual item (e.g., a box) from a set of items (e.g., boxes) is detected, but at least one that is then used in the exploration policy.

D. OBJECT POSITION EXTRACTION

Once objects are semantically segmented in a 2D image, the next step is to determine the 3D positions of those objects in the world to be used for 3D exploration. This process involves utilizing both an object mask and a camera point cloud (Fig. 1). For each object o_i with semantic label s_i from the set S, the 3D position of the object $\mathbf{p}_{obj_i} = \begin{bmatrix} x_{oi} & y_{oi} & z_{oi} \end{bmatrix}^T$ needs to be found. The image mask serves as a binary representation of the segmented objects in the image, where each pixel belonging to an object is assigned a value of 1, while pixels outside the objects are assigned a value of 0. This mask essentially acts as a filter, isolating the regions of interest from the background and other irrelevant elements in the image. Masks are then extracted for each object detected in the image. Alongside the image mask, a camera point cloud C is utilized. In general, a point cloud C is defined as:

$$C = \{\mathbf{c}_i | i = 1, 2, \ldots, N\}, \tag{4}$$

where $\mathbf{c}_i \in \mathbb{R}^3$ while N represents the number of points. In other words, a point cloud is a collection of 3D points that represent the surface geometry of objects in the scene.

To perform the object position extraction, the algorithm utilizes both the image mask and the camera point cloud. It associates the segmented objects in the image mask with their corresponding points in the point cloud. If H_{o_i} is the 2D image mask corresponding to the object o_i, then $C_{o_i} \in C$ is the set of 3D points from RGB-D camera that is aligned with the mask. It means that for the image mask of each object, there is a corresponding point cloud. The 3D position $\mathbf{p}_{obj_i}$ of the object o_i is then determined using point cloud C_{o_i}. Namely, the centroid technique is used to estimate the position from the points in C_{o_i}, expressed in the camera coordinate frame:

$$\mathbf{p}^{\mathrm{c}}_{obj_i} = \frac{\sum_{i=1}^{N_{pixels}} \mathbf{c}_{o_i}}{N_{pixels}}, \tag{5}$$

where N_{pixels} represents the total number of points aligned with the image mask of the object. Note that the number of points from the point cloud aligned with the mask is equal to the number of pixels from the image mask. By aligning the 3D points with the 2D image coordinates, the algorithm determines the position of each object in the coordinate system of the camera. Given the UAV state in the world frame, the position of the object in a global 3D map can be determined:

$$\mathbf{p}^{\mathrm{w}}_{obj_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \mathbf{T}^{\mathrm{b}}_{\mathrm{w}} \cdot \mathbf{T}^{\mathrm{c}}_{\mathrm{b}} \cdot \begin{bmatrix} \mathbf{p}^{\mathrm{c}}_{obj_i} & 1 \end{bmatrix}^T, \tag{6}$$

where $\mathbf{T}^{\mathrm{b}}_{\mathrm{w}} \in \mathbb{R}^{4\times4}$ is the homogeneous transform matrix defining the position and orientation of the UAV in the world coordinate frame, and $\mathbf{T}^{\mathrm{c}}_{\mathrm{b}} \in \mathbb{R}^{4\times4}$ defines the fixed transformation between the UAV body and camera. Note that the first matrix in the equation transforms the four-dimensional vector into a three-dimensional position vector. This combined approach effectively maps the 2D image objects to their corresponding 3D locations and allows for accurate and robust extraction of object positions, as shown in Section VI.

V. SEMANTIC-BASED EXPLORATION

In this section, the proposed semantic-based exploration method is described in detail. Our previously developed frontier-based method [9] is used to extract frontier voxels (frontiers) from the OctoMap. Those frontiers are candidates for the next waypoints of exploration. Each candidate is evaluated using a semantically-aware policy and, finally, the best candidate is selected as the next waypoint to which the UAV plans a path and navigates. The main contribution in

this part is the extension of the information gain function to include semantic data of the environment.

A. FRONTIER DETECTION

A frontier, F, can be defined as a set of voxels $\mathbf{v}_f$ with the following property [9]:

$$F = \{\mathbf{v}_f \in V_{free} : \exists neighbor(\mathbf{v}_f) \in V_{un}\}. \quad (7)$$

In other words, a frontier consists of free voxels with at least one unknown neighbor. The center of a frontier voxel is often called the frontier point. Since the space V is bounded, once the frontier set becomes empty, $F = \emptyset$, the exploration is considered done.

The OctoMap used for frontier detection is generated using camera point clouds C. The OctoMap is in the form of OcTrees, a format suitable for path planning. During the exploration, the OctoMap M is built iteratively using the method described in [2]. Within this work, the current OctoMap M^i is created from the current point cloud C^i added to the OctoMap explored so far:

$$M^i = f(M^{i-1}, C^i), M^0 = \emptyset. \quad (8)$$

With each new-coming point cloud, a new OctoMap is created according to Eq. 8. The OctoMap is updated as each point cloud is processed. At the same time, a frontier detection cycle is performed periodically to ensure that frontiers are constantly updated. Please note that the rate of an OctoMap update process is lower than the frontier detection process since the OctoMap update is a computationally demanding process, especially when using dense point clouds.

Let V^i_{free} and V^{i-1}_{free} correspond to the free voxels in two consecutive OctoMaps, M^i and M^{i-1}. The local frontier F_l, which contains only newly created frontier points can be calculated as follows [9]:

$$F_l = \{\mathbf{v}_f \in V^i_{free} \setminus V^{i-1}_{free} : \exists neighbor(\mathbf{v}_f) \in V^i_{un}\}. \quad (9)$$

The global frontier F_g is a union of all past local frontiers, updated in each iteration and filtered to exclude voxels that do not satisfy the property Eq. 7 anymore. F_g is calculated as follows:

$$\begin{aligned} F^i_g &= F^i_l \cup F^i_{gf} \\ F^i_{gf} &= \{\mathbf{v}_f \in F^{i-1}_g : \exists neighbor(\mathbf{v}_f) \in V^i_{un}\}, F^0_g = \emptyset. \end{aligned} \quad (10)$$

There is usually a large number of voxels in the global frontier (referred to only as frontier from now on) and their evaluation is expensive in view of the computing effort involved. In our previous work, we cluster the global frontier voxels F^i_g, as explained in [9], to get frontier voxels which are candidates, denoted as F_c, for becoming a next waypoint for the exploration. As stated in [9], the frontier is clustered using multi-resolution clustering and mean shift clustering algorithms. In the proposed approach, candidates F_c are frontier F_g, clustered using the mean shift clustering algorithm.

B. SEMANTICALLY-AWARE FRONTIER EVALUATION

The main goals of this approach are to explore the environment and label the objects of interest on the map. Labeled objects of interest are included in the exploration policy, assuming that this leads to faster object labeling. To evaluate each voxel in F_c, the *total gain* of every candidate $\mathbf{v}_c \in F_c$ is defined using the following function:

$$G(\mathbf{v}_c) = \alpha I_{gg}(\mathbf{v}_c) + \beta I_{sg}(\mathbf{v}_c), \quad (11)$$

where α and β are positive constants, while $I_{gg}(\mathbf{v}_c)$ and $I_{sg}(\mathbf{v}_c)$ represent *geometric information gain* and *semantic information gain* of each candidate $\mathbf{v}_c$, respectively. Therefore, α and β represent the trade-off between the geometric and semantic information gain. The values of α and β are experimentally determined and depend on the environment layout.

The geometric information gain $I_{gg}(\mathbf{v}_c)$ is defined using the function similar to the one proposed in [19]:

$$I_{gg}(\mathbf{v}_c) = \frac{I_{un}(\mathbf{v}_c)}{e^{\lambda L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}_c})}}, \quad (12)$$

where λ is positive constant, $L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}_c})$ is the distance between the robot's current position $\mathbf{p}_i$ and the position of the candidate $\mathbf{p}_{\mathbf{v}_c}$, while $I_{un}(\mathbf{v}_c)$ is a *information gain* i.e. a measure of the unexplored region of the environment that is potentially visible from $\mathbf{v}_c$. A high information gain indicates that a specific $\mathbf{v}_c$ provides significant information about the environment, while a low information gain suggests that the $\mathbf{v}_c$ contributes less to reducing unknown space. The information gain $I_{un}(\mathbf{v}_c)$ is defined as the share of unknown voxels in a cube placed around $\mathbf{v}_c$, as described in [9]. The cuboid width and height are defined by the parameter I_{range}, which depends on the used sensor range and the environment size. Often, the information gain is estimated using a ray tracing algorithm and a real sensor field of view instead of using a cube-based approximation. By using the proposed simplification, the high calculation effort required by ray tracing is avoided. The estimated distance is approximated using the Euclidean distance between the robot position $\mathbf{p}_i$ and the position of the candidate (voxel center) $\mathbf{p}_{\mathbf{v}_c}$, $L(\mathbf{p}_i, \mathbf{p}_{\mathbf{v}_c}) = \|\mathbf{p}_i - \mathbf{p}_{\mathbf{v_c}}\|$. The constant λ weights the importance of robot motion cost against the expected information gain. A small λ gives priority to the information gain, while $\lambda \to \infty$ means that the motion is so expensive that only $\mathbf{v}_c$ near the robot is selected. As described in [9], λ is set to satisfy the ratio between the desired information gain and the distance with respect to the desired behavior of the system. To include semantically segmented objects from the environment in the exploration policy, $I_{sg}(\mathbf{v}_c)$ is introduced, as shown in Eq. 11. $I_{sg}(\mathbf{v}_c)$ represents the semantic information gain of each candidate. Let n_{obj} be the number of currently semantically segmented objects in

the environment, then $I_{sg}(\mathbf{v}_c)$ is defined as:

$$I_{obj}(\mathbf{v}_c) = \begin{cases} \dfrac{1}{L(\mathbf{p}_{obj}, \mathbf{p}_{\mathbf{v}_c})} & \text{if } L(\mathbf{p}_{obj}, \mathbf{p}_{\mathbf{v}_c}) \leq I_{\text{range}}, \\ 0 & \text{otherwise}, \end{cases} \tag{13}$$

$$I_{sg}(\mathbf{v}_c) = \sum_{obj=1}^{n_{obj}} I_{obj}(\mathbf{v}_c), \tag{14}$$

where $L(\mathbf{p}_{obj}, \mathbf{p}_{\mathbf{v}_c})$ is the distance between the position of the object $\mathbf{p}_{obj}$ and the position of the candidate $\mathbf{p}_{\mathbf{v}_c}$. Position of the object $\mathbf{p}_{obj}$ is calculated as stated in Subsection IV-D. In other words, the semantic information gain of each candidate $\mathbf{v}_c$ is the sum of all visible objects from the candidate $\mathbf{v}_c$ inversely proportional to the distance of the object.

