Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi

Virtual Reality for Robotics

Report

Long-term patrolling and relay positioning

Advisor: Prof. Gianni Viardo Vercelli

UNIVERSITY OF GENOA, FEBRUARY 2023

Member list & Workload

No.	Full name	Student ID	Percentage of work
1	Matteo Maragliano	4636216	25%
2	Mattia Musumeci	4670261	25%
3	Davide Leo Parisi	4329668	25%
4	Sara Sgambato	4648592	25%

Contents

1 Introduction **4**

2 State of Art (max 1 page) **5**
2.1 Introduction 5
2.2 UAVs for Patrolling Difficult Areas 5
2.3 Sharing Information Among UAV Swarms 5
2.4 Pros and Cons of Using UAVs vs. Ground Robots 6
2.5 Conclusion 6

3 Tools **7**
3.1 Softwares used 7
3.2 Git-Hub repository 7

4 Description **8**
4.1 PID Controller 8
4.2 Graph Loader 8
4.3 Drone Coverage 9
4.4 Relay Node 9

5 Conclusions & Results **10**
5.1 Improvements 10

1 Introduction

Our project addresses the challenge of conducting periodic surveillance in areas that are difficult to traverse, particularly in emergency situations such as post-landslides or post-avalanches, where search and rescue operations need to be performed as quickly and quickly and effectively as possible.

To tackle this challenge, we propose using a team of Unmanned Aerial Vehicles (UAVs) for aerial observation of the area. The use of UAVs offers several benefits, including increased efficiency in terms of the time required to cover the entire area and the ability to access otherwise difficult-to-reach locations.
To ensure seamless communication and coordination between the UAVs, we employ a relay chaining approach where the UAVs are arranged in such a way that they maintain a connection with the base. This allows for real-time monitoring and control of the UAVs, improving the efficiency of the reconnaissance mission.

In our solution, we focus on emergency situations in the mountains, where the goal is to transmit the information as quickly as possible from the top to the bottom of the mountain. The control of the drones in the system is based on the Robot Operating System (ROS) framework, which provides a powerful and flexible architecture for building robotic systems. The ROS nodes used for drone control are specifically designed to implement the coverage algorithm and relay chain system, as described in detail in section (4) of our project documentation.

Our project aims to develop a method for efficient and effective reconnaissance in emergency situations in areas with difficult-to-traverse terrain, using a combination of UAVs and relay chaining.

2 State of Art (max 1 page)

2.1 Introduction

Unmanned aerial vehicles (UAVs), or drones, have become increasingly popular in recent years due to their ability to operate in difficult or hazardous environments. One promising application of UAVs is in the area of patrolling, where they can be used to monitor large or remote areas for security, surveillance, or environmental purposes.
One challenge in using UAVs for patrolling is the need to efficiently cover a large area while avoiding obstacles and maintaining communication with a central command. One approach to addressing this challenge is to use a swarm of UAVs, where each UAV is able to communicate with the others and share information about their surroundings.

In this state of the art, we will review the current state of the art in the use of UAVs for patrolling difficult areas and sharing information among UAVs in swarm configuration. We will also discuss the pros and cons of using UAVs for this purpose, as compared to ground robots.

2.2 UAVs for Patrolling Difficult Areas

UAVs have several advantages over ground robots for patrolling difficult areas:

1. UAVs are able to fly over obstacles and terrain that may be impassable for ground robots. This allows them to cover a larger area more efficiently, as they do not need to navigate around obstacles or climb over rough terrain.

2. UAVs have a higher degree of mobility than ground robots, as they are able to move in three dimensions rather than just two. This allows them to access areas that may be difficult or impossible to reach by ground, such as the tops of buildings or cliffs.

3. UAVs are generally more lightweight and portable than ground robots, making them easier to deploy and transport to different locations.

There are also some disadvantages to using UAVs for patrolling. One major limitation is the limited endurance of most UAVs, which limits the amount of time they can spend in the air before needing to land or recharge. Another limitation is the vulnerability of UAVs to interference or jamming of their communication and navigation systems.

2.3 Sharing Information Among UAV Swarms

One way to overcome the limitations of individual UAVs is to use a swarm of UAVs working together. In a UAV swarm, each individual UAV is able to communicate with the others and share information about its surroundings. This allows the swarm to make more informed decisions about how to cover the area being patrolled, and to adapt to changing conditions or obstacles.

There are several approaches to sharing information among UAV swarms. One approach is to use a decentralized control system, where each UAV makes its own decisions based on local information and the actions of its neighbors. This can be effective for simple tasks such as coverage or flocking, but may be less robust for more complex tasks that require coordination or planning.
Another approach is to use a centralized control system, where a central command unit makes decisions for the entire swarm based on global information. This can be more efficient for tasks that require precise coordination or planning, but may be more vulnerable to interference or failure of the central command unit.

