

A G H

AGH University of Krakow

FACULTY OF COMPUTER SCIENCE, ELECTRONICS AND TELECOMMUNICATIONS

Master Thesis

Machine Learning Path Planning for a Solar-Powered Unmanned Aerial Vehicle considering Area Coverage and Scene Parsing for Disaster Recovery

Planowanie ścieżki bezzałogowego statku powietrznego zasilanego energią słoneczną, z wykorzystaniem uczenia maszynowego z uwzględnieniem pokrycia obszaru i analizowania scen na potrzeby usuwania skutków katastrof

Author: *Filip Pieniążek*
Degree programme: *Information and Communication Technologies*
Supervisor: *Andres Vejar, PhD. Eng.*

Kraków, 2024

Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystycznego wykonania albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, videogram lub nadanie.”, a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.): „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej «sądem koleżeńskim».”, oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

I would like to express my sincere gratitude to my thesis supervisor for his dedication and valuable guidance during the writing of this thesis.

I would also like to thank the members of AGH Solar Plane for the discussions we had together, exchanging knowledge and helping me develop my interests. Your passion for science and cooperation have significantly influenced my development. Special thanks go to Szczepan Malaga, whose emotional support and practical help have been invaluable to me in difficult moments. Thank you for always being able to count on you.

Once again, thank you to everyone who contributed to this thesis!

Abstract

Solar-Powered Unmanned Aerial Vehicles (SPUAVs) are successfully tested for monitoring large geographical areas, given their extended fly autonomy, configuration versatility, and relatively low cost. A critical area of application is to provide early support in disaster recovery. The SPUAV needs to monitor the landscape and parse the scenes, inspecting for a predefined set of targets: e.g., persons, homes, flooding, avalanches, fires, and roads. This master thesis's goal is to develop a machine learning model that plans the path for the SPUAV to cover a selected area, parse the scenes captured and return to the base before the batteries are exhausted. The machine learning model will be deployed in one prototype device of the research and development group "AGH Solar Plane" and tested under real operating conditions.

Abstract

Bezzałogowe statki powietrzne zasilane energią słoneczną (ang. solar-powered unmanned aerial vehicle, SPUAV) są z powodzeniem testowane pod kątem monitorowania dużych obszarów geograficznych, biorąc pod uwagę ich rozszerzoną autonomię lotu, wszechstronność konfiguracji oraz stosunkowo niski koszt. Krytycznym obszarem zastosowania jest zapewnienie wczesnego wsparcia w usuwaniu skutków katastrof. SPUAV musi monitorować krajobraz i analizować sceny, sprawdzając wstępnie zdefiniowany zestaw celów: np. osoby, domy, powodzie, lawiny, pożary i drogi. Celem tej pracy magisterskiej jest opracowanie modelu uczenia maszynowego, który planuje ścieżkę dla SPUAV w celu pokrycia wybranego obszaru, przeanalizowania przechwyconych scen i powrotu do bazy przed wyczerpaniem baterii. Model uczenia maszynowego zostanie wdrożony w jednym prototypowym urządzeniu koła naukowego "AGH Solar Plane" i przetestowany w rzeczywistych warunkach pracy.

Contents

1. Introduction	5
1.1. Motivation	5
1.2. Case study description.....	5
2. Theory	7
2.1. Introduction to machine learning	7
2.2. Traveling salesman algorithm	8
2.3. Mavlink protocol	9
2.4. Ardupilot and mission planner software	9
2.5. Comparison between fixed wing and rotary UAV	10
2.6. Aviation law in Poland	12
2.7. System components.....	13
3. Software implementation	17
3.1. Software components	17
3.2. Machine learning model implementation.....	19
3.3. Processing data to Mavlink protocol	19
4. Hardware implementation	21
4.1. Raspberry Pi - Main computing unit	21
4.2. Pixhawk Cube flight controller	22
4.3. Unmanned Aerial Vehicle electronics	23
4.3.1. Electric motor.....	23
4.3.2. Battery.....	24
4.3.3. Servos.....	24

4.3.4. Pitot tube	25
4.3.5. Solar cells.....	26
5. Design of Unmanned Aerial Vehicle.....	29
5.1. Project design in CAD software	29
5.2. Build process of Unmanned Aerial Vehicle	30
5.2.1. Wings build process	30
5.2.2. Stabilizers build process	31
5.2.3. Fuselage build process	31
5.2.4. Camera gimbal build process.....	32
6. Results	33
6.1. Prepare data for case study	33
6.2. Results collected on case study	34
6.3. Analysis and processing of logs	39
6.4. Developed aircraft	41
7. Conclusions and development opportunities.....	43
7.1. Machine learning.....	43
7.2. Plane launcher	43
7.3. Antenna tracker	45

1. Introduction

1.1. Motivation

Unmanned aerial vehicles are becoming an increasingly common industry with the development of modern technology. Rapid development in the field of communications, power sources or propulsion sources allows to expand the range of applications of these devices. Previously used mainly for hobby purposes or as photo-video devices, they are increasingly being used in industry as inspection drones, among other things. They allow the automation of daily activities performed by people, thus reducing costs as well as risks in certain harsh working conditions. Unmanned aerial vehicles are also being used as medical transport in countries with underdeveloped road infrastructure. Drones, due to their characteristics, make it possible to get into terrain that seems inaccessible to any other service under the given conditions. It is for this reason that UAVs can be used as reconnaissance units in hard-to-reach terrain such as those affected by environmental disasters. They can also be used as patrol or rapid response units in fire-prone areas due to high temperatures and low humidity. Due to the limited battery capacity of such devices, route planning algorithms are useful for patrolling a given area or points of interest within its boundaries for as long as possible. Thanks to route planning, battery utilization can be optimized, and in addition, with the installation of photovoltaic cells, vehicle operation can be extended.

1.2. Case study description

. This work considers the scenario of a possible natural disaster in an inaccessible area. The terrain may be affected by fire, flood or, for example, the aftermath of an earthquake thereby preventing easy access for emergency services. In such situations, the first important aspect to

consider is to determine the extent of the threat present in the area. This objective can be facilitated by an unmanned aircraft with a mounted camera as presented on Figure 1.1 transmitting up-to-date geolocation data to a base station located at a considerable distance. During the first

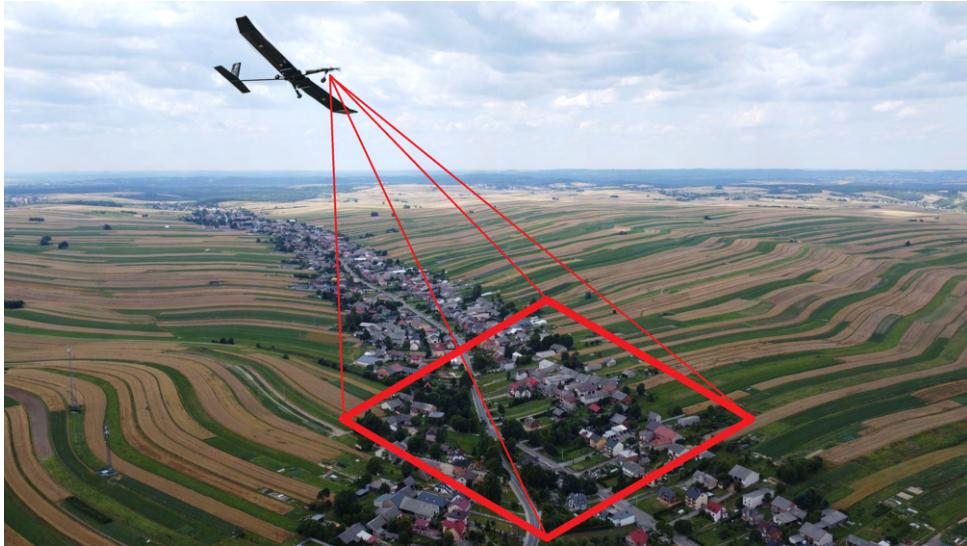


Figure 1.1. Case study visualisation

reconnaissance flight, the aircraft flies over a given area to locate potential points of interest for further investigation, in this step it is at a higher altitude above the ground in order to cover a larger area of the terrain with the camera angle. Once the points of interest have been established during subsequent overflights, the aircraft can already move at a lower altitude within the pre-determined areas thereby collecting more accurate information. Route planning algorithms will be used to optimize the patrolling time of a specific area.

2. Theory

2.1. Introduction to machine learning

Machine Learning (ML) is a field of artificial intelligence in which computer systems learn from data rather than being directly programmed. ML algorithms analyze large data sets, recognize patterns and make decisions or predict outcomes based on that. In the topics covered in this project, machine learning could be used for several purposes:

- forecasting weather conditions to avoid bad conditions in real time based on historical data collected from the internet
- obstacle avoidance through image analysis and the use of additional sensors
- energy optimization to increase flight range
- adapting the route to dynamic changes forced by changes in flight zones or current air traffic in the area

The traveling salesman algorithm described in the following section was used in the project, which can also be optimized through the use of machine learning in the following ways:

- supervised learning - used to train the model on historical routing data. Such a model can learn the optimal route for a specific configuration of points which allows it to later pre-generate close to optimal routes
- genetic algorithms - can be assisted by ML to find better solutions faster by dynamically adjusting algorithm parameters

- point clustering - the algorithm can reduce the amount of computation by initially dividing the problem into smaller problems(geographic clusters of similar points), which will be much easier to compute
- reinforcement learning - allows to teach the traveling salesman algorithm to search the space faster and to preemptively skip inefficient paths based on previous experience

The optimal solution seems to be a hybrid model that is a combination of traditional optimization methods and machine learning that allows to quickly find initial solutions close to the optimal one.