Finally, the best frontier voxel is one that maximizes the total information gain $G(\mathbf{v}_c)$:

$$\mathbf{v}_{bf} = \arg\max_{\mathbf{v}_c \in F_C} G(\mathbf{v}_c). \tag{15}$$

The best frontier voxel $\mathbf{v}_{bf}$ is forwarded as a target point to path planner module.

C. PATH PLANNING AND NAVIGATION

As soon as the best frontier point is selected, it is forwarded to a path planner as a waypoint. The robot starts to follow the planned path and navigates to the best frontier point $\mathbf{v}_{bf}$.

The path planning module includes the Rapidly-exploring Random Tree Star (RRT*) algorithm, an extension of the original RRT algorithm. Unlike the RRT, RRT* improves the convergence properties of generated paths by performing rewiring operations during the expansion phase, unlike the RRT algorithm that terminates upon finding a first feasible path. This adaptive rewiring step allows the tree to continuously refine the path as more iterations are performed, eventually converging to near-optimal solutions. The approach utilized within this paper has been developed in our previous work [51], work [51], [52], and is available online [53]. In each iteration, the planner avoids occupied voxels in the OctoMap and generates a path through the free voxels up to the best frontier point. The crucial part of the planner is the state validity checker, which evaluates the validity of configurations based on system constraints such as collision avoidance and environment-specific criteria. As new configurations are sampled or interpolated between the existing configurations during the RRT* expansion step, the state validity checker is invoked to evaluate their validity. In the practical implementation, the path planner takes a binary representation of the OctoMap as an input, which provides an efficient and compact description of the environment. The UAV is represented as a rectangular prism of appropriate dimensions within the state validity checker.

Once a target point is specified and the state validity checker is defined, the path planner creates a collision-free path from the current UAV position to the target point. However, during the path execution, the OctoMap is updated and newly discovered obstacles may appear in or near the path, as noticed in the experimental analysis within our previous work [9]. To overcome this issue, the path planner is extended to check the validity of the path during motion, as the OctoMap is updated. For each point along the path, it checks whether the UAV can execute it without interfering with obstacles, as shown in Fig. 3.

If the validity checker detects an obstacle on the path up to the current goal point, the UAV is stopped and the current goal point is classified as unreachable. The exploration planner starts a new iteration (frontiers update, the best frontier selection and path planning). This ensures that the UAV does not attempt to traverse newly discovered obstacles, increasing the safety and efficiency of the exploration. Once the goal point is set as unreachable, it is no longer considered a candidate for the best frontier during the exploration process. Re-planning to the same goal point is left for future work.

A new cycle of the procedure to determine the best frontier point is started either when the previous path is discarded or after the previous frontier point is reached by the UAV. During the exploration, the number of frontiers is changing and once the entire environment is explored and a complete map of the environment is created, the exploration process is considered done.

The path execution and UAV control are achieved using an MPC-based tracking method. The original implementation is presented in [54] while an adapted version of their work is presented in [55] and used in this paper. The main motivation for using this tracking method is that it allows the UAV to smoothly follow and quickly change the UAV trajectory based on the current system state and model dynamics. Furthermore, the tracker enables safe and stable flight, regardless of the target point resulting from the exploration planner.

VI. SIMULATION ANALYSIS

Simulations are performed in the Gazebo environment using the Robot Operating System (ROS) and a quadcopter. The quadcopter is equipped with a camera with specifications from Intel(R)RealSense(TM) Depth Camera D455. It has a horizontal and vertical FOV $\alpha_h = 90°$, $\alpha_v = 65°$, respectively, and maximum depth defined in Table 1. For collision checking, the dimensions of a rectangular prism around the UAV are set to $l = 0.6$ m, $w = 0.6$ m, $h = 0.5$ m.

The proposed ASEP algorithm is compared with the closest frontier method (CF) introduced by [3] and adapted to our planner. Additionally, it is compared to our more recent multi-resolution frontier planner (MRF) [9]. The parameters used in the MRF are set to their default values explained in [9], with velocities as given in Table 1. Both the CF and the MRF are adapted to our quadcopter, equipped with a camera, and to our control system to allow the fairest possible comparison. The approaches are compared in two scenarios

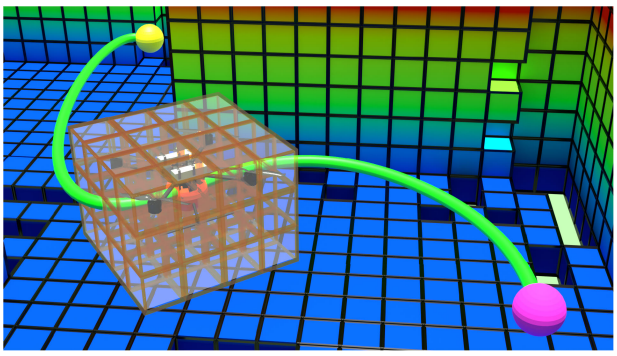
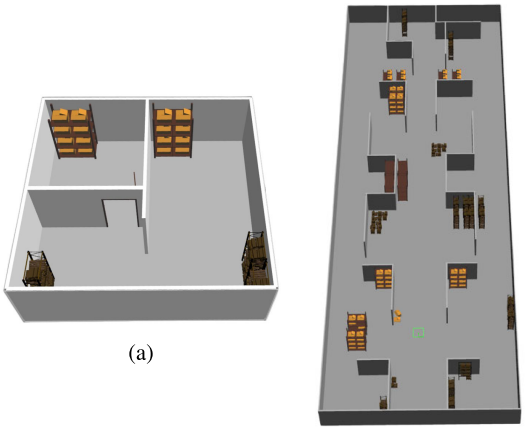


FIGURE 3. A UAV executes a planned path (green) from a start point (pink) to a target point (yellow) in an OctoMap. A rectangular prism-shaped state validity checker (transparent white and orange) simplifies the representation of the UAV. The validity checking is performed for each point (orange point) of the planned path.



(a) (b)

FIGURE 4. Gazebo warehouse scenarios. (a) Simple warehouse scenario. (b) Complex warehouse scenario.

(Fig. 4) with different environment sizes and OctoMap resolutions r. All simulations have been run 10 times on Intel(R)Core(TM) i7-10750H CPU @ 2.60GHz × 12.

The first scenario refers to a 10 m × 10 m × 3 m relatively simple warehouse (Fig. 4 (a)). The second scenario refers to a 20 m × 60 m × 3 m complex warehouse environment (Fig. 4 (b)). Please note that in the simulation analysis the odometry is provided by the simulator, while in real experiments the odometry is provided by the SLAM. This is mainly due to the fact that the camera simulation is not realistic enough to be suitable for the ORB-SLAM3 algorithm.

A. OBJECT LABELING

Within this work, the objective is to label objects in warehouse scenarios (shelves, boxes and doors). It means that N_{labels} is set to three. Note that the total number of labels in a general case is adapted to the environment and the elements expected in it. The total number of objects is initially unknown, but after exploration, it is

TABLE 1. Exploration parameters for simulation scenarios. r is OctoMap resolution, R_{max} is the maximum range of the camera, $\dot{\mathbf{p}}_{max}^{\{x,y,z\}}$ and $\dot{\psi}_{max}$ are max UAV velocities in the x, y, z axes and yaw direction, respectively, while $\ddot{\mathbf{p}}_{max}^{\{x,y,z\}}$ is the acceleration for the same degrees of freedom. I_{range} is the cuboid width and height used for information gain calculation, while λ, α and β are constants used in equations for exploration policy.

Parameter	Simple warehouse	Complex warehouse	Real world
r [m]	0.1, 0.2	0.1, 0.2	0.1, 0.2
R_{max} [m]	6.0	6.0	6.0
$\dot{\mathbf{p}}_{max}^{\{x,y,z\}}$ [m/s]	1.0	1.0	0.25
$\ddot{\mathbf{p}}_{max}^{\{x,y,z\}}$ [m/s^2]	0.5	0.5	0.2
$\dot{\psi}_{max}$ [rad/s]	0.75	0.75	0.25
I_{range} [m]	3.0	5.0	5.0
λ	0.1	0.1	0.1
α	0.35	0.35	0.35
β	0.7	0.7	0.7

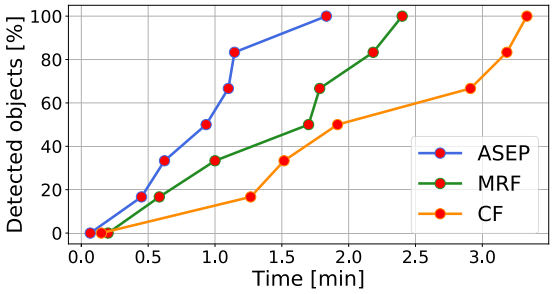


FIGURE 5. The detected objects in time for the simple warehouse environment at $r = 0.2$ m.

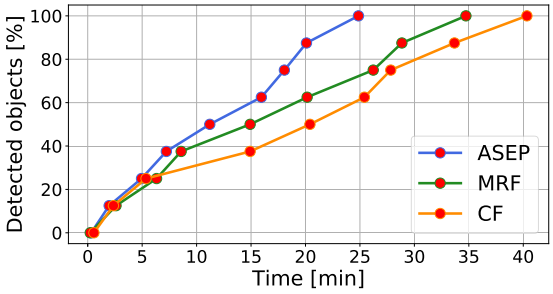


FIGURE 6. The detected objects in time for the complex warehouse environment at $r = 0.1$ m.