2.4 Pros and Cons of Using UAVs vs. Ground Robots

As we have seen, there are both advantages and disadvantages to using UAVs for patrolling difficult areas and sharing information among UAV swarms. Some of the main pros and cons are summarized in the table below.

	UAVs	**Ground Robots**
Mobility	Can move in three dimensions	Limited to two dimensions
Obstacle Avoidance	Can fly over obstacles	Must navigate around obstacles
Endurance	Limited endurance	Can operate for longer periods
Communication	Vulnerable to interference	Less vulnerable to interference
Portability	Lightweight and portable	Generally heavier and less portable

Overall, UAVs have the advantage of greater mobility and the ability to cover a larger area more efficiently. However, they are limited by their limited endurance and vulnerability to interference. Ground robots, on the other hand, have the advantage of longer endurance and less vulnerability to interference, but are limited in their mobility and ability to cover a large area efficiently.

2.5 Conclusion

In summary, UAVs have the potential to be a powerful tool for patrolling difficult areas and sharing information among UAV swarms. However, they also have some limitations, such as limited endurance and vulnerability to interference. Choosing between UAVs and ground robots will depend on the specific needs and constraints of the task at hand.

3 Tools

3.1 Softwares used

Here there is the list of all the tools used in the project:

WSL : Windows Subsystem for Linux (v. 1.0.3.0)
It is a compatibility layer that allows users to run Linux applications on Windows 10. It provides a Linux-compatible environment for running native Linux command-line tools, utilities, and applications directly on Windows, without the need for a virtual machine. WSL provides a more seamless integration of Linux and Windows, allowing developers to work with both systems without the need to switch between them.

ROS : Robot Operating System (v. Noetic)
It is an open-source framework for building robot applications. It provides libraries and tools for creating, developing, and deploying robots, as well as a standard set of messaging and communication protocols for exchanging data between different components of a robot system. ROS is used by many researchers and engineers around the world to build complex robotic systems, as it provides a robust and scalable platform for developing advanced robotics applications.

UE : Unreal Engine (v. 4.27.2)
It is a game engine and development platform for creating high-quality video games, architectural visualizations, and other interactive 3D experiences. It provides a suite of tools for 3D modelling, animation, and simulation, as well as a powerful scripting system for creating interactive environments. Unreal Engine has been used to create many successful games and interactive experiences, and its popularity has led to its use in a wide range of industries, including architecture, film, and engineering.

AirSim : (v. 1.8.1)
It is an open-source simulation platform for testing autonomous systems, specifically drones, in a virtual environment. It provides a high-fidelity simulation environment that accurately models the physics and dynamics of aerial vehicles, as well as a variety of sensor models for testing perception and control algorithms. AirSim is built on the Unreal Engine, providing a powerful and flexible platform for developing and testing autonomous systems in a safe and controlled environment.

- **AirSim ROS Wrapper**: (v. 1.8.1)
  It is a software package that provides a bridge between the AirSim simulation platform and the Robot Operating System (ROS). It allows users to interface AirSim with ROS, enabling the use of ROS tools and libraries for developing and testing autonomous systems in a simulated environment. The AirSim ROS Wrapper provides a convenient and flexible way to integrate AirSim simulations with the broader ROS ecosystem, making it easier to build and test advanced robotics applications.

3.2 Git-Hub repository

Git-Hub repository: https://github.com/mmatteo-hub/VR4R_Assignment.

4 Description

We developed a **multi-UAV** simulation using a build of the AirSim simulator for Unreal Engine 4.27.
The drones inside the simulation are managed by ROS nodes thanks to the **airsim_ros_pkgs** wrapper which provides a very simple-to-use API for the communication and control of the drones through the ROS ecosystem.
Our project is composed of the following ROS nodes, which provide a basic software architecture for developing a more complex **coverage algorithm** and **relay chain system**. The connections between these nodes are defined with the definition of the functionality of each node.

4.1 PID Controller

We developed a simple **PID Controller** ROS node in order to control, singularly and concurrently, the many drones inside the simulation. This implies that an instance of the PID controller is required for each drone. This node uses a **/local_goal** subscriber to receive new target positions for the drone. Then, the PID controller automatically moves the drone to the target position using the drone odometry. This behaviour is only possible thanks to the **/local_vel_cmd_world_frame** topic provided by the AirSim ROS Wrapper. In fact, this velocity is then transposed into the local velocity of the drone.

4.2 Graph Loader

Our architecture is developed on the basis of a graph in which the nodes are tridimensional positions in which the drones might move. We suppose that this graph is given in a JSON file with the following format:

- A "nodes" map which maps names into tri-dimensional arrays containing [x,y,z] coordinates.
- A "connections" array contains bi-dimensional arrays containing [name0,name1] connections.