2.2. Traveling salesman algorithm

The traveling salesman algorithm is a solution to the classical optimization problem known as the traveling salesman problem. It consists in finding a route passing through all designated points visiting each of them only once and returning to the starting point. This problem is especially complicated for larger instances where the computation time increases significantly and there is no algorithm that can always compute the problem in a short time.

Definition of the problem:

- input - a set of points and distances between them, most often given as a distance matrix
- task - to find the shortest path passing through each point exactly once, returning to the starting point

This problem appears in many practical applications, such as:

- planning delivery routes
- logistics
- integrated circuit design
- the problem of drones delivering packages

2.3. Mavlink protocol

MAVLink (Micro Air Vehicle Link) is a lightweight and powerful message-based communication protocol specifically designed for use with unmanned vehicles (UAVs) such as drones. It is commonly used in autopilot systems such as ArduPilot and PX4, as well as in various ground applications and peripherals. [1] Main features of the MAVLink protocol:

- **Lightweight:** MAVLink is designed to minimize communication overhead, which is crucial for applications in resource-constrained systems such as drones.
- **Support for multiple message types:** The protocol supports a variety of message types, including those related to system status, position, speed, mission control, and telemetry.
- **Message structure:** MAVLink messages have a defined structure that includes a header, payload and checksum to ensure data integrity.
- **Support for multiple communication channels:** MAVLink can operate on a variety of communication media, such as UART, USB, telemetry radio and IP networks (WiFi, Ethernet).
- **Support for redundant communications:** The protocol allows redundancy to be configured, which increases the reliability of the communication system.

2.4. Ardupilot and mission planner software

Ardupilot is a software that allows the creation of autonomous unmanned vehicles. It provides a large number of tools allowing to configure both driving, floating and flying vehicles in any way. Through its possible integration with multiple flight controllers, the system is one of the most popular software used in unmanned autonomous and remote-controlled vehicles. The software allows configuring the vehicle's parameters depending on its physical parameters and possible use. Among other things, it allows direct real-time communication with the vehicle, reading all its current parameters and adjusting them in real time. This information is also saved in the form of log files that allow analysis of the vehicle's behavior. The Ardupilot software also allows autonomous modes of vehicle movement, which can be activated remotely on the vehicle

in advance or in real time. A route planning tool allows the user to place points on a map, set parameters such as speed, altitude or turning radius, for example, and then calculate a possible route for the vehicle to move. [2]

2.5. Comparison between fixed wing and rotary UAV

Unmanned aerial vehicles (UAVs) are mainly divided into two types: fixed-wing and rotary-wing. Each of these types has its own unique design features, applications and advantages. The comparison can be divided into several aspects:

Construction

1. Fixed-wing UAVs:

- **Design:** Have rigid wings that generate lift as the UAV moves forward
- **Engines:** Usually equipped with one or more piston, jet or electric motors that drive the propeller. Smaller designs usually use either a brushless electric drive moving the propeller or an EDF turbine drive.
- **Wing design:** wings are usually long and narrow, which promotes greater aerodynamic efficiency thus allowing gliding without the need for an engine(Figure 2.1).



Figure 2.1. Fixed wing UAV

2. Rotary-wing UAVs:

- **Design:** They have a main rotor (similar to a helicopter) that generates both lift and propulsion.

- **Motors:** Mostly equipped with electric motors that drive the rotor.
- **Rotor design:** Rotors can be single-rotor or multi-rotor (for example, quadcopters have four rotors as shown on Figure 2.2). The most popular designs are those with 4,6 or 8 rotors. A larger number of motors increases the possible payload of the vehicle thereby reducing its operating time. The number of rotors should be adjusted according to the planned application.



Figure 2.2. Rotary wing UAV [3]

Application

1. Fixed-wing UAVs:

- **Military applications:** Long-range reconnaissance missions, surveillance, aerial attacks.
- **Civilian applications:** Cartography, environmental monitoring, border patrol, long-distance transportation of goods.
- **Commercial applications:** Agriculture (crop monitoring), delivery (e.g., courier shipments).

2. Rotary-wing UAVs:

- **Military applications:** Short-distance reconnaissance missions, logistics support, special operations.
- **Civilian applications:** Aerial photography and filming, infrastructure inspections (e.g., bridges, towers), rescue.
- **Commercial applications:** Short-distance deliveries, asset monitoring, building inspections.

2.6. Aviation law in Poland

All flights by unmanned aircraft, as well as manned flights, must meet the legal requirements described in the Polish Aviation Law. The main legal authority issuing the following permits and supervising air traffic is the Polish Air Navigation Services Agency and the Civil Aviation Authority. Each flight must absolutely comply with the general law and regional restrictions prevailing in the area where the air mission is taking place. Both the pilot of the vehicle responsible for the control and the operator responsible for the organization of the mission and taking responsibility for the mission are required to have the appropriate authorizations. [4]

In the case of current Polish aviation law, there are 3 categories of unmanned flights:

- open category - allows the execution of low-risk flights that do not require prior establishment of the necessary approvals with the previously mentioned state authorities. Operations performed in this category include flying only within the visual range of the operator(VLOS) at a distance above the ground(AGL) of no more than 120m. The weight of the drone must not exceed 25kg. However, this category does not limit whether the flight is performed with pilot control or autonomously. What is required, however, is the ability of the pilot to take control at any time during the flight. The open category is divided into three subcategories: A1, A2, A3, differing by vehicle takeoff weight, among other factors. The A1 and A3 category is the easiest to obtain, completing a free online course is sufficient for this. It allows drones <250g to be flown without flying over bystanders. Many manufacturers of hobby photo-video drones adhere to this restriction thereby allowing most users to fly legally without having to take a paid course [5]
- special category - allows medium-risk flights, including flights over bystanders. In most cases, it requires prior, usually 24 hours before the start of the operation to apply for permission to perform operations in the state PansaUTM system. Flights in this category can be performed both within the pilot's line of sight(VLOS) and beyond it(BVLOS). The maximum weight and flight altitude above the ground as in the open category is limited to 25kg and 120m. It is divided into 8 subcategories NSTS-01 to NSTS-08, where each subcategory specifies a different type of unmanned aircraft and its maximum takeoff

weight. A theoretical and practical course completed by an external state exam is required to obtain the authorization to fly in the special category [6]

- certified category - allows to perform high-risk flights, for which approval is mandatory.
This is an extension of the special category that allows flights over gatherings of people, related to the transportation of people, and related to the transportation of hazardous materials or medical transport

For most hobbyist photo-video flights, the open category is sufficient. In the case of the contemplated monitoring of the affected space, flying out of sight is required thereby already requiring classification as a special category. However, when considering such a scenario, it is worth taking into account the possibility of state entities creating a special air zone in the area, where other restrictions apply or not. If the airspace is closed to outsiders, it is possible to raise the maximum flight ceiling thereby enabling coverage of a larger area with a single aircraft shot. It is also worth noting that at the beginning of this year changes were made to the Polish aviation law abolishing the previous division into 3 categories, and introducing a unified system to be applied throughout the European Union in the future. However, this law is still under construction and many aspects are still unclear due to which Poland is currently in a transition period, which still allows the use of the old aviation privileges.

2.7. System components

The presented system solution can be divided into two main parts:

- software part - responsible for calculating the appropriate route having certain parameters
- hardware part - responsible for acquiring information and for moving the UAV

The software part is implemented using code written in Python run on a Raspberry Pi board. The choice of this microcomputer was dictated mainly by the requirement of small size and weight, due to the installation on the aircraft. Despite its small size, this computer presents surprisingly high computing capabilities sufficient in this case to run the software described later in this chapter dedicated to the software part.

The hardware part includes both the overall construction of the aircraft necessary as an execution platform, the computing units and the individual devices necessary for data collection. The

description of the construction of the aircraft is described in a later section, while as far as the other aspects are concerned, an indispensable part of the project is the camera that allows the collection of images processed at a later stage by the computing units. The selection of a suitable image sensor was made according to the following criteria:

- matrix resolution - it is responsible for the number of pixels contained in one image frame. By selecting the appropriate value, and then making some calculations, it is possible to calculate what area of land corresponds to one pixel of the matrix
- field of view - the value given in degrees of angle, determines how wide the image is covered by the lens. This value varies for different axes. In combination with the resolution parameter, it helps determine the real size of a pixel at a given height. When wide-angle lenses are used, there is distortion in the lateral parts of the image frame thus distorting the results. It is for this reason that lenses with too large a field of view are avoided
- price, availability and compatibility - in this case, the main aspect considered was compatibility with the Raspberry Pi board. All cameras included in the comparison include a ribbon that allows them to plug directly into the camera socket on the board, thus not requiring any converters or unusual libraries in the code used

The target camera comparison is shown on Figure 2.3 and Figure 2.4. The comparison was made by calculating the maximum possible area covered by a single image captured by the arrays at the appropriate height. The maximum altitude for which calculations were made was limited to 120m above ground level(AGL). This value corresponds to the maximum flight altitude of an unmanned aircraft according to Polish aviation law without additional special permits. The minimum value was also chosen as the lowest required for safe flight.

After analyzing the results, the camera chosen was D - Raspberry Pi Camera HD v3 12MPx - wide presenting very good performance without the need to mount a lens with a very large viewing angle, which could already distort the image in the side regions of the frame. For subsequent calculations and testing of the finished software, the parameters of this particular image sensor were used.