$N_{obj} = 30$ and $N_{obj} = 160$ for the simple and complex scenarios, respectively. All three approaches are compared in both simulation scenarios, with different resolutions. The results are shown in Fig. 5 and Fig. 6 as a percentage of detected objects in time. It can be observed that the ASEP needs less time to detect all given objects since it is directed by the semantic information from the environment. For instance, in a complex environment, all objects are detected and labeled on the map in 25 minutes when using ASEP, while MRF and CF need 35 and 40 minutes, respectively.

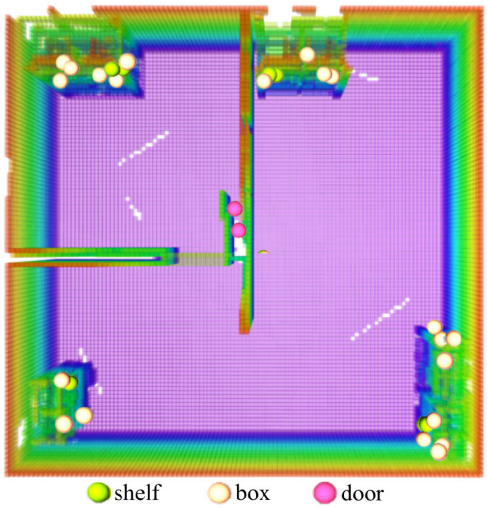


FIGURE 7. The OctoMap of the simple warehouse environment with transparency effect applied on and with detected objects during the exploration. Shelves, boxes and doors are shown with different colors on the map.

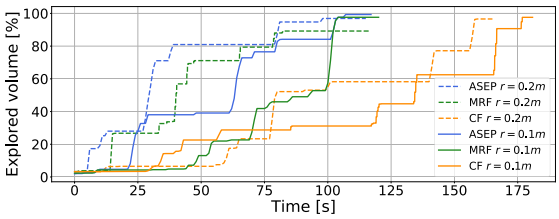


FIGURE 8. The explored volume in total exploration time for the simple warehouse scenario.

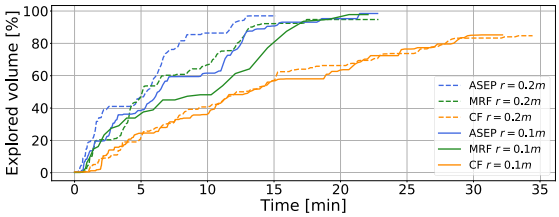


FIGURE 9. The explored volume in total exploration time for the complex warehouse scenario.

Fig. 7 shows the labeled objects on the map of the simple warehouse scenario. However, due to factors such as UAV tilting and limitations of the detection algorithm, the object positions in the OctoMap are not entirely accurate. To address this, any new object that is within 0.3 m of an existing object with the same semantic label is averaged with the centroid of the existing object, and the object position in 3D (described in IV-D) is then updated. In this way, multiple labeling of the same object is avoided. Consequently, labeled objects deviate from their real positions in the environment, given by the simulator. The mean deviation of each labeled object from its real position in the simulator is calculated by comparing the position of the semantic label to the real position of the closest object of that label obtained from the simulator. The calculated values are as follows: 0.41 m, 0.48 m, and 0.54 m for boxes, shelves, and doors, respectively. These values indicate that, considering the overall scale and dimensions of the scenarios, the position of the labeled objects matches very well with their actual position in the environment.

B. COMPUTATION PERFORMANCE

Computation times t_c and total exploration times t_{exp} for all 10 runs are shown in Table 2. The performance of all three approaches is compared in both simulation scenarios and at different resolutions. The computation time is equal to the time required to detect frontiers, update global frontiers and find the best frontier $\mathbf{v}_{bf}$. For the MRF, t_c includes time to cluster frontiers using multi-resolution frontier clustering (with exploration depth d_{exp} set to 15) and mean shift clustering algorithm, as described in [9]. The clustering methods in the MRF affect the t_c and thus the t_{exp}. It can be observed that the computation times for the MRF approach are higher than in our approach, especially when using a high-resolution map. Such results are expected since the MRF multi-resolution frontier clustering is computationally expensive, especially with a high number of frontiers. Furthermore, the use of multi-resolution frontier clustering in the MRF causes the computation time to increase as the complexity of the environment increases. On the other hand, the computation times in the CF depend mainly on the size of the environment and the number of frontiers, rather than on the map resolution (which is evident in both scenarios). For instance, in the simple scenario at the resolution $r = 0.1$ m, our planner runs more than ten times faster than the MRF. The ASEP and the CF have similar computation times since the difference is only in information gain calculation. The computation time in the proposed planner increases with the finer resolution but is still low enough not to affect the exploration. The results have confirmed that the multi-resolution frontier clustering may cause a bottleneck during exploration in larger and more complex scenarios with a large number of frontiers and at fine resolutions.

C. GLOBAL EXPLORATION USING PROPOSED PLANNER

Simulations were also performed to compare the total exploration time of our exploration planner with the MRF and the CF. The algorithms were tested using a voxel resolution of $r = 0.2$ m and $r = 0.1$ m. Note that the goal of the ASEP is to label objects of interest as soon as possible. Therefore, the placement of objects of interest in the environments may affect the total exploration time. However, a comparison of total exploration time shows that it is possible to perform exploration in a comparable time to the state-of-the-art algorithms, while in the same time doing the object labeling.

Fig. 8 shows the explored volume over time for algorithms in simple warehouse scenario. It can be observed that the

TABLE 2. The tuples of mean and standard deviation for the total exploration time t_{exp} and the computational time per iteration t_c.

		ASEP		MRF [9]		CF [3]	
Scenario	**r** [m]	$\mathbf{t_c}$ [ms]	$\mathbf{t_{exp}}$[min]	$\mathbf{t_c}$ [ms]	$\mathbf{t_{exp}}$[min]	$\mathbf{t_c}$ [ms]	$\mathbf{t_{exp}}$[min]
Simple warehouse	0.2	(5.00, 3.97)	(1.94, 0.48)	(7.61, 8.31)	(2.02, 0.55)	(8.73, 2.69)	(2.67, 0.67)
	0.1	(10.83, 7.78)	(2.01, 0.52)	(129.92, 47.21)	(2.08, 0.61)	(15.64, 5.65)	(3.01, 0.75)
Complex warehouse	0.2	(5.03, 5.41)	(18.88, 3.78)	(4.39, 25.00)	(20.85, 4.89)	(6.32, 2.51)	(33.78, 5.95)
	0.1	(8.24, 5.24)	(21.74, 3.85)	(1043.44, 163.79)	(22.67, 4.85)	(13.51, 4.75)	(35.14, 6.17)

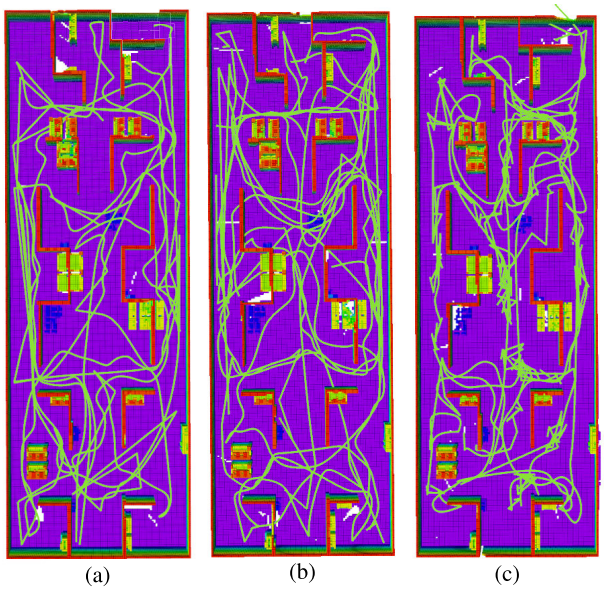


FIGURE 10. The OctoMap of the complex scenario created by the ASEP, MRF and CF, respectively, with planned paths, at the resolution $r = 0.2$ m.

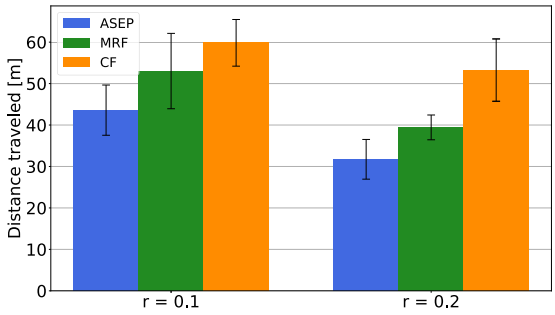


FIGURE 11. Total distance traveled in the simple environment for the ASEP, MRF and CF. Data are given as means of 10 runs with standard deviations.

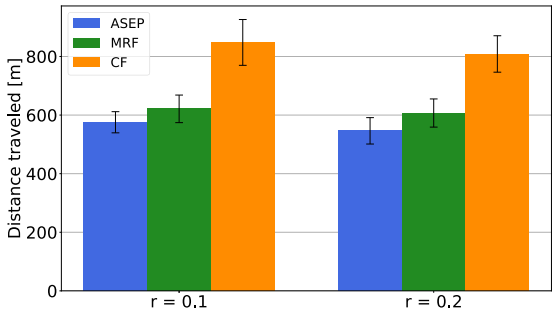


FIGURE 12. Total distance traveled in the complex environment for the ASEP, MRF and CF. Data are given as means of 10 runs with standard deviations.