Therefore, the positions are identified by names which are used to create connections.
We developed a **Graph Loader** ROS node which provides a service for loading a graph from a JSON file using the just-defined structure. Then, this node provides another service which requires the name of the starting position and the name of the goal position in order to compute the path between the two nodes. Once this is successfully completed, the path is published on the publishing on the **/compute_path** topic.
Internally, the Graph Loader uses an implementation of the A* algorithm for obtaining the optimal path between the two requested nodes.
This implementation provides a simple way for selecting different paths inside the graph of positions, which will later be used by the Drone Coverage node to move the drones into the related positions.

4.3 Drone Coverage

The **Drone Coverage** ROS node contains the names of all the available drones in the system and listens on the path computed and published by the Graph Loader node on the **/compute_path** topic. When a new path is received, it assigns a drone to each node in the path and communicates to the corresponding PID Controller in order to move the drone to the correct position. This is done through the **/local_goal** topic.

4.4 Relay Node

Supposing that, when performing the coverage procedure, two successive drones are able to communicate with each other, each drone is considered a node inside a **relay chain** which can be used to transmit information from the first drone to the last drone (and vice versa) in the chain which is not alone able to communicate with each other. To obtain this behaviour, we implemented a **Relay Node** ROS node which uses the following topics:

- The **/relay_chain/self/forward** topic is used for receiving information from the previous drone in the chain which needs to be transmitted to the next drone in the chain. This is done by publishing the information on the **/relay_chain/next/forward** topic related to the next drone.
- The **/relay_chain/self/backward** topic is used for receiving information from the next drone in the chain which needs to be transmitted to the previous drone in the chain. This is done by publishing the information on the **/relay_chain/prev/backward** topic related to the previous drone.

5 Conclusions & Results

In this project, Unreal Engine and AriSim were used to simulate the behaviour of multiple drones that share information among themselves. The localization of the drones was performed using GPS coordinates published by AirSim, which provided longitude, latitude, and height. However, the use of geographical coordinates poses a challenge, as they need to be converted into Euclidean coordinates for the purpose of odometry. This conversion can result in inaccuracies, especially when the relative position of the drone to the world frame is not known.

The relay positioning algorithm used in this project was A*, which was specified by the professor. However, it would be beneficial to test the simulation using a more sophisticated and realistic relay positioning algorithm in order to provide a more accurate representation of the communication between drones. This would allow for a better understanding of the strengths and limitations of different algorithms and help to improve the overall performance of the drone communication system.

In conclusion, the localization of drones in the simulation project could be improved by considering alternative methods. Integrating additional sensors, using visual odometry or simultaneous localization and mapping (SLAM) algorithms, or creating a map of the environment for accurate position estimation are all potential solutions that could be explored in future works. Furthermore, testing the simulation using a more sophisticated and realistic relay positioning algorithm would provide valuable insights into the performance of the drone communication system.

As possible implementations for future works, the following could be considered:

- Integration of additional sensors such as cameras or lidars to provide more accurate position estimation and overcome limitations of GPS-based localization.
- Use of visual odometry or SLAM algorithms to estimate the position of the drone relative to the environment, even in cases where GPS signals are weak or unavailable.
- Development of a decentralized communication system that allows drones to exchange information and coordinate their actions in real-time, without relying on a central controller.

These future works would help to overcome the limitations of the current simulation and provide valuable insights into the potential of drone systems for a wide range of applications.

5.1 Improvements

To improve our solution and make it useful, there could be some improvements, such as transmitting information through a relay of drones controlled in Virtual Reality (VR) through a First Person View (FPV) headset. This would enable the operator to have an immersive and intuitive experience while controlling the drones, allowing for quicker and more effective transmission of information.
Additionally, the VR-FPV interface should also provide the ability to control each drone individually, moving

it within its range of action. This would enable the operator to monitor the state of the mountain in real-time, making it possible to identify and respond to potential hazards quickly.

References

[1] Donald E. Knuth. Literate programming. *The Computer Journal*, 27(2):97–111, 1984.

[2] Donald E. Knuth. *The TEX Book*. Addison-Wesley Professional, 1986.

[3] Leslie Lamport. *LATEX: a Document Preparation System*. Addison Wesley, Massachusetts, 2 edition, 1994.

[4] Michael Lesk and Brian Kernighan. Computer typesetting of technical journals on UNIX. In *Proceedings of American Federation of Information Processing Societies: 1977 National Computer Conference*, pages 879–888, Dallas, Texas, 1977.

[5] Frank Mittelbach, Michel Gossens, Johannes Braams, David Carlisle, and Chris Rowley. *The LATEX Companion*. Addison-Wesley Professional, 2 edition, 2004.