A	Raspberry Pi Camera HD v2 8MPx								
B	Sony IMX219 8MPx 160 degree wide lens								
C	Raspberry Pi Camera HD v3 12MPx								
D	Raspberry Pi Camera HD v3 12MPx - wide								
E	Arducam B0240 - IMX477P 12,3MPx HQ 6mm lens								

Figure 2.3. Camera names

	Camera model									
	A		B		C		D		E	
View Angle	62,2	48,8	160	160	66	41	102	67	55	45
Resolution	3280	2464	3281	2464	4608	2592	4608	2592	4056	3040
AGL distance[m]	Covered area[m]									
30	36,19	27,22	340,28	340,28	38,96	22,43	74,09	39,71	31,23	24,85
60	72,39	54,43	680,55	680,55	77,93	44,87	148,19	79,43	62,47	49,71
80	96,52	72,58	907,41	907,41	103,91	59,82	197,58	105,90	83,29	66,27
100	120,65	90,72	1134,26	1134,26	129,88	74,78	246,98	132,38	104,11	82,84
120	144,78	108,87	1361,11	1361,11	155,86	89,73	296,38	158,85	124,94	99,41
AGL distance[m]	Distance per one pixel[cm]									
30	1,10	1,10	10,37	13,81	0,85	0,87	1,61	1,53	0,77	0,82
60	2,21	2,21	20,74	27,62	1,69	1,73	3,22	3,06	1,54	1,64
80	2,94	2,95	27,66	36,83	2,25	2,31	4,29	4,09	2,05	2,18
100	3,68	3,68	34,57	46,03	2,82	2,88	5,36	5,11	2,57	2,73
120	4,41	4,42	41,48	55,24	3,38	3,46	6,43	6,13	3,08	3,27

Figure 2.4. Camera comparison [7]

3. Software implementation

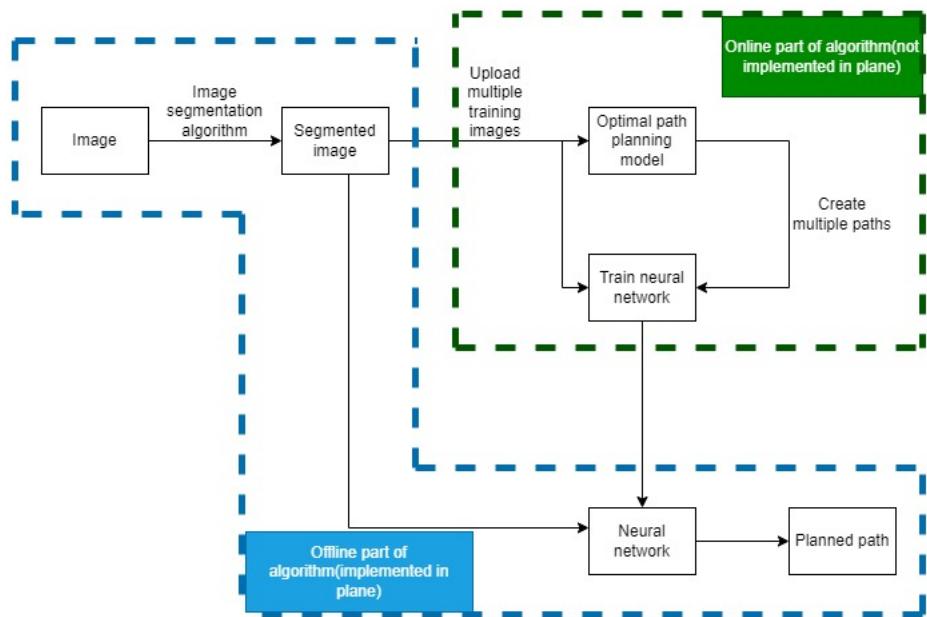


Figure 3.1. Algorithm diagram

3.1. Software components

The programming part of the issue at hand is mainly based on the issue of image processing, and then generating appropriate points with specific geolocation data. Once these values are determined, the data can be sent using the MavLink protocol to the Pixhawk controller. The appropriate configuration of the controller depends on the aircraft model used, the nature of the mission and weather conditions, but these are parameters independent of the image processing script and are a separate part of the implementation. The parameters required to properly calibrate the algorithm that depend on the aircraft configuration are the aircraft's minimum turn

radius and altitude. These are the parameters required to determine the appropriate distance between points of interest and to calculate the route generated by the traveling salesman algorithm.

The software that processes the images collected by the aircraft is divided into the following stages:

- the first stage is to load the image taken from the camera array, and then prepare it and crop it to the appropriate size
- the next stage is to separate appropriate zones in the image, which we define as points of interest. To do this, we need to segment the image, which will allow us to categorize the elements of the image in a certain way or to separate individual points of interest. Due to the limited hardware and computing capabilities of the Raspberry Pi main unit, a simpler algorithm is required to separate such fields. For this purpose, the SLIC(Simple Linear Iterative Clustering) segmentation method was used, which is based on the superpixel clustering algorithm. This segmentation involves dividing the image into an appropriate number of superpixels, specified in the parameters of the algorithm. These are smaller, more homogeneous image elements having the same visual characteristics or texture located in their neighborhood. This makes it easy to isolate image elements that stand out from the rest. In the case of considering disaster scenarios, for example, these could be fire areas segmented from outside the forest area. After segmentation, the segmented areas will have only the forest or the fire within them, then through color analysis you can determine which areas are the ones at risk and create a target point there. After determining the appropriate number of points, which can be, for example, the central part of a given superpixel, you should proceed to calculate the appropriate route between them
- an important aspect addressed in the next step of the algorithm is to determine the geolocation of the individual points determined in the previous step. Knowing the current position of the aircraft, its height and the angle of view of the camera, it is possible to determine the position of the points designated in the image with considerable accuracy. This data will be required to send to the flight controller using the MavLink protocol

- after appropriate preparation of the data on the points of interest, a linear solver is created, which is used to solve the comovement problem. It allows to determine the order of visited points in an optimal way
- after calculating the appropriate order of points and the first shape of the route, it is still necessary to adjust the route, the turns of which must not be perpendicular lines. The nature of aircraft movement and turning requires the use of curves with a predetermined radius at the points of change of flight direction. Thus, the waypoints in the vicinity of the turn must be compacted so that the controller has more accurate information about the planned route. This should be done even before sending geolocation of points
- once the geolocation data of all points are prepared, they can be sent using the MavLink protocol after processing them in advance
- further operation of the aircraft is determined by the appropriate configuration and operation of the flight controller

3.2. Machine learning model implementation

Due to time constraints and the complexity of implementing and training the machine learning model, this part was not implemented in the final version of the project. The creation of the model requires a large dataset to train and adjust the various parameters of the model. The solution presented was based on SLIC segmentation and the traveling salesman algorithm. Computing a larger problem involving the collection of dozens of points would require more computing power or the use of machine learning.

3.3. Processing data to Mavlink protocol

Once the geolocation data of interest has been determined, it should be transformed into a set of three data tuples:

- latitude,
- longitude,

- amount

After preparing the data in this way, we can use the code in Python using the ready-made Py-MAVLink library. Among other things, this library has functions responsible for adding waypoints to the current mission plan. After running the code with properly prepared data, sending it to the Pixhawk controller is done using a UDP connection according to the following scheme presented on Figure 3.2:

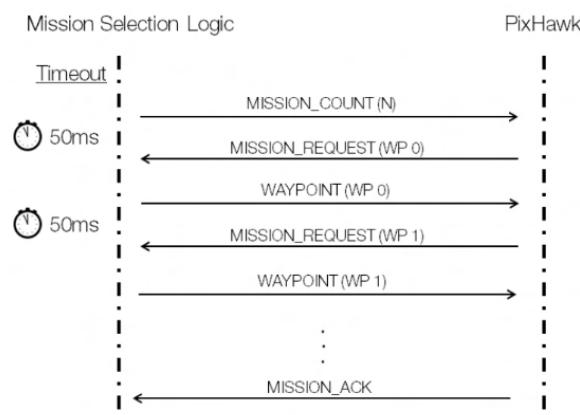


Figure 3.2. Mavlink waypoints update diagram [8]

4. Hardware implementation

4.1. Raspberry Pi - Main computing unit

The main computing unit of the presented solution was the Raspberry Pi 4 minicomputer(Figure 4.1). It is a compact single-board computer with small dimensions and low weight, which is a significant factor when building an aircraft. This minicomputer runs Raspberry Pi OS, which is a version of Ubuntu optimized specifically for use on this device. This computer, despite being equipped with 8GB of RAM and a multi-core processor, is not capable of performing very complex calculations in real time. Therefore, machine learning will be implemented in the system, allowing the model to be trained locally on a more powerful computing unit and then used on the board. The computer has multiple connectors including a port for direct connection of a dedicated RaspiCam.



Figure 4.1. Raspberry Pi 4 computing unit [9]

4.2. Pixhawk Cube flight controller

The Pixhawk flight controller(Figure 4.2) is one of the more popular open-source controllers that allow you to install ArduPilot software in a configuration suitable for your UAV. Its small size and very powerful functionality works well for both simple models requiring only direct transfer of control from the control apparatus to the model, as well as complex systems. It allows the operation of demanding systems where, without the use of external controllers, it is able to process control signals and those from external sensors, adjusting the output signals accordingly to the situation at hand.



Figure 4.2. Pixhawk Cube flight controller [10]

This controller is equipped with multiple modes, each of which is independently configurable and the model can behave completely differently. The possible flight modes can be divided into 3 groups:

- manual - this mode directly transmits signals from the control apparatus to the controller's outputs according to the rules and signal gains specified in the configuration. In its simplest form, the output signal is equal to the input signal
- assisted - these are modes in which the pilot's control is assisted to a certain degree by sensors built into the flight controller. These sensors include an accelerometer, gyroscope, GPS or pitot tube responsible for measuring speed relative to the wind. With the help of this information, these modes allow the aircraft to maintain a horizontal position, maintain an appropriate speed or GPS-related functions such as geofencing that limits the maximum flight area

- autonomous - these are flight modes fully controlled by the controller based on parameters read from sensors and a specific flight path previously uploaded to the controller. These modes also include features such as Return To Home(RTH) or Circle to allow circling in a given location. These modes are also used as failsafe, i.e., procedures that have previously been programmed as fail-safe, e.g., in the event of a broken control signal from the pilot

The controller is equipped with several interfaces(I2C, UART, CAN) allowing for appropriate selection depending on the sensor used. Thanks to the use of several interfaces, it is possible to maintain the redundancy of sensors thus creating a safer executive system.