ASEP and the MRF complete the exploration of the entire area at almost the same time and remarkably faster than the CF. The graph shows that our method and the MRF need around 100 s to explore the simple scenario at different map resolutions, while the CF needs more than 160 s. However, the MRF results in less explored volume, especially at lower resolutions. Namely, at the $r = 0.2$ m it explores around 92% of the environment, while the ASEP and CF explore almost 98%. This occurs because MRF uses multi-resolution clustering at lower OcTree depth $d_{exp} = 15$, leading to the next best frontier selected at d_{exp}, which is shifted from the detected frontier at the deepest level of OcTree, which is 16.

In the complex scenario, the explored volume in time is shown in Fig. 9. Our method explores the entire environment more than twice as fast as the CF and about 5 minutes faster than the MRF. ASEP and MRF behave similarly at the beginning, but over time the MRF and especially the CF show their drawbacks, which affect the total exploration time. The ASEP explores the complex environment in 18.88 min with a standard deviation of 3.78 min and in 21.74 min with a standard deviation of 3.85 min for a resolution of 0.2 m and 0.1 m, respectively. A thorough comparison of the experimental results with sampling-based approaches, such as [4] and [28] is omitted since they use different approaches for frontier generation and selection. Taking these results into consideration, it is shown that combining the semantic information from the environment in the frontier evaluation can result in a faster total exploration, but it definitely results in faster labeling of all given objects in the environment. Please note that the object arrangement influences the total exploration time. For instance, if objects of interest are tightly grouped at a single point in the environment, we expect the UAV to first circle the objects and then move to other parts of the environment. This configuration could increase the overall exploration time.

The OctoMap of the complex scenario generated by all three planners at $r = 0.2$ m is shown in Fig. 10 along with the corresponding UAV paths. Note that some of the paths shown are not fully executed because the collision

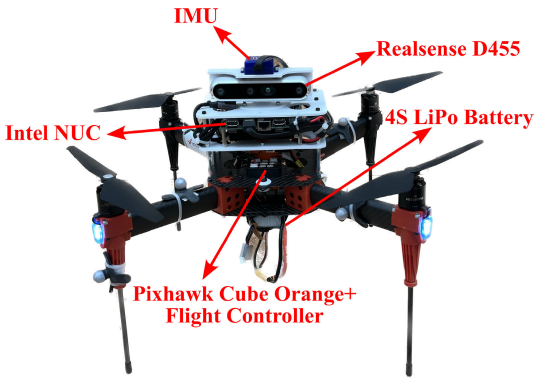


FIGURE 13. A *Hexsoon EDU-450* quadcopter equipped with a *Intel NUC*, a Realsense *D455* camera, an IMU, a battery and a flight controller.

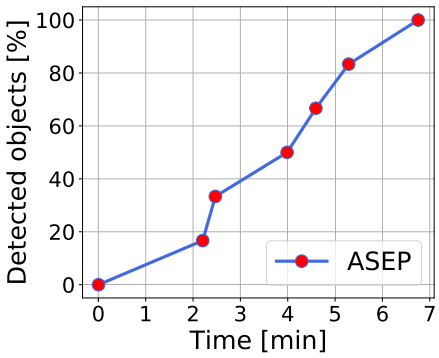


FIGURE 14. The detected objects in time for the real-world environment at $r = 0.1$ m.

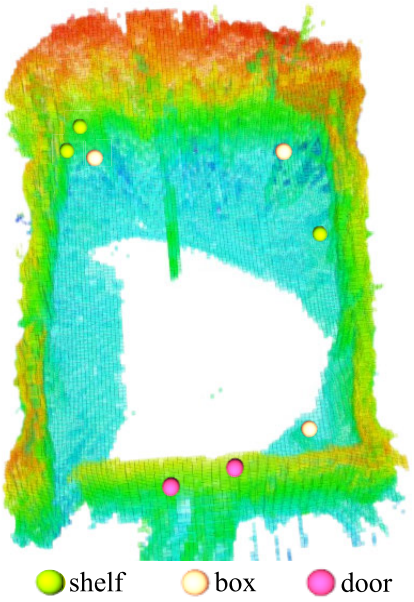


FIGURE 15. The OctoMap of the real-world environment with transparency effect applied on and with detected objects during the exploration of a real-world scenario. Shelves, boxes and doors are shown with different colors on the map.

checker was triggered. In Fig. 11 and Fig. 12, the distance traveled by the UAV is visualized in both simple and complex scenarios at different resolutions. The metric for the distance traveled is derived from the UAV odometry obtained from the simulator. The ASEP achieves the shortest path planned, as well as traveled among the three algorithms, while CF shows a tendency towards larger traveled distances. This was particularly evident in the complex environment at $r = 0.1$ m where the CF recorded an average distance of 847.98 m and the ASEP 575.50 m. The nature of the CF algorithm often results in significant back-and-forth movement, which can lead to a less efficient exploration trajectory (Fig. 10(c)). The distance traveled for the MRF is similar to the ASEP. Namely, for the simple warehouse scenario at $r = 0.1$ m the UAV traveled on average 43.59 m and 53.02 m by ASEP and MRF respectively, while at $r = 0.2$ m, 31.73 m and 39.43 m.

VII. EXPERIMENTAL ANALYSIS

A. SETUP

For our indoor experimental analysis, a *Hexsoon EDU-450* quadcopter is used (Fig. 13) which features four *T-motors* HS2216 920KV motors attached to a carbon fiber frame. The dimensions of the UAV are 0.36 m × 0.36 m × 0.3 m, which makes it a relatively small UAV suitable for indoor environments. The total flight time of the UAV is around 8 min with a mass of m = 2.5 kg, including batteries, electronics and sensory apparatus. The *Cube Orange+* flight controller unit is attached to the center of the UAV body, and it is responsible for the low-level attitude control of the vehicle. Furthermore, the UAV is equipped with an *Intel NUC*, i7-8650U CPU @ 1.90GHz × 8, onboard computer for collecting and processing sensory data. The onboard computer runs *Linux Ubuntu 18.04* with *ROS Melodic* framework that communicates with the autopilot through a serial interface. The UAV is equipped with a Realsense *D*455 camera with a maximum range of 6 m. The parameters used in the real world are stated in Table 1. The experiments are performed in an environment of 10 m × 8 m × 3 m. Experimental evaluations were tested using a voxel resolution of $r = 0.2$ m and $r = 0.1$ m, the same as in the simulation analysis.

B. RESULTS AND DISCUSSION

Running the planner with limited onboard resources and in real-time, the exploration and object labeling are demonstrated. As in the simulation scenario, N_{labels} is set to three, while $N_{obj} = 8$ is detected during exploration. The percentage of detected objects in time is shown in Fig. 14. Fig. 15 shows the labeled objects on the map of the real-world scenario at $r = 0.1$ m.

Computation times t_c and total exploration times t_{exp} for both runs are shown in Table 3. The average computation time is comparable to the time achieved in the simulation

TABLE 3. The tuples of mean and standard deviation for the total exploration time t_{exp} and the computational time per iteration t_c for the real-world scenario.

Scenario	r [m]	t_c [ms]	t_{exp}[min]
Real world	0.2	(3.95, 5.68)	(6.96, 1.08)
	0.1	(24.60, 36.60)	(7.95, 1.52)

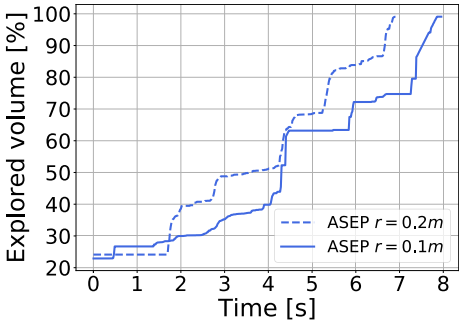


FIGURE 16. The explored volume in time for the real-world scenario.

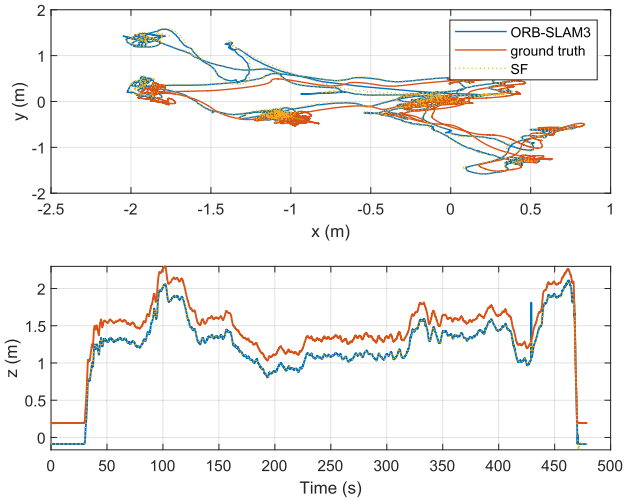


FIGURE 17. Comparison of ORB-SLAM3, multi-sensor fusion and ground truth positions for the x, y, and z coordinates in a single exploration run.

setup. Fig. 16 shows that the total exploration time is about 7 minutes for $r = 0.2$ m and about 8 minutes for $r = 0.1$ m. The result of the exploration is the OctoMap of the environment shown in Fig. 15, in which the objects labeled during the exploration are also shown. Running the planner in the real world and in real time, the successful exploration and object labeling is demonstrated while running the SLAM, semantic segmentation and exploration on the UAV with limited onboard resources.

ORB-SLAM3 algorithm is used for localization during real-world exploration. The positions estimated by the ORB-SLAM3, multi-sensor fusion (SF), and ground truth are shown in Fig. 17. It is noteworthy that the ORB-SLAM3 and SF are very similar and both show a deviation from the ground truth position over time, emphasizing the importance of ongoing calibration. The results highlight the potential of the ORB-SLAM3 algorithm and the multi-sensor fusion approach for reliable localization suitable for exploration.