4.3. Unmanned Aerial Vehicle electronics

The aircraft was equipped with the electronics necessary for the proper operation of the system responsible for calculating the route, the flight controller and other electronics that enable the aircraft to move and operate correctly. To this end, the aircraft was equipped with the components described, later in this chapter.

4.3.1. Electric motor

The aircraft is powered by two engines mounted to the leading edge of the wings. This arrangement of engines allows the camera to be mounted centrally on the front of the model. An alternative option would be to use a single engine in a push rather than pull configuration. Such motor mounting also rules out the problem of air disturbance generated by the propeller being driven. If the camera is positioned behind the motor, there is the possibility of vibration of the camera thereby creating blurred images. Taking into account the planned target weight of the model, two engines were used symmetrically on the wings. The models used(Figure 4.3) allow the model to fly safely up to 6kg, which exceeds its planned weight. For appropriate power control, Electronic Speed Controller(ESC) was used, allowing to regulate the speed of the brushless motor controlled by PWM signal from the Pixhawk controller.



Figure 4.3. Electric motor and electronic speed controller [11]

4.3.2. Battery

A lithium-polymer battery (Figure 4.4) was used because of the high motor discharge current required. This type of power source is commonly used in remote-controlled models, where it is important to keep the battery weight low while maintaining a very high discharge current.



Figure 4.4. Lithium polymer battery [12]

4.3.3. Servos

Servos are responsible for moving the individual control surfaces of the wings and stabilizers, allowing them to maintain the deflection angle set by the controller, thereby controlling the corresponding aileron or flap of the model. The servos are controlled using a PWM signal. The model requires 6 servos with appropriately selected parameters for the surface they control. These parameters are the moment of force and the supply voltage, which must be compatible with the rest of the electrical system found in the aircraft or require a separate step-down converter. The torque of force provided by the servo must allow the control surface to consistently hold a preset position at the specified maximum speed at which the vehicle will travel. If the

value is too low, there is a risk of losing control of the model due to lack of swing of the given control surface. Used KST DS135MG servos are shown on Figure 4.5.



Figure 4.5. Servo [13]

4.3.4. Pitot tube

The pitot tube is a measuring instrument used to calculate the speed of an aircraft relative to the wind. Such a measurement is necessary to determine the necessary power supplied to the engine to achieve the desired airspeed. This sensor is built from a tube equipped with two holes and two pressure gauges (Figure 4.6). With these, with the appropriate processing of the output signals and pressures, the speed of the incoming fluid, in this case air, can be counted. Using only the measurement of velocity relative to the ground according to the GPS can lead to a condition where the aircraft will use too much energy to fly. The second possible scenario is when it reaches too low a speed relative to the wind thereby becoming unsteady, resulting in a possible crash of the aircraft. In order for the vehicle to maintain lift, it is necessary to maintain a speed higher than the stall speed relative to the wind. This means that despite the fact that on the GPS measurement the aircraft will be stationary, in strong winds the aircraft will still maintain flying ability. Or, on the contrary, in the case of a strong tailwind, despite a considerable speed relative to the ground, the aircraft may even have zero speed relative to the wind. The stall speed of an aircraft is defined by the airfoil applied to the supporting surfaces, in this case the wings.

With proper aerodynamic analysis and the use of such a sensor, it is possible to put the minimum required speed for a given model into the aircraft's control code.



Figure 4.6. Pitot tube with essential electronics [14]

4.3.5. Solar cells

Various plans can be used to optimize the operation of the engine and other electronic components to increase the model's uptime. However, this is only one way and severely limited in severe weather conditions, when the aircraft is often exposed to sudden maneuvers. Another way is to equip the model with photovoltaic cells that allow it to harvest solar energy to recharge its batteries. The surface and aerodynamic profile of the wing are designed to accommodate such cells. The main limitation in this case is the maximum deflection of a single cell, which with curves of smaller radius can break. The cells used are laminated in a special shrink film and fiberglass sheathing. Thanks to this treatment, it is possible to achieve greater deflection while maintaining the parameters of the cell. Sample laminated solar cells are shown on Figure 4.7. The voltage generated with the panels is transmitted to the MPPT(Maximum Power Point Tracking)(Figure 4.8). This device is responsible for intelligently adjusting the battery's charging current and voltage based on the possible power transmitted by the solar cells at any given time.



Figure 4.7. Laminated solar cells

ing)(Figure 4.8). This device is responsible for intelligently adjusting the battery's charging current and voltage based on the possible power transmitted by the solar cells at any given time.

MPPT must also be matched to the number of cells on the wing, as well as the voltage rating of the battery. Each has its own operating range within which it can charge the battery. MPPTs are divided into those operating in STEP-UP, STEP-DOWN and hybrid ranges. The choice of inverter type determines the operating voltage range. STEP-DOWN inverters, in which the voltage is stepped down to match the battery voltage, have the highest efficiency. The number of

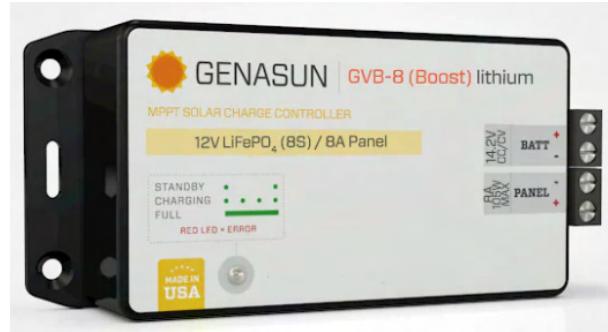


Figure 4.8. Genasun MPPT

cells per leaf matches the leaf area and the required voltage rating of the battery. When considering a battery with 4 cells in series, it is 14.8V. A single cell has a voltage rating of 0.6V. Thus, when considering a STEP-DOWN MPPT inverter, you should assume a cell voltage of at least the rated voltage of the battery. This means in this case at least 26 photovoltaic cells. Example wing with laminated solar cells on carbon fiber wing is shown on Figure 4.9.



Figure 4.9. Example wing with laminated solar cells

5. Design of Unmanned Aerial Vehicle

The aircraft was designed as a mobile flying platform that allows modular adaptation of its equipment to the nature of the mission it performs. The premise included the ability to assemble various executive components while maintaining the flight capability of the entire platform. Another of the aspects considered during the design of the solution was the ability to easily change the configuration of the modules as well as the availability of materials and the speed of manufacturing subsequent components. In order to achieve this criterion, ASA plastic was chosen as the main material used for the construction of the individual modules, from which the components were made on a 3D printer.

5.1. Project design in CAD software

The design of the unmanned aircraft first required determining the boundary conditions that would allow the creation of an aerodynamic model. In the case of the modular aircraft described here, the criterion was the model's weight reaching about 5kg and a low cruising speed allowing easy camera recording. The low speed combined with a wingspan of 3.5m allows for higher flight economy thereby extending the possibility of patrolling the area. Once these parameters were determined, aerodynamic analyses were performed in the XFLR5 program allowing the selection of a wing profile and calculation of the appropriate wing area and stabilizers. CAD model of plane is shown on Figure 5.1.



Figure 5.1. Plane design in Autodesk Fusion

5.2. Build process of Unmanned Aerial Vehicle

. The unmanned aerial vehicle model was built in cooperation with the AGH Solar Plane scientific circle. Workshop space was lent as well as the necessary tools and materials for the construction of the described unmanned aerial vehicle.

5.2.1. Wings build process

The wings, with a span of 3.5m, have been divided into three elements folded together using removable connectors for easier transport. The wings were made using composite lamination technology. Extruded polystyrene was used as a filling material, the appropriate shape of which was cut using a thermal plotter. The surface material used on the wings is carbon fiber. This material makes it possible to achieve considerable strength at low weight thus being an ideal choice for aeronautical applications. The material used was laid out at a 45-degree angle. This way of arranging the material allows for both bending and torsion strength. The wing girder responsible for giving rigidity was made of three layers of unidirectional carbon fabric. The wings were equipped with additional reinforcements made of carbon fabric around the mounting points of the engines and the bayonets connecting the different parts of the wing.

The bayonets connecting the wings were also made of carbon fiber, thus maintaining a low weight compared to the aluminum bayonets used previously. The amount of material used for the connectors was determined by research conducted by members of the AGH Solar Plane

scientific society. The component must be able to withstand the maximum anticipated overloads occurring during flight, thus being designed to break first when taking the force of impact.

5.2.2. Stabilizers build process

The stabilizers, like the wings, were made of carbon fiber using composite lamination technology. The construction process as well as the materials used are identical, but in this case no connecting rods were used due to the much smaller span of the ballasts. The aerodynamic profile of the ballasts differs from those used in the wings. This is due to the fact that in the case of the stabilizers, they should only stabilize the flight, not affect the deflection of the aircraft in either direction. The airfoil used on the wings has a differentiated shape, and its upper edge is longer than the lower edge thus generating lifting power at the appropriate speed of the descending air. The aerofoil used in the stabilizers is symmetrical, which means that the length of both edges is the same, and the lifting force is not generated. Controlled surfaces are responsible for the proper steering and yaw of the aircraft.

5.2.3. Fuselage build process

The fuselage of the aircraft was designed to reduce the weight of the structure as much as possible, thus leaving its modularity aspects. The fabricated model was adapted to the requirements of rapid assembly and disassembly for easy transportation. For this reason, the selected structural elements do not necessarily have the best aerodynamic properties, while the execution of the elements is possible under virtually any conditions. This is one of the necessary aspects considering the target use of the model in emergency situations. The elements were made with 3D printing technology using ASA filament and carbon tubes. Each of the elements used is sized to be printed on most available 3D printers, which means that completing a missing element is very easy. Carbon tubes were used to increase the rigidity of the structure. Power and control wires were routed inside the carbon tubes for aerodynamic and aesthetic purposes of the structure.

5.2.4. Camera gimbal build process

Installation of the camera in an aircraft requires proper mounting of both the sensor itself with the lens and the mechanism responsible for proper rotation and stabilization of the image. The aircraft tilts to the side during flight, which causes that at any given time the camera is not pointed vertically down thus not seeing the analyzed terrain. To eliminate this phenomenon, a servo has been installed that allows rotation in the axis of the plane's wings. The control signal is sent directly from the Raspberry Pi computer using a PWM signal. Information on the required orbit angle at any given time is taken from the gyroscope and accelerometer of the Pixhawk flight computer. It makes this information available via the Mavlink protocol, which is also used in this case, among other things, to send the next flight waypoints. The servo mechanism and the cradle to which the camera is mounted are located inside a modeled component made using 3D printing technology. CAD model of gimbal is shown on Figure 5.2.

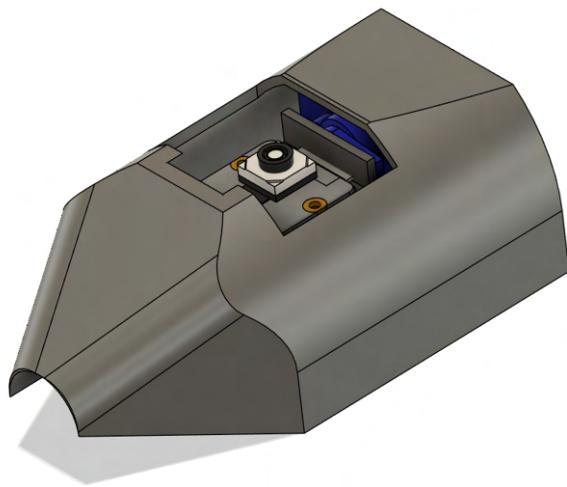


Figure 5.2. Gimbal design in Autodesk Fusion

6. Results

6.1. Prepare data for case study

The data used to test the performance of the algorithm on real data were prepared using a commercial drone DJI Mini 2. Pictures were taken moving along the designed route generating an 6x4 image matrix preserving geolocation data of individual samples as shown on Figure 6.1. Individuals were used to test the developed software. The images were taken in the village of Przysietnica in the Podkarpacie region, Poland. At a later stage, the photos were combined into

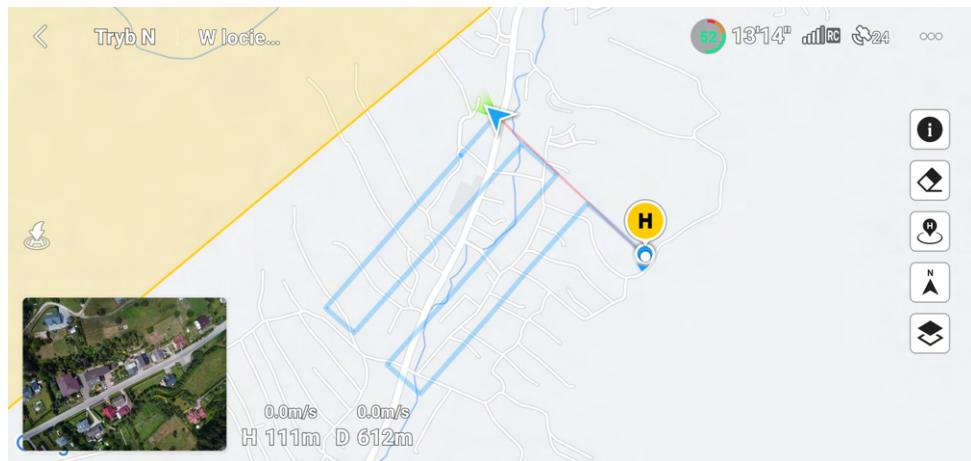


Figure 6.1. Geologs attached to DJI panorama photo

a photo panoramic using Adobe Lightroom Classic software generating a map of the area covering a rectangle of about 1x0.5[km] which is shown on Figure 6.2. Thus prepared data after appropriate processing allow to test the performance of the algorithm on real data.



Figure 6.2. A panorama photo taken with DJI drone

6.2. Results collected on case study

In order to test the performance of the algorithm, numerous tests were conducted on the previously presented test photo taken with a drone. The first task of the algorithm is to divide the image into individual fields using the SLIC segmentation algorithm. It allows to divide the image into smaller elements containing similar characteristics of the image inside. The running time of this algorithm is much smaller compared to the running time of the later-used algorithm of the traveling salesman. In addition, the research conducted showed that there is no dependence of the execution time on the number of elements of a given segmentation. This means that when executing the target algorithm, it is worth using this algorithm with more segmentations. In the graphic shown on Figure 6.3, you can see the comparison of results for 20, 40 and 50 elements of SLIC segmentation.

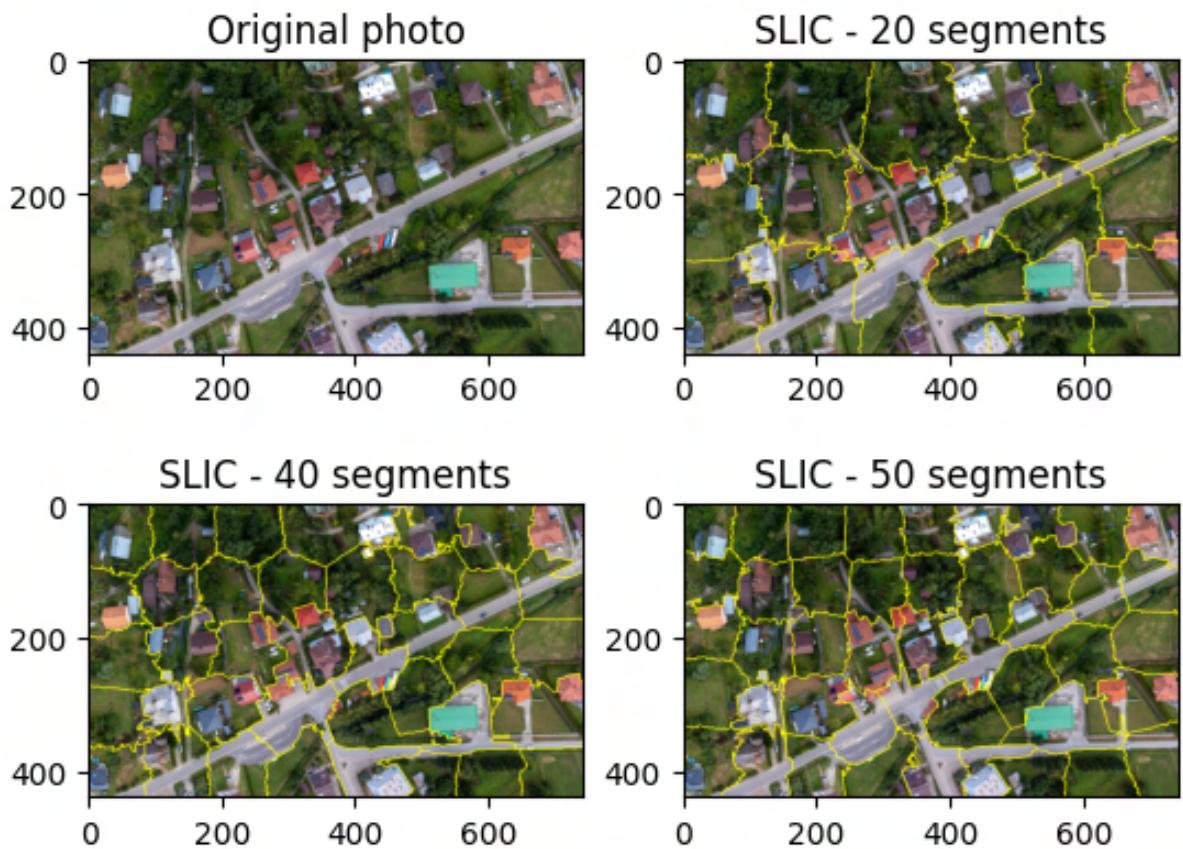


Figure 6.3. SLIC segmentation comparison

In the case of the site photo under consideration, the division into 20 elements is not sufficient. The divided segments have a diverse landscape, i.e. a single element contains both trees, houses and a road. Increasing the number of segments has a positive effect on the division, but the image is still not clear. In order to verify the correctness of the algorithm for a given application, a graphic with added single-color fields was developed. They are equivalent to an example of a natural disaster such as a fire or a flood. In the case studied, the superimposed fields are red and orange, symbolizing fire. Slic segmentation with added fields are shown on Figure 6.4. The efficiency of the algorithm in the case in question has improved significantly, and at a value of 40 segments the corresponding fields are separated. This allows us to identify points of interest for the route planning algorithm.