VIII. CONCLUSION

This paper deals with a novel semantically-enhanced frontier-based exploration planner called ASEP. The ASEP is capable of autonomously exploring a previously unknown GPS-denied area, creating an occupancy grid map OctoMap and labeling objects of interest in the OctoMap. Results show improved behavior in terms of time needed to label all objects in the environment compared to state-of-the-art strategies. An exploration strategy that combines both geometric and semantic information from the environment speeds up the exploration of all objects, while a novel object labeling algorithm ensures real-time object detection and evaluation. This 3D exploration planner has been successfully tested in both simulation scenarios and a real-world experiment using a quadcopter equipped with a camera. Video recordings of the semantically augmented 3D exploration can be found on YouTube [56].

REFERENCES

[1] UNIZG-FER and LARICS. *AeroSTREAM Project*. Accessed: Jun. 6, 2023. [Online]. Available: https://aerostream.fer.hr/aerostream

[2] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auto. Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.

[3] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. Int. Symp. Comput. Intell. Robot. Automat. (CIRA)*, Jul. 1997, pp. 146–151.

[4] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon 'next-best-view' planner for 3D exploration," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 1462–1468.

[5] R. Ashour, T. Taha, J. M. M. Dias, L. Seneviratne, and N. Almoosa, "Exploration for object mapping guided by environmental semantics using UAVs," *Remote Sens.*, vol. 12, no. 5, p. 891, Mar. 2020.

[6] R. P. de Figueiredo, J. le Fevre Sejersen, J. G. Hansen, M. Brandão, and E. Kayacan, "Real-time volumetric-semantic exploration and mapping: An uncertainty-aware approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 9064–9070.

[7] B. Yu, H. Kasaei, and M. Cao, "Frontier semantic exploration for visual target navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 4099–4105.

[8] T. Dang, C. Papachristos, and K. Alexis, "Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2526–2533.

[9] A. Batinovic, T. Petrovic, A. Ivanovic, F. Petric, and S. Bogdan, "A multi-resolution frontier-based planner for autonomous 3D exploration," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4528–4535, Jul. 2021.

[10] M. Juliá, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Auto. Robots*, vol. 33, no. 4, pp. 427–444, Nov. 2012.

[11] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unknown indoor environments," *Int. J. Micro Air Vehicles*, vol. 1, no. 4, pp. 217–228, Dec. 2009.

[12] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 2135–2142.

[13] C. Zhu, R. Ding, M. Lin, and Y. Wu, "A 3D frontier-based exploration tool for MAVs," in *Proc. IEEE 27th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2015, pp. 348–352.

[14] A. Mannucci, S. Nardi, and L. Pallottino, "Autonomous 3D exploration of large areas: A cooperative frontier-based approach," in *Proc. Int. Workshop Modelling Simulation Auton. Syst.*, 2017, pp. 18–39.

[15] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, "Fast frontier-based information-driven autonomous exploration with an MAV," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 9570–9576.

[16] M. Faria, R. Marín, M. Popović, I. Maza, and A. Viguria, "Efficient lazy Theta* path planning over a sparse grid to explore large 3D volumes with a multirotor UAV," *Sensors*, vol. 19, no. 1, p. 174, Jan. 2019.

[17] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "FUEL: Fast UAV exploration using incremental frontier structure and hierarchical planning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 779–786, Apr. 2021.

[18] B. Fang, J. Ding, and Z. Wang, "Autonomous robotic exploration based on frontier point optimization and multistep path planning," *IEEE Access*, vol. 7, pp. 46104–46113, 2019.

[19] H. H. González-Baños and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *Int. J. Robot. Res.*, vol. 21, nos. 10–11, pp. 829–848, Oct. 2002.

[20] T. Baiming, S. Jicheng, D. Chaofan, and L. Qingbao, "A target point based MAV 3D exploration method," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2018, pp. 2406–2413.

[21] D. Joho, C. Stachniss, P. Pfaff, and W. Burgard, "Autonomous exploration for 3D map learning," in *Autonome Mobile Systeme*. Cham, Switzerland: Springer, 2007, pp. 22–28.

[22] S. M. Lavalle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*. Boca Raton, FL, USA: CRC Press, 2000, pp. 293–308.

[23] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia (ICRA)*, Apr. 2000, pp. 995–1001.

[24] D. Deng, Z. Xu, W. Zhao, and K. Shimada, "Frontier-based automatic-differentiable information gain measure for robotic exploration of unknown 3D environments," 2020, *arXiv:2011.05288*.

[25] C. Witting, M. Fehr, R. Bähnemann, H. Oleynikova, and R. Siegwart, "History-aware autonomous exploration in confined environments using MAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–9.

[26] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1500–1507, Apr. 2020.

[27] A. Batinovic, A. Ivanovic, T. Petrovic, and S. Bogdan, "A shadowcasting-based Next-Best-View planner for autonomous 3D exploration," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2969–2976, Apr. 2022.

[28] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt, "Efficient autonomous exploration planning of large-scale 3-D environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1699–1706, Apr. 2019.

[29] Z. Meng, H. Qin, Z. Chen, X. Chen, H. Sun, F. Lin, and M. H. Ang, "A two-stage optimized next-view planning framework for 3-D unknown environment exploration, and structural reconstruction," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1680–1687, Jul. 2017.

[30] V. M. Respall, D. Devitt, R. Fedorenko, and A. Klimchik, "Fast sampling-based Next-Best-View exploration algorithm for a MAV," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 89–95.

[31] C. Wang, D. Zhu, T. Li, M. Q.-H. Meng, and C. W. de Silva, "Efficient autonomous robotic exploration with semantic road map in indoor environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2989–2996, Jul. 2019.

[32] C. Gomez, M. Fehr, A. Millane, A. C. Hernandez, J. Nieto, R. Barber, and R. Siegwart, "Hybrid topological and 3D dense mapping through autonomous exploration for large indoor environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 9673–9679.

[33] Y. Wang, D. Ewert, D. Schilberg, and S. Jeschke, "Edge extraction by merging the 3D point cloud and 2D image data," in *Automation, Communication and Cybernetics in Science and Engineering 2013/2014*. Cham, Switzerland: Springer, 2014, pp. 773–785.

[34] E. P. H. Alarcón, D. B. Ghavifekr, G. Baris, M. Mugnai, M. Satler, and C. A. Avizzano, "An efficient object-oriented exploration algorithm for unmanned aerial vehicles," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2021, pp. 330–337.

[35] K. Ota, Y. Sasaki, D. K. Jha, Y. Yoshiyasu, and A. Kanezaki, "Efficient exploration in constrained environments with goal-oriented reference path," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 6061–6068.

[36] L. Campos-Macías, R. Aldana-López, R. de la Guardia, J. I. Parra-Vilchis, and D. Gómez-Gutiérrez, "Autonomous navigation of MAVs in unknown cluttered environments," *J. Field Robot.*, vol. 38, no. 2, pp. 307–326, May 2020.

[37] W. Kwon, J. H. Park, M. Lee, J. Her, S.-H. Kim, and J.-W. Seo, "Robust autonomous navigation of unmanned aerial vehicles (UAVs) for warehouses' inventory application," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 243–249, Jan. 2020.

[38] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[39] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LiDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1271–1278.

[40] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[41] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

[42] L. Markovic, M. Kovac, R. Milijas, M. Car, and S. Bogdan, "Error state extended Kalman filter multi-sensor fusion for unmanned aerial vehicle localization in GPS and magnetometer denied indoor environments," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2022, pp. 184–190.

[43] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5686–5696.

[44] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[45] *Semantic Segmentation on MIT ADE20K Dataset in PyTorch*. Accessed: Feb. 21, 2023. [Online]. Available: https://github.com/CSAILVision/semantic-segmentation-pytorch

[46] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, Mar. 2019.

[47] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5122–5130.

[48] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[49] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[50] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 3213–3223, doi: 10.1109/CVPR.2016.350.

[51] B. Arbanas, A. Ivanovic, M. Car, M. Orsag, T. Petrovic, and S. Bogdan, "Decentralized planning and control for UAV-UGV cooperative teams," *Auto. Robots*, vol. 42, no. 8, pp. 1601–1618, Dec. 2018.

[52] A. Ivanovic and M. Orsag, "Parabolic airdrop trajectory planning for multirotor unmanned aerial vehicles," *IEEE Access*, vol. 10, pp. 36907–36923, 2022.

[53] UNIZG-FER and LARICS. *Path and Trajectory Planning*. Accessed: Jun. 6, 2023. [Online]. Available: https://github.com/larics/larics_motion_planning

[54] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 6753–6760.

[55] A. Batinovic, J. Goricanec, L. Markovic, and S. Bogdan, "Path planning with potential field-based obstacle avoidance in a 3D environment by an unmanned aerial vehicle," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2022, pp. 394–401.

[56] *ASEP: An Autonomous Semantic Exploration Planner with Object Labeling*. Accessed: Aug. 10, 2023. [Online]. Available: https://www.youtube.com/playlist?list=PLC0C6uwoEQ8ZHLRJLCw2YEaQE4aR-Y65f

ANA MILAS (Member, IEEE) received the B.S.E.E. and M.S.E.E. degrees from the University of Zagreb, Croatia, in 2016 and 2018, respectively, where she is currently pursuing the Ph.D. degree. She is also a Research Associate with the Faculty of Electrical Engineering and Computing, University of Zagreb. In 2019, she did an internship with the University of Seville, Spain. In 2020, she attended the IEEE RAS Summer School on Multi-Robot Systems, Prague, Czech Republic. As a Ph.D. student, she participated in a Mohamed Bin Zayed International Robotics Challenge (MBZIRC2020) and MBZIRC2023 Project. Her research interests include aerial robotics, autonomous mapping, and exploration. During the bachelor's degree, she received the University of Zagreb Rector Award for the work titled "Coordinated Multi-Robot Exploration Based on Graph SLAM Method and Rapidly Exploring Random Trees," in 2018.

ANTUN IVANOVIC (Student Member, IEEE) received the M.S.E.E. and Ph.D. degrees from the University of Zagreb, Croatia, in 2015 and 2023, respectively.