Potentially, in future applications, it would be possible to divide segmentation into two stages:

- first - would take place while flying at a higher altitude. In this case, more detail is required due to the coverage of a larger area. However, by the same token, it allows for a longer

route calculation time due to the higher altitude. In this case, it is better to use segmentation with more elements

- second - after determining the location of the focus of danger, the aircraft makes another flight at a lower altitude with the possibility of a more detailed picture of the terrain. The covered area is much smaller and less diverse, in which case segmentation for fewer elements would suffice. An aircraft at a lower ceiling also requires a shorter reaction time, which is directly related to the shorter flight path counting time

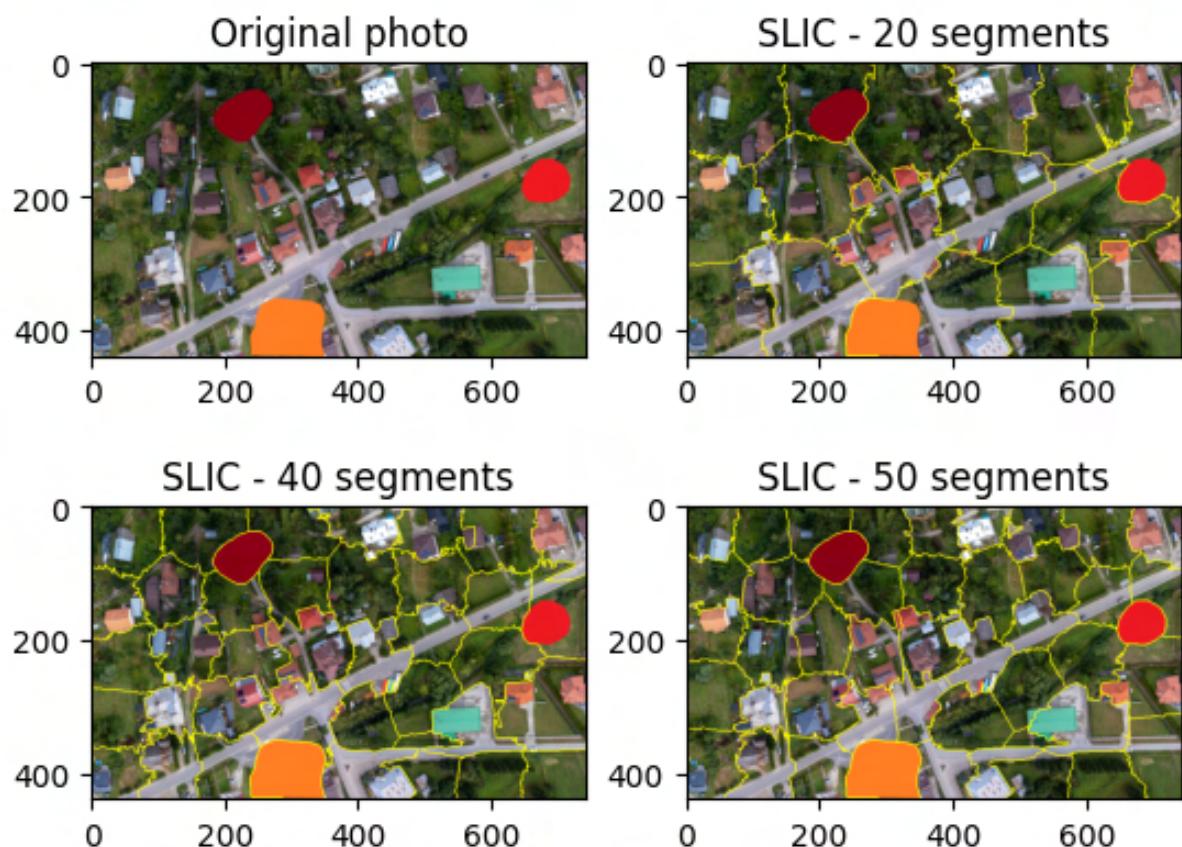


Figure 6.4. SLIC segmentation comparison on edited image

In the implemented code, all segments are taken into account when determining the path, while in order to reduce the execution time of the traveling salesman algorithm, the number of segments can be reduced after the first operation. After determining the centers of the segments, the average color of the pixels in the vicinity should be examined, and then compared with the predicted color occurring in the case of a given disaster. Thus, in the case presented and the designation of 50 segments, you can check in which of them the predominant color is red. Having

done so, only those segments should be designated as points of interest, and then passed on to the next algorithm. Thus, the resource-intensive traveling salesman algorithm has less data to process and its execution is possible on less resource-intensive computers. Path planning executed on 50 segments is shown on Figure 6.5.

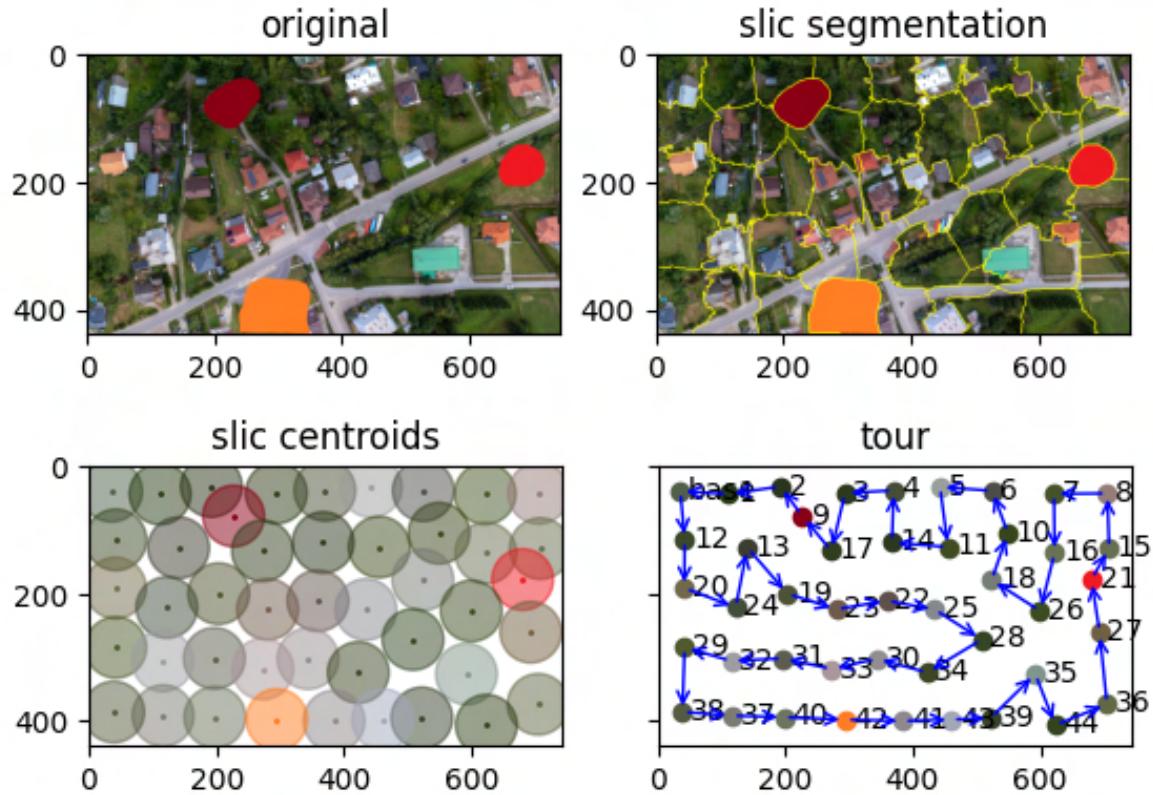


Figure 6.5. Path planning executed on 50 segments

The traveling salesman algorithm's determined route has to be adjusted to the aircraft's flight capabilities. The result generates a route consisting of straight lines drawn between points which is impossible for an unmanned aircraft equipped with wings. In the case of a multicopter, stopping at points is possible, and in such a case this requirement would not apply. The planned route for the case under consideration requires a turning radius of at least 10 meters, which is determined by the design parameters of the aircraft. To this end, the next step of the algorithm is to calculate a curve passing through the required points, but with smoothed turns. Once the waypoints determined according to the B-spline have been determined, they should be uploaded using the Mavlink protocol to the Pixhawk flight controller. Keep in mind that the aircraft will follow the specified route as far as possible, while in case of severe conditions such as wind, it

allows the model to be slightly shifted relative to the course. Path planning with route smoothening is shown on Figure 6.6.