He is currently a Postdoctoral Researcher with the Laboratory for Robotics and Intelligent Control Systems, University of Zagreb Faculty of Electrical Engineering and Computing (UNIZG-FER), where he joined, in 2015. During the bachelor's degree, he received the Rector's Award for the work titled "Augmented Human Machine Interface for Aerial Manipulators." His research interests include robotics, unmanned aerial vehicles, aerial manipulation, and motion planning. As a Ph.D. student, he participated in a European Robotics Challenge (EuRoC) and MBZIRC2020 (Mohamed Bin Zayed International Robotics Challenge) competitions, and multiple international projects. He is also working on several EU or nationally-funded projects, including Specularia, VirtualUAV, and ASAP. In 2018, he was a Visiting Researcher with the United States Military Academy West Point, USA, where he collaborated on work related to aerial-ground cooperative manipulation.

TAMARA PETROVIC (Member, IEEE) received the master's degree from the University of Zagreb Faculty of Electrical Engineering and Computing (UNIZG-FER), in 2007. Her Ph.D. dissertation was titled "Centralized Control of Variable Structure Multi-Vehicle Systems Based on Resource Allocation," in 2014.

Since 2008, she has been with UNIZG-FER, LARICS, as a Research Fellow, and since 2014, she has been a postgraduate. Her research interests include multi-robot systems and discrete event systems. She has participated in the work of several international and domestic scientific research projects. During her studies, she was a recipient of the bronze plaque "Josip Lonar," in 2007, and three "Josip Lonar" awards, in 2003, 2004, and 2005, for outstanding academic success. She published two book chapters, six journal articles, and 17 conference papers. She participates as a reviewer in several international scientific journals and conferences. She participates in teaching activities with the Department of Control and Computer Engineering, which include exercises and lectures, co-mentoring of students, and improvement of teaching content and materials. In 2016 and 2017, she was carried out the duties of the Faculty Erasmus+ Coordinator for traineeships and the mobility of control engineering and automation students.

Bibliography

[1] Zhu, C., Ding, R., Lin, M., Wu, Y., “A 3D frontier-based exploration tool for mavs”, in 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), 2015, pp. 348-352.

[2] Mannucci, A., Nardi, S., Pallottino, L., “Autonomous 3D exploration of large areas: A cooperative frontier-based approach”, in Modelling and Simulation for Autonomous Systems, Vol. 10756, 2018, pp. 18–39.

[3] Burgard, W., Moors, M., Stachniss, C., Schneider, F., “Coordinated multi-robot exploration”, IEEE Transactions on Robotics, Vol. 21, No. 3, 2005, pp. 376–386.

[4] Faria, M., Marín, R., Popović, M., Maza, I., Viguria, A., “Efficient lazy theta* path planning over a sparse grid to explore large 3D volumes with a multirotor UAV”, Sensors, Vol. 19, No. 1, 2019, pp. 174.

[5] Oršulić, J., Miklić, D., Kovačić, Z., “Efficient dense frontier detection for 2-D graph slam based on occupancy grid submaps”, IEEE Robotics and Automation Letters, Vol. 4, No. 4, 2019, pp. 3569-3576.

[6] Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., Siegwart, R., “Receding horizon "next-best-view" planner for 3D exploration”, in 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 1462-1468.

[7] Cieslewski, T., Kaufmann, E., Scaramuzza, D., “Rapid exploration with multi-rotors: A frontier selection method for high speed flight”, in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017.

[8] Selin, M., Tiger, M., Duberg, D., Heintz, F., Jensfelt, P., “Efficient autonomous exploration planning of large-scale 3D environments”, IEEE Robotics and Automation Letters, Vol. 4, No. 2, 2019, pp. 1699-1706.

[9] Yamauchi, B., "A frontier-based approach for autonomous exploration", in Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97., 1997, pp. 146-151.

[10] Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M., "Real-time 3d reconstruction at scale using voxel hashing", ACM Transactions on Graphics, Vol. 32, No. 6, Nov. 2013, pp. 1–11.

[11] Kähler, O., Prisacariu, V., Valentin, J., Murray, D., "Hierarchical voxel block hashing for efficient integration of depth images", IEEE Robotics and Automation Letters, Vol. 1, No. 1, 2016, pp. 192-197.

[12] Meagher, D., "Geometric modeling using octree encoding", Computer Graphics and Image Processing, Vol. 19, No. 2, 1982, pp. 129-147.

[13] Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., Burgard, W., "OctoMap: an efficient probabilistic 3D mapping framework based on octrees", Autonomous Robots, Vol. 34, No. 3, 2013, pp. 189–206.

[14] Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., Nieto, J., "Voxblox: Incremental 3D euclidean signed distance fields for on-board MAV planning", in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 1366-1373.

[15] Oleynikova, Helen, Millane, Alexander, Taylor, Zachary, Galceran, Enric, Nieto, Juan, Siegwart, Roland, "Signed distance fields: A natural representation for both mapping and planning", RSS 2016 Workshop: Geometry and Beyond - Representations, Physics, and Scene Understanding for Robotics, 2016.

[16] Narita, G., Seno, T., Ishikawa, T., Kaji, Y., "Panopticfusion: Online volumetric semantic mapping at the level of stuff and things", in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 4205-4212.

[17] Pham, Q.-H., Hua, B.-S., Nguyen, T., Yeung, S.-K., "Real-time progressive 3d semantic segmentation for indoor scenes", in 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019, pp. 1089-1098.

[18] McCormac, J., Handa, A., Davison, A., Leutenegger, S., "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks", in 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 4628-4635.

[19] Grinvald, M., Furrer, F., Novkovic, T., Chung, J. J., Cadena, C., Siegwart, R., Nieto, J., "Volumetric instance-aware semantic mapping and 3d object discovery", IEEE Robotics and Automation Letters, Vol. 4, No. 3, 2019, pp. 3037-3044.

[20] Rosinol, A., Abate, M., Chang, Y., Carlone, L., "Kimera: an open-source library for real-time metric-semantic localization and mapping", in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 1689-1696.

[21] Yang, S., Huang, Y., Scherer, S., "Semantic 3d occupancy mapping through efficient high order crfs", in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 590-597.

[22] Hess, W., Kohler, D., Rapp, H., Andor, D., "Real-time loop closure in 2D lidar slam", in 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 1271-1278.

[23] MacKay, D. J. C., Information Theory, Inference, and Learning Algorithms. Copyright Cambridge University Press, 2003.

[24] González-Baños, H. H., Latombe, J. C., "Navigation strategies for exploring indoor environments", The International Journal of Robotics Research, Vol. 21, No. 10-11, 2002, pp. 829–848.

[25] Shannon, C. E., Weaver, W., The Mathematical Theory of Communication. University of Illinois Press, 1949.

[26] Palazzolo, E., Stachniss, C., "Information-driven autonomous exploration for a vision-based mav", ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-2/W3, 2017, pp. 59–66.

[27] Palazzolo, E., Stachniss, C., "Effective exploration for mavs based on the expected information gain", Drones, Vol. 2, No. 1, 2018.

[28] Umari, H., Mukhopadhyay, S., "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees", in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, September 2017.

[29] Kaufman, E., Takami, K., Ai, Z., Lee, T., "Autonomous quadrotor 3d mapping and exploration using exact occupancy probabilities", in 2018 Second IEEE International Conference on Robotic Computing (IRC), 2018, pp. 49-55.

[30] Bissmarck, F., Svensson, M., Tolt, G., "Efficient algorithms for next best view evaluation", in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 5876-5883.

[31] Vasquez-Gomez, J. I., Sucar, L. E., Murrieta-Cid, R., "Hierarchical ray tracing for fast volumetric next-best-view planning", in 2013 International Conference on Computer and Robot Vision, 2013, pp. 181-187.

[32] Witting, C., Fehr, M., Bähnemann, R., Oleynikova, H., Siegwart, R., "History-aware autonomous exploration in confined environments using MAVs", in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1-9.

[33] Dang, T., Papachristos, C., Alexis, K., "Autonomous exploration and simultaneous object search using aerial robots", in 2018 IEEE Aerospace Conference, 2018, pp. 1-7.

[34] Papachristos, C., Khattak, S., Alexis, K., "Uncertainty-aware receding horizon exploration and mapping using aerial robots", in 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 4568-4575.

[35] Elvins, T. T., "A survey of algorithms for volume visualization", ACM SIGGRAPH Computer Graphics, Vol. 26, No. 3, 1992, pp. 194–201.

[36] Bergström, B., "FOV using recursive shadowcasting", http://www.roguebasin.com/index.php?title=FOV_using_recursive_shadowcasting.

[37] Hughes, J. F., van Dam, A., McGuire, M., Sklar, D. F., Foley, J. D., Feiner, S., Akeley, K., Computer Graphics: Principles and Practice, 3rd ed. Upper Saddle River, NJ: Addison-Wesley, 2013.

[38] Debenham, E., Solis-Oba, R., "Efficient field of vision algorithms for large 2D grids", International Journal of Computer Science and Information Technology (IJCSIT), Vol. 13, No. 1, 2021.

[39] Debenham, E., "New algorithms for computing field of vision over 2D grids", Electronic Thesis and Dissertation Repository. 6552. The University of Western Ontario, 2019.

[40] Massague Respall, V., Devitt, D., Fedorenko, R., Klimchik, A., "Fast sampling-based next-best-view exploration algorithm for a MAV", in 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021.

[41] Hart, P. E., Nilsson, N. J., Raphael, B., "A formal basis for the heuristic determination of minimum cost paths", IEEE Transactions on Systems Science and Cybernetics, Vol. 4, No. 2, 1968, pp. 100-107.

[42] Dijkstra, E. W., "A note on two problems in connexion with graphs", Numerische Mathematik, Vol. 1, No. 1, Dec. 1959, pp. 269–271.

[43] Lavalle, S. M., Kuffner, J. J., Jr., "Rapidly-exploring random trees: Progress and prospects", in Algorithmic and Computational Robotics: New Directions, 2000, pp. 293–308.

[44] Kuffner, J., LaValle, S., "RRT-connect: An efficient approach to single-query path planning", in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings, Vol. 2, 2000, pp. 995-1001.

[45] Arbanas, B., Ivanovic, A., Car, M., Orsag, M., Petrovic, T., Bogdan, S., "Decentralized planning and control for UAV-UGV cooperative teams", Autonomous Robots, Vol. 42, No. 8, 2018, pp. 1601–1618.

[46] Ivanovic, A., Orsag, M., "Parabolic airdrop trajectory planning for multirotor unmanned aerial vehicles", IEEE Access, Vol. 10, 2022, pp. 36 907-36 923.

[47] Juliá, M., Gil, A., Reinoso, O., "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments", Autonomous Robots, Vol. 33, No. 4, 2012, pp. 427–444.

[48] Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., Younes, H., "Coordination for multi-robot exploration and mapping; aaai", 2000.

[49] Moorehead, S. J., Simmons, R., Whittaker, W. L., "Autonomous exploration using multiple sources of information", in Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), Vol. 3. IEEE, 2001, pp. 3098–3103.

[50] Carlone, L., Lyons, D., "Uncertainty-constrained robot exploration: A mixed-integer linear programming approach", in 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014, pp. 1140–1147.

[51] Mei, Y., Lu, Y.-H., Lee, C. G., Hu, Y. C., "Energy-efficient mobile robot exploration", in Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. IEEE, 2006, pp. 505–511.

[52] Ström, D. P., Bogoslavskyi, I., Stachniss, C., "Robust exploration and homing for autonomous robots", Robotics and Autonomous Systems, Vol. 90, 2017, pp. 125–135.

[53] Gautam, A., Murthy, J. K., Kumar, G., Ram, S. A., Jha, B., Mohan, S., "Cluster, allocate, cover: An efficient approach for multi-robot coverage", in 2015 IEEE International Conference on Systems, Man, and Cybernetics. IEEE, 2015, pp. 197–203.

[54] Bachrach, A., He, R., Roy, N., "Autonomous flight in unknown indoor environments", International Journal of Micro Air Vehicles, Vol. 1, No. 4, 2009, pp. 217–228.

[55] Batinovic, A., Petrovic, T., Ivanovic, A., Petric, F., Bogdan, S., "A multi-resolution frontier-based planner for autonomous 3D exploration", IEEE Robotics and Automation Letters, Vol. 6, No. 3, 2021, pp. 4528-4535.

[56] Dai, A., Papatheodorou, S., Funk, N., Tzoumanikas, D., Leutenegger, S., "Fast frontier-based information-driven autonomous exploration with an MAV", in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 9570-9576.

[57] Senarathne, P. G. C. N., Wang, D., "Towards autonomous 3D exploration using surface frontiers", in 2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, October 2016.

[58] Fang, B., Ding, J., Wang, Z., "Autonomous robotic exploration based on frontier point optimization and multistep path planning", IEEE Access, Vol. 7, 2019, pp. 46 104-46 113.

[59] Stachniss, C., Grisetti, G., Burgard, W., "Information gain-based exploration using rao-blackwellized particle filters.", in Robotics: Science and systems, Vol. 2, 2005, pp. 65–72.

[60] Deng, D., Xu, Z., Zhao, W., Shimada, K., "Frontier-based automatic-differentiable information gain measure for robotic exploration of unknown 3D environments", https://arxiv.org/abs/2011.05288, 2020.

[61] Zhong, P., Chen, B., Lu, S., Meng, X., Liang, Y., "Information-driven fast marching autonomous exploration with aerial robots", IEEE Robotics and Automation Letters, Vol. 7, No. 2, 2022, pp. 810-817.

[62] Zhou, X., Yi, Z., Liu, Y., Huang, K., Huang, H., "Survey on path and view planning for uavs", Virtual Reality & Intelligent Hardware, Vol. 2, No. 1, 2020, pp. 56–69.

[63] Baiming, T., Jicheng, S., Chaofan, D., Qingbao, L., "A target point based MAV 3D exploration method", in 2018 IEEE International Conference on Mechatronics and Automation (ICMA), 2018.

[64] Joho, D., Stachniss, C., Pfaff, P., Burgard, W., "Autonomous exploration for 3D map learning", in Autonome Mobile Systeme. Springer, 2007, pp. 22–28.

[65] Schmid, L., Pantic, M., Khanna, R., Ott, L., Siegwart, R., Nieto, J., "An efficient sampling-based method for online informative path planning in unknown environments", IEEE Robotics and Automation Letters, Vol. 5, No. 2, 2020, pp. 1500-1507.

[66] Vasquez-Gomez, J. I., Sucar, L. E., Murrieta-Cid, R., Lopez-Damian, E., "Volumetric next-best-view planning for 3D object reconstruction with positioning error", Int. Jour. of Adv. Robotic Systems, Vol. 11, No. 10, 2014, pp. 159.

[67] Dornhege, C., Kleiner, A., "A frontier-void-based approach for autonomous exploration in 3D", Advanced Robotics, Vol. 27, No. 6, 2013, pp. 459–468.

[68] Zhou, B., Zhang, Y., Chen, X., Shen, S., "Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning", IEEE Robotics and Automation Letters, Vol. 6, No. 2, 2021, pp. 779–786.

[69] Meng, Z., Qin, H., Chen, Z., Chen, X., Sun, H., Lin, F., Ang, M. H., "A two-stage optimized next-view planning framework for 3-D unknown environment exploration, and structural reconstruction", IEEE Robotics and Automation Letters, Vol. 2, No. 3, 2017, pp. 1680-1687.

[70] Zhou, B., Xu, H., Shen, S., "Racer: Rapid collaborative exploration with a decentralized multi-uav system", IEEE Transactions on Robotics, 2023.

[71] Wang, C., Zhu, D., Li, T., Meng, M. Q.-H., de Silva, C. W., "Efficient autonomous robotic exploration with semantic road map in indoor environments", IEEE RAL, Vol. 4, No. 3, 2019, pp. 2989–2996.

[72] Gomez, C., Fehr, M., Millane, A., Hernandez, A. C., Nieto, J., Barber, R., Siegwart, R., "Hybrid topological and 3d dense mapping through autonomous exploration for large indoor environments", in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 9673-9679.

[73] de Figueiredo, R. P., le Fevre Sejersen, J., Hansen, J. G., Brandão, M., Kayacan, E., "Real-time volumetric-semantic exploration and mapping: An uncertainty-aware approach", in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021, pp. 9064-9070.

[74] Ashour, R., Taha, T., Dias, J. M. M., Seneviratne, L., Almoosa, N., "Exploration for object mapping guided by environmental semantics using uavs", Remote Sensing, Vol. 12, No. 5, 2020.

[75] Dang, T., Papachristos, C., Alexis, K., "Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics", in 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 2526-2533.

[76] Wang, Y., Ewert, D., Schilberg, D., Jeschke, S., "Edge extraction by merging the 3d point cloud and 2d image data", in Automation, Communication and Cybernetics in Science and Engineering 2013/2014. Springer International Publishing, 2014, pp. 773–785.

[77] Yu, B., Kasaei, H., Cao, M., "Frontier semantic exploration for visual target navigation", in 2023 IEEE International Conference on Robotics and Automation (ICRA), 2023, pp. 4099-4105.

[78] Alarcón, E. P. H., Ghavifekr, D. B., Baris, G., Mugnai, M., Satler, M., Avizzano, C. A., "An efficient object-oriented exploration algorithm for unmanned aerial vehicles", in 2021 International Conference on Unmanned Aircraft Systems (ICUAS), 2021, pp. 330-337.

[79] Ota, K., Sasaki, Y., Jha, D. K., Yoshiyasu, Y., Kanezaki, A., "Efficient exploration in constrained environments with goal-oriented reference path", in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020.

[80] Shelhamer, E., Long, J., Darrell, T., "Fully convolutional networks for semantic segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 4, 2017, pp. 640-651.

[81] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L., "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs", IEEE transactions on pattern analysis and machine intelligence, Vol. 40, No. 4, April 2018, pp. 834—848.

[82] Ronneberger, O., Fischer, P., Brox, T., "U-net: Convolutional networks for biomedical image segmentation", in Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 234–241.

[83] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., "Pyramid scene parsing network", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[84] He, K., Gkioxari, G., Dollar, P., Girshick, R., "Mask r-CNN", in 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, Oct. 2017.

[85] Ren, S., He, K., Girshick, R., Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 6, 2017, pp. 1137-1149.

[86] Sun, K., Xiao, B., Liu, D., Wang, J., "Deep high-resolution representation learning for human pose estimation", in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5686-5696.

[87] Csurka, G., Volpi, R., Chidlovskii, B., "Semantic image segmentation: Two decades of research", 2023.

[88] "Semantic Segmentation on MIT ADE20K Dataset in Pytorch", https://github.com/CSAILVision/semantic-segmentation-pytorch, accessed: 2023-02-21.

[89] Zhou, B., Gao, F., Wang, L., Liu, C., Shen, S., "Robust and efficient quadrotor trajectory generation for fast autonomous flight", IEEE Robotics and Automation Letters, Vol. 4, No. 4, 2019, pp. 3529–3536.

[90] Song, S., Jo, S., "Online inspection path planning for autonomous 3D modeling using a micro-aerial vehicle", in 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 6217-6224.

[91] Dornhege, C., Kleiner, A., "A frontier-void-based approach for autonomous exploration in 3D", in 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, 2011, pp. 351-356.

[92] Fukunaga, K., Hostetler, L., "The estimation of the gradient of a density function, with applications in pattern recognition", IEEE Transactions on Information Theory, Vol. 21, No. 1, 1975, pp. 32–40.

[93] Duda, R. O., Hart, P. E., Stork, D. G., Pattern Classification. Wiley, 2001.

[94] Karaman, S., Frazzoli, E., "Sampling-based algorithms for optimal motion planning", The International Journal of Robotics Research, Vol. 30, No. 7, Jun. 2011, pp. 846–894.

[95] Gao, J., Ye, W., Guo, J., Li, Z., "Deep reinforcement learning for indoor mobile robot path planning", Sensors, Vol. 20, No. 19, Sep. 2020, pp. 5493.