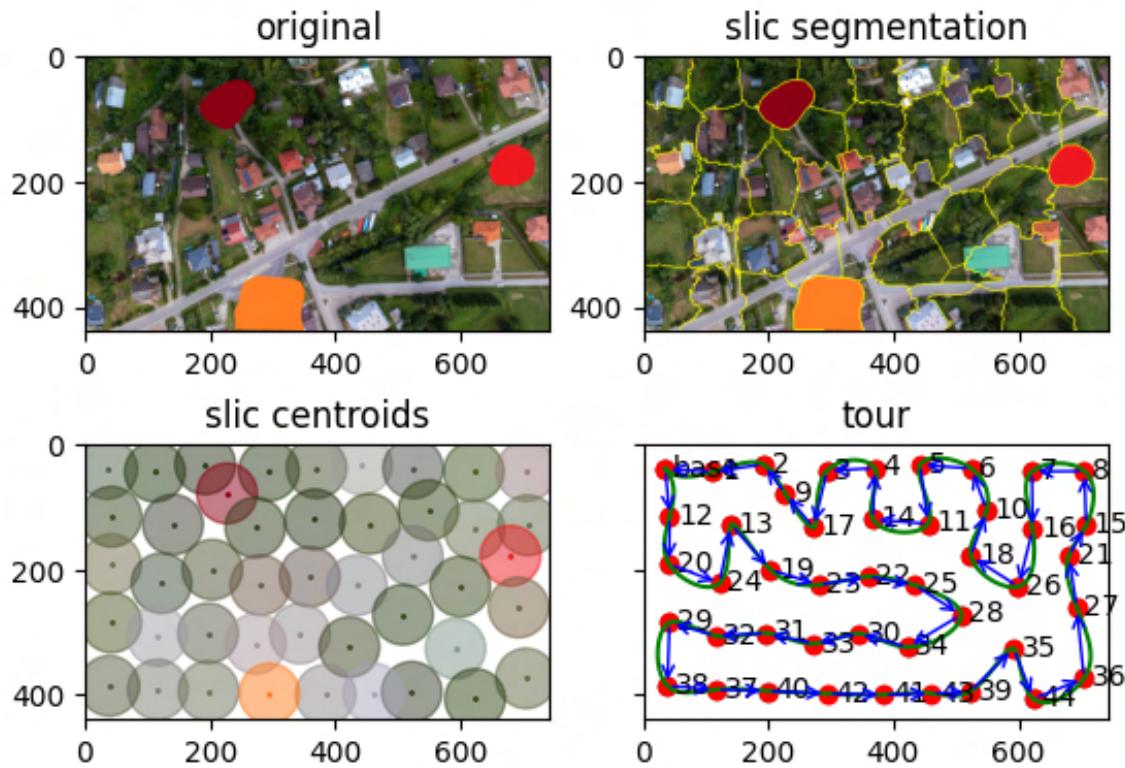


Figure 6.6. Path planning executed on 50 segments with adding spline as a route

The controller takes into account the possible shift and corrects the route to return to it at the next points. An important setting is to disable returning to missed points. It allows you to specify the tolerance of a waypoint so that the aircraft, having missed it, can move on to the next one instead of trying to return to the previous one. This is a very important configuration parameter when flying in windy weather. As for the determination of the minimum turning radius, the aircraft will adjust automatically with the help of the limitation of the maximum lateral yaw parameter. This means that even in the case of incorrect determination of the waypoint (too small a turn radius) it is the controller itself that will increase the arc of the turn going to the next point. The yaw angle should be adjusted according to the cruise speed, weight and wing area of the model. The higher the cruise speed, the tighter the turns can be, but this will make it more difficult to take pictures and reduce the energy efficiency of the vehicle.

Code used in this project is available at: <https://github.com/FilipPieniazek/Path-Planning-for-a-Solar-Powered-Unmanned-Aerial-Vehicle> [15]

6.3. Analysis and processing of logs

The Pixhawk flight controller collects data in the form of .tlog files containing both stored vehicle configuration values and information on current flight parameters. The information can be visualized using the Mission Planner software, where full simulations can be run for the map, artificial horizon, speed, or values fed to individual servo and motor control channels, among other things as shown on Figure 6.7.

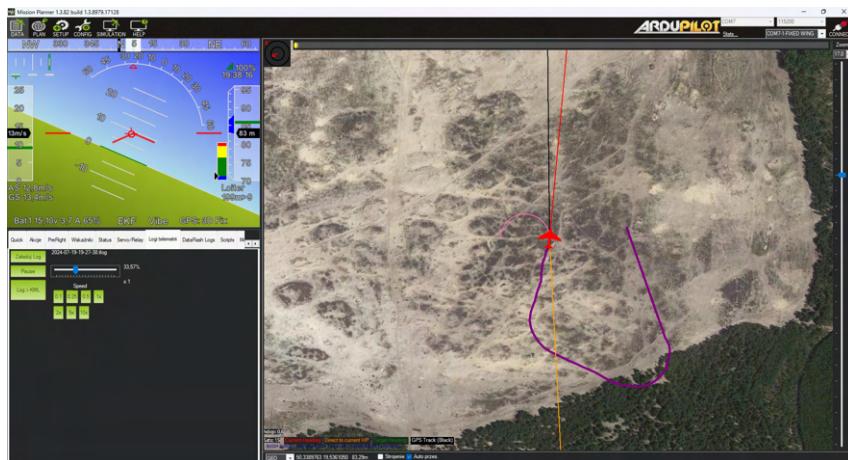


Figure 6.7. Mission planner logs visualisation

These files can also be converted to .txt or .csv file format using this very software. Having the data prepared in this way, it is possible to prepare a more detailed analysis, for example, with the help of the Python language, where, using the pymavlink library [16], it is possible to read the individual messages sent during the controller startup session. Placing the data in a programming environment allows full analysis of the data, depending on the parameters of interest. This functionality is useful, for example, for analyzing energy consumption during flight, or individual maneuvers, thus creating the possibility of adapting parameters for optimizing the route planning algorithm. In the first of the graphs shown below, you can see the values of the accelerometer readings of individual aircraft deflections. Analysis of these parameters, combined with knowledge of the aircraft's characteristics, can allow limiting individual deflections for

safety or energy optimization reasons. Visualisation of attitude mid-flight created with Python code is shown on Figure 6.8.

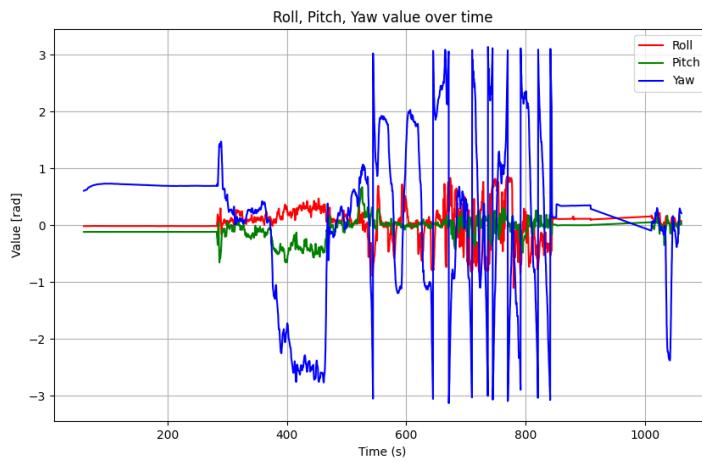


Figure 6.8. Python logs attitude visualisation

The graphic below shows geolocation data from the flight in question. This information can be useful in combination with the API provided by Google [17] allowing to acquire a real-time map of the area. This makes it possible to obtain topographic maps of the terrain in addition to overlaying the image for a more realistic representation of the area. These contain elevation values of the terrain at particular points, and thus allow corrections to be made to the route planning algorithm. This aspect is especially important for missions in high mountainous terrain, where the difference in terrain elevation is significant, or there are objects against which the aircraft may crash during flight. In such a case, the aircraft, by sending a query to the API, can obtain information on the required increase in ceiling at particular locations or skip certain zones. Visualisation of GPS location created with Python code is shown on Figure 6.9.

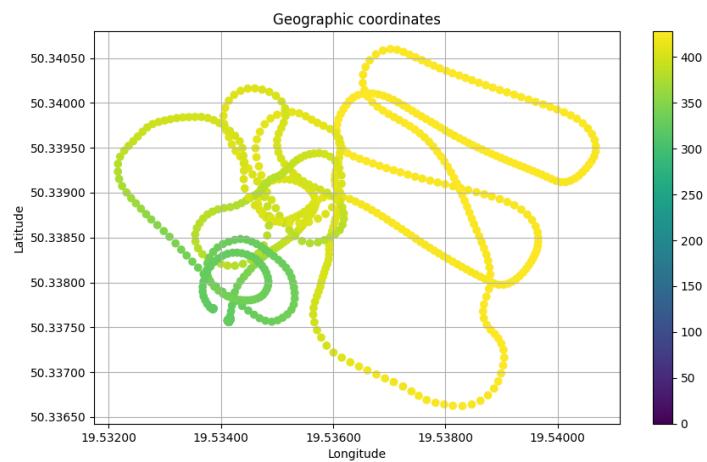


Figure 6.9. Python logs position visualisation

Another of Mission Planner software's features is the ability to generate .gpx route files of a given flight. The file records geolocation data as well as information on the flight mode the vehicle was currently in. This makes it easy to distinguish between moving in autonomous mode and manual flight fully controlled by the pilot, or flying in one of the modes with stabilization assistance (e.g. Stabilize, Loiter, FFWA). This visualisation is shown on Figure 6.10.

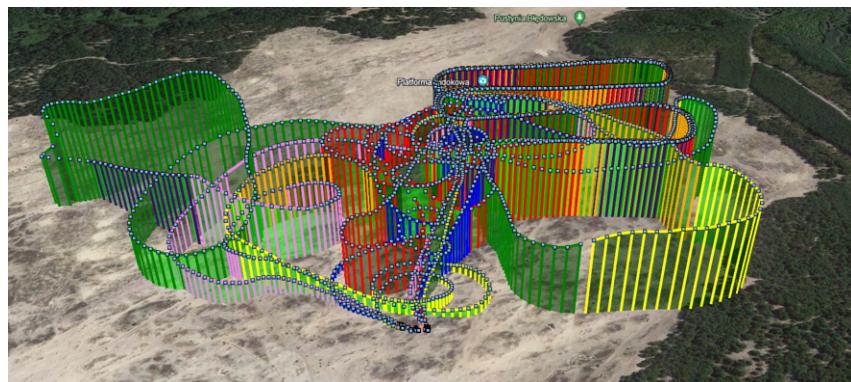


Figure 6.10. Google Earth 3D logs visualisation

6.4. Developed aircraft

The aircraft model presented in the previous chapters was realistically made at the prototype stage. The structure was made according to the design keeping the previous assumptions. Unfortunately, due to limited time, the aircraft was tested only in autonomous modes implemented

within the Pixhawk controller. Autonomous modes and data transmission to the controller using the Mavlink protocol were tested. A camera was not mounted on the vehicle, thus making it impossible to test image capture and real-time route calculation. The model was made with the help of members of the AGH Solar Plane research club. The aircraft in its current state will be used as a modular aircraft project at the Droniada 2024 competition. Developed model, see Figure 6.11 and Figure 6.12.



Figure 6.11. Developed UAV model

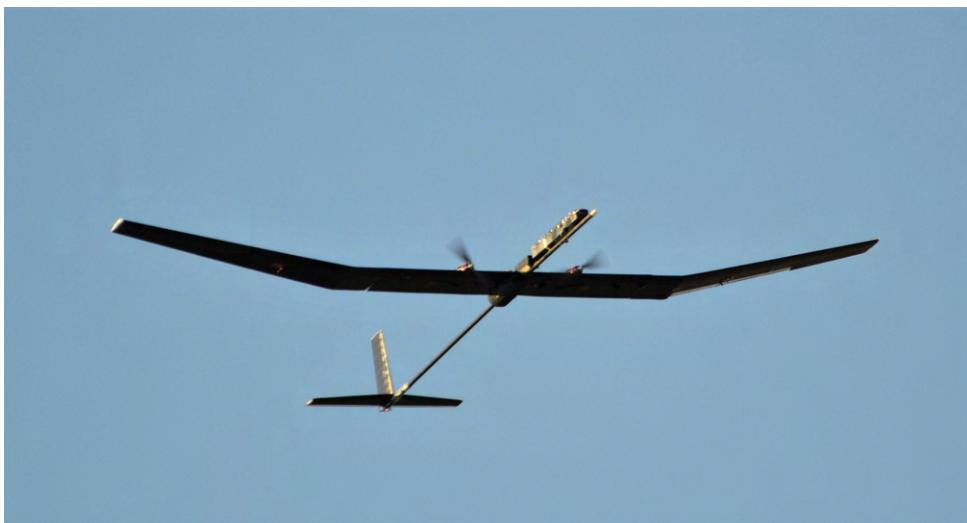


Figure 6.12. Developed UAV model in flight

7. Conclusions and development opportunities

The project in its current form allows for the construction and implementation of a working patrol platform used during emergency management missions. Equipping the system with additional peripherals may allow the solution to have a longer range or enable it to be used in hard-to-reach terrain. In addition to mechanical and electronic solutions that optimize the solution's range, attention should also be paid to the speed of the computing part and the range of the platform's information transfer.

7.1. Machine learning

Machine learning can be used to speed up the computation time of a route planning algorithm. As mentioned earlier, it can be used in several ways, such as quickly eliminating inefficient routes, indicating an initial solution close to the optimal one, or splitting the problem into smaller problems - easier for the traveling salesman algorithm to compute. Without a real-world implementation, it's hard to determine the possible benefit during program execution, while when considering running on a portable board computer, hardware requirements should be kept to a minimum.

7.2. Plane launcher

The blueprint of the actual unmanned aircraft described in the project can be further expanded with elements to facilitate its operation in real applications. Aircraft built in this way are most often launched using a handheld launcher by a person qualified to do so. However, this method is limited by the maximum weight of the aircraft and its span. As the values of these parameters increase, the activity is made more difficult as shown on Figure 7.1. In the case of



Figure 7.1. Plane throw at UAV Competition in Turkey

the design described above, this is almost impossible due to the significant weight of the aircraft. One possibility for launching vehicles of such mass is to build a suitable launcher that imparts acceleration to the aircraft as it takes off. One of the projects of the AGH Solar Plane circle is a launcher suitable for launching aircraft weighing up to 10kg. See figure 7.2. The use of such a design would make it possible to take off in any conditions, without the need to equip the aircraft with landing gear, and thus no need for a runway. The launcher is made of aluminum profiles divided into sections up to 1.5m long. The solid structure is collapsible, thus, like an aircraft, it can be transported by any means of transport to hard-to-reach areas. The force-generating element of the launcher are two rubber bands connected in parallel located inside the launcher. They are tensioned by means of a steel cable and a hand winch. Ready for launch, the aircraft is placed on an engineered cart equipped with four sets of wheels placed in guides.



Figure 7.2. Plane launcher designed at AGH Solar Plane

7.3. Antenna tracker

The unmanned aerial vehicle is equipped with several data transmission systems using different transmission bands. The presented solution uses the following radio bands:

- 868MHz - the band used to transmit telemetry data between the aircraft and a ground station equipped with Mission Planner software allowing both sides to communicate and change parameters in real time
- 2.4GHz - radio band used by the control apparatus. Used by the pilot when flying in manual or assisted mode. Lack of communication in this case triggers the failsafe procedure in the model. Most often, this means an autonomous return to the starting point or circling the model while waiting for the signal to recover
- 5.8GHz - the band used by the transmitters of FPV(First Person View) cameras allowing to monitor the situation “from the cockpit” of the model

All the frequencies shown, depending on the region of the world, are limited by legal regulations, where the maximum power of the transmitted transmission is described. This means that the real range of the solution used is limited. However, these regulations do not specify the antennas used for data transmission. This allows the use of directional antennas, covering a much smaller area of the sphere, but having higher gain. With proper control of the antennas so that they continuously follow the moving aircraft, the transmission range can be significantly increased, thereby allowing further patrolling of the area. An antenna tracker is a device that changes the position of antennas placed on an arm that is moved by stepper motors. Using information on the station's current geolocation data and the position of the aircraft in the sky, we are able to determine the angle at which to position the transmitting and receiving antennas.

A prototype of such a solution, like the launcher, was made as part of the activities of the AGH Solar Plane research circle as presented on Figure 7.3. With such a solution, the transmission range increases even several times, especially in a situation where there is no requirement for two-way communication(only sending from the station, no requirement for confirmation from the aircraft). Effective two-way communication requires the placement of higher-gain antennas on the aircraft as well, which can be a design challenge.



Figure 7.3. Antenna tracker designed at AGH Solar Plane

Bibliography

- [1] Mavlink. *Mavlink protocol documentation*. [Online; accessed September 20,2024].
- [2] Ardupilot. *Ardupilot and mission planner documentation*. [Online; accessed September 20,2024].
- [3] Ceneo. *Dji Mini 2 drone*. [Online; accessed September 20,2024].
- [4] ULC. *Aviation law in Poland*. [Online; accessed September 20,2024].
- [5] Pansa. *Aviation law - open category*. [Online; accessed September 20,2024].
- [6] Pansa. *Aviation law - special category*. [Online; accessed September 20,2024].
- [7] Raspberry. *Raspberry Pi camera module documentation*. [Online; accessed September 20,2024].
- [8] Colorado. *Mavlink protocol diagram*. [Online; accessed September 20,2024].
- [9] Farnell. *Raspberry Pi*. [Online; accessed September 20,2024].
- [10] Drony. *Pixhawk*. [Online; accessed September 20,2024].
- [11] Modelemax. *Motor and ESC*. [Online; accessed September 20,2024].
- [12] Avifly. *Battery*. [Online; accessed September 20,2024].
- [13] Modelarski. *Servo*. [Online; accessed September 20,2024].
- [14] Mateksys. *Pitot*. [Online; accessed September 20,2024].
- [15] Filip Pieniazek. *Project code repository*. [Online; accessed September 23,2024].
- [16] Mavlink. *Pymavlink library documentation*. [Online; accessed September 20,2024].
- [17] Google. *Google maps developer API*. [Online; accessed September 20,2024].

- [18] Wenxin Le et al. “Coverage Path Planning Based on the Optimization Strategy of Multiple Solar Powered Unmanned Aerial Vehicles”. In: *Drones* 6.8 (2022). ISSN: 2504-446X. DOI: [10.3390/drones6080203](https://doi.org/10.3390/drones6080203).
- [19] Siyuan Li et al. “Navigation and Deployment of Solar-Powered Unmanned Aerial Vehicles for Civilian Applications: A Comprehensive Review”. In: *Drones* 8.2 (2024). ISSN: 2504-446X. DOI: [10.3390/drones8020042](https://doi.org/10.3390/drones8020042).
- [20] Dipraj Debnath et al. “An Integrated Geometric Obstacle Avoidance and Genetic Algorithm TSP Model for UAV Path Planning”. In: *Drones* 8.7 (2024). ISSN: 2504-446X. DOI: [10.3390/drones8070302](https://doi.org/10.3390/drones8070302).
- [21] Salvatore Rosario Bassolillo et al. “Path Planning for Fixed-Wing Unmanned Aerial Vehicles: An Integrated Approach with Theta* and Clothoids”. In: *Drones* 8.2 (2024). ISSN: 2504-446X. DOI: [10.3390/drones8020062](https://doi.org/10.3390/drones8020062).
- [22] Lexu Du et al. “A Multi-Regional Path-Planning Method for Rescue UAVs with Priority Constraints”. In: *Drones* 7.12 (2023). ISSN: 2504-446X. DOI: [10.3390/drones7120692](https://doi.org/10.3390/drones7120692).
- [23] Hailong Huang and Andrey V. Savkin. “Path Planning for a Solar-Powered UAV Inspecting Mountain Sites for Safety and Rescue”. In: *Energies* 14.7 (2021). ISSN: 1996-1073. DOI: [10.3390/en14071968](https://doi.org/10.3390/en14071968).
- [24] Luis D. Ortega et al. “Low-Cost Computer-Vision-Based Embedded Systems for UAVs”. In: *Robotics* 12.6 (2023). ISSN: 2218-6581. DOI: [10.3390/robotics12060145](https://doi.org/10.3390/robotics12060145).
- [25] Maria Höffmann, Shruti Patel, and Christof Büskens. “Optimal Coverage Path Planning for Agricultural Vehicles with Curvature Constraints”. In: *Agriculture* 13.11 (2023). ISSN: 2077-0472. DOI: [10.3390/agriculture13112112](https://doi.org/10.3390/agriculture13112112).
- [26] Jinbiao Yuan et al. “UAV Path Planning With Terrain Constraints for Aerial Scanning”. In: *IEEE Transactions on Intelligent Vehicles* 9.1 (Jan. 2024), pp. 1189–1203. ISSN: 2379-8904. DOI: [10.1109/TIV.2023.3307217](https://doi.org/10.1109/TIV.2023.3307217).
- [27] Jinyu Fu et al. “An FTSA Trajectory Elliptical Homotopy for Unmanned Vehicles Path Planning With Multi-Objective Constraints”. In: *IEEE Transactions on Intelligent Vehicles* 8.3 (Mar. 2023), pp. 2415–2425. ISSN: 2379-8904. DOI: [10.1109/TIV.2023.3237518](https://doi.org/10.1109/TIV.2023.3237518).