[96] Pham, H., Pham, Q., "A new approach to time-optimal path parameterization based on reachability analysis", IEEE Transactions on Robotics, Vol. 34, No. 3, 2018, pp. 645-659.

[97] Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A., "Semantic understanding of scenes through the ade20k dataset", International Journal on Computer Vision, 2018.

[98] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A., "Scene parsing through ade20k dataset", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[99] Lin, T.-Y., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., "Microsoft coco: Common objects in context", in European Conference on Computer Vision, 2014.

[100] Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., Zisserman, A., "The pascal visual object classes (VOC) challenge", International Journal of Computer Vision, Vol. 88, No. 2, 2009, pp. 303–338.

[101] Cordts, M., Omran, M., Ramos, S., Scharwächter, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., "The cityscapes dataset", in CVPR Workshop on The Future of Datasets in Vision, 2015.

[102] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., "Focal loss for dense object detection", in Proceedings of the IEEE international conference on computer vision, 2017.

[103] Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C. C., Lin, D., "MMDetection: Open mmlab detection toolbox and benchmark", arXiv preprint arXiv:1906.07155, 2019.

Biography

Ana Milas received her BSc and MSc degree in electrical engineering and information technology from the University of Zagreb, Faculty of Electrical Engineering and Computing (UNIZG-FER) in 2016 and 2018, respectively. During her master program, she was awarded a Rector's Award for Individual Scientific Work with the project entitled "Coordinated multi-robot exploration based on Graph SLAM method and rapidly exploring random trees". Alongside her studies, she gained industrial working experience in student internships at Inetec d.o.o and at AVL-AST d.o.o. After graduation, she joined LARICS (Laboratory for Robotics and Intelligent Control Systems) as a research assistant at the Department of Control and Computer Engineering (ZARI) at FER. She has worked on several international and domestic scientific projects. As a part of the ERL 2019 (European Robotics League), she worked on the development of an algorithm for simultaneous localization and mapping. She participated in the MBZIRC2020 (Mohamed Bin Zayed International Robotics Challenge), working on a task related to autonomous exploration od the environment and firefigting. She also participated in MBZIRC2023, where she lead the autonomous search team. In the HEKTOR (Heterogeneous autonomous robotic system in viticulture and mariculture), she developed the algorithm for autonomous exploration and mapping of the vineyard while in the VIRTUALUAV (Development of a system of unmanned aerial vehicles (UAVs) controlled in virtual environments) she worked on autonomous exploration of warehouses, boxes and shelves detection and labeling in the map. In 2019, she did an internship at the University of Seville, Spain, where she collaborated on work related to mobile and aerial robot collaboration as a part of the AeRoTwin project. In 2020, she attended the IEEE RAS Summer School on Multi-Robot Systems in Prague, Czech Republic. In 2023, as a part of the program Outgoing mobility of assistants of the Croatian Science Foundation (HRZZ), she participated in a research visit at CATEC (Advanced Center for Aerospace Technologies) on the project BEEYONDERS, where she developed an algorithm for autonomous exploration of the environment, construction site-like object detection and mapping. She completed her doctoral studies within the "Young researchers' career development project–training of doctoral students" of the Croatian Science Foundation. Her research interests are unmanned

aerial vehicles, autonomous systems, autonomous 2D and 3D mapping and exploration and autonomous navigation with obstacle avoidance in 3D environments. She is an active member of IEEE Croatia section since her master's study. Currently, she is Chair of Croatia Young Professionals and IEEE Croatia Section Treasurer. She is an author or co-author of 4 papers published in peer-reviewed journals, 6 papers presented at international conferences and a book chapter. The full list of publications is given below.

Full List of Publications

Book chapters

1. Arbanas, B., Petric, F., Batinović, A., Polić, M., Vatavuk, I., Marković, L., Car, M., Hrabar, I., Ivanović, A., and Bogdan, S. (2021). From ERL to MBZIRC: Development of An Aerial-Ground Robotic Team for Search and Rescue. In Automation and Control [Working Title]. IntechOpen.

Journal papers

1. Batinović, A., Petrović, T., Ivanović, A., Petric, F., and Bogdan, S. (2021). A Multi-Resolution Frontier-Based Planner for Autonomous 3D Exploration. IEEE Robotics and Automation Letters, 6(3), 4528–4535.
2. Batinović, A., Ivanović, A., Petrović, T., and Bogdan, S. (2022). A Shadowcasting-Based Next-Best-View Planner for Autonomous 3D Exploration. IEEE Robotics and Automation Letters, 7(2), 2969–2976.
3. Goricanec, J., Milas, A., Markovic, L., and Bogdan, S. (2023). Collision-Free Trajectory Following With Augmented Artificial Potential Field Using UAVs. IEEE Access, 11, 83492-83506.
4. Milas, A., Ivanovic, A., and Petrovic, T. (2023). ASEP: An Autonomous Semantic Exploration Planner With Object Labeling. IEEE Access, 11, 107169-107183.

Conference papers

1. Batinović, A., Oršulić, J., Petrović, T., and Bogdan, S. (2020). Decentralized Strategy for Cooperative Multi-Robot Exploration and Mapping. IFAC-PapersOnLine, 53(2), 9682-9687.
2. Oršulić, J., Milijaš, R., Batinović, A., Marković, L., Ivanović, A., and Bogdan, S. (2021). Flying with Cartographer: Adapting the Cartographer 3D Graph SLAM Stack for UAV

Navigation. 2021 Aerial Robotic Systems Physically Interacting with the Environment (AIRPHARO), 1–7.

3. Batinovic, A., Goricanec, J., Markovic, L., and Bogdan, S. (2022). Path Planning with Potential Field-Based Obstacle Avoidance in a 3D Environment by an Unmanned Aerial Vehicle. International Conference on Unmanned Aircraft Systems (ICUAS), 394-401.
4. Peti, M., Milas, A., Kraševac, N., Križmančić, M. Lončar, I., Mišković, N., and Bogdan, S. (2023). A Search Strategy and Vessel Detection in Maritime Environment Using Fixed-Wing UAVs. IEEE Underwater Technology (UT), 1-8.
5. Changoluisa Caiza I. D., Milas A., Montes Grova M. A., Perez-Grau F. J., Petrovic T. (2024). Autonomous Exploration of Unknown 3D Environments Using a Frontier-Based Collector Strategy. IEEE International Conference on Robotics and Automation (ICRA), Accepted for publication.
6. Zorić F., Milas A., Petrović T., Kovačić Z., Orsag M. (2024). AI-Enhanced Structural Health Monitoring with a Multi-Rotor Aerial Vehicle. International Conference on Unmanned Aircraft Systems (ICUAS), Accepted for publication.

Životopis

Ana Milas stekla je zvanje prvostupnice, odnosno magistre elektrotehnike i informacijske tehnologije Sveučilišta u Zagrebu, Fakulteta elektrotehnike i računarstva (UNIZG-FER) 2016. te 2018. godine. Kao studentica diplomskog studija nagrađena je Rektorovom nagradom za individualni rad pod naslovom "Koordinirano višerobotsko istraživanje prostora zasnovano na Graph SLAM metodi i brzorastućim slučajnim stablima". Uz studij je stekla radno iskustvo u industriji kroz studentske prakse u tvrtkama Inetec d.o.o i AVL-AST d.o.o. Nakon završetka studija, zaposlila se u Laboratoriju za robotiku i inteligentne sustave upravljanja (LARICS) kao asistentica na Zavodu za automatiku i računalno inženjerstvo (ZARI) na FER-u. Sudjelovala je na više međunarodnih i domaćih znanstvenih projekata. U sklopu ERL 2019 (European Robotics Leaguea) radila je na razvoju algoritma za istovremenu lokalizaciju i kartiranje prostora. Sudjelovala je na MBZIRC2020 (Mohamed Bin Zayed International Robotics Challenge) natjecanju, radeći na zadatku autonomnog pretraživanja prostora i gašenja požara. Također, sudjelovala je i na MBZIRC2023, gdje je vodila tim za autonomno pretraživanje prostora. Na projektu HEKTOR (Heterogeneous autonomous robotic system in viticulture and mariculture) razvila je algoritma za autonomno istraživanje i kartiranje vinograda, dok je na domaćem projektu VIRTUALUAV (Development of a system of unmanned aerial vehicles (UAVs) controlled in virtual environments) radila na autonomnom istraživanju industrijskog skladišta, detekciji kutija i polica u skladištu i označavanju istih u karti prostora. Godine 2019. boravila je kao istraživačica na Sveučilištu Sevilla, Španjolska, gdje je surađivala na radu vezanom za kooperaciju mobilnog robota i bespilotne letjelice sklopu projekta AeRoTwin. Godine 2020. sudjelovala je u IEEE RAS ljetnoj školi za višerobotske sustave u Pragu, Češka Republika. 2023. godine, u sklopu potprograma Odlazna mobilnost asistenata Hrvatske zaklade za znanost (HRZZ) sudjelovala je u znanstveno-istraživačkom posjetu na CATEC (Advanced Center for Aerospace Technologies), kao istraživačica na projektu BEEYONDERS, gdje je razvijala algoritam za autonomno istraživanje prostora i detekciju i kartiranje objekata na gradilištu. Doktorski studij odradila je unutar Programa razvoja karijera mladih istraživača HRZZ-a. Njezine interesne sfere su bespilotne letjelice, autonomni sustavi, autonomno kartiranje i istraživanje 2D i 3D

prostora te autonomna navigacija uz izbjegavanje prepreka u 3D prostoru. Aktivna je članica Hrvatske sekcije IEEE od diplomskog studija. Trenutno je predsjednica interesne skupine Young Professionals Hrvatske sekcije IEEE i blagajnica Hrvatske sekcije IEEE. Autorica je ili koautorica 4 rada objavljena u recenziranim časopisima, 6 radova prezentiranih na međunarodnim konferencijama i jednog poglavlja knjige.