# Optimized UAV Waypoints selection for maximum area coverage

Mark Bastourous

LE2I - Laboratoire Electronique, Informatique et Image
Universite Bourgogne Franche-Comte


Under supervsion of:
David FOFI

A Thesis Submitted for the Degree of
MSc in Computer Vision (MSCV)

· 2016 ·

**Abstract**

Area coverage is a very crucial application in many domains. It requires time, precision, and sometimes repetition which prevails the important use of robots. Over many years in research, several robotics platforms on the ground, in aerial, or underwater have been studied,developed and tested. These robots can reach,navigate and execute tasks that humans are not capable of or can be tedious or harmful.

Coverage path planning (CPP) is the problem of finding the most suitable path with least cost in terms of time, power, and dynamics that can cover a given area. In this work the CPP of unmanned aerial vehicles (UAV) problem is partitioned into optimally selecting the poses that will guarantee highest area coverage, then plan the path traversing these selected poses. Path planning is done in two stages consisting of; global and local planners. The global planner is responsible for generating the path with minimal length passing by the predefined waypoints. The local planner's role is to find the clear way in between these waypoints that will be used to build the robot's trajectory.

The problem is approached from an optimization point of view. Genetic algorithm (GA) and particle swarm optimization (PSO) are tested to solve the problem of choosing the coverage waypoints. Then GA is re-used again to solve the global planner challenge. The robot poses from the coverage are formulated by the well-known computer science problem travelling salesman problem (TSP). Then several methods are tested to solve the local planner challenge like linear, spline piecewise, artificial potential field. After generating the path, in the phase of executing the UAV trajectory both in simulation and on the practical platform.

Promising simulation results on V-REP led to experimenting it on practical UAV from Parrot company which is called AR.drone 2.0. The acquired images are then mosaicked to show the coverage of the area with as minimal amount of images as possible. Images are acquired at the waypoints, not during the whole flights like what is common in the literature. This will significantly reduce the computation time needed to generate the final mosaick of the scene.

# Contents

# List of Figures

# List of Tables

# Acknowledgments

I would like to express my deep gratitude to my professors for their tremendous help. There are no words can describe my recognition to the effort, patience and guidance provided by my supervisor prof. David Fofi. Also i can not forget the great help and collaboration done by David Strubel during the whole thesis period. Helping me with all the possible ways he can. Also all the colleagues and stuff in LE2I lab where i have spent magnificent 5 months research internship experience.

To all my new friends from MSCV6 and the previous or following batches. For my old friends who are all the time showing support and understanding.

To my father, Nabil, whose memorial day is approaching and great eternal memory is giving me support, and the memory of my brother Michael, to my mother, Mervat. To my dear aunt Afaf, my uncle Ibrahim, Soliman for their great support. To the memory of my recently passing away uncle whom my studies were a barrier infront of not attending his funeral. Finally, special thanks to my family and parents for their unconditional and endless support to help me achieve who I am today.

# Chapter 1

# Introduction

## 1.1 INTRODUCTION

Aerial robots have grown great attention in the last decades due to its capabilities in solving many problems in many domains. They are still under study, investigation and development because of the several constrains and challenges that are in software and hardware of its construction on many levels and layers.

Some of the software challenges are state estimation, control, decision making, mapping, and path planning. Some of the hardware challenges are the weight/load ratio, power source, sensors, etc.

There are various applications that practically make use of unmanned aerial vehicles (UAV) and much more are still under study and development in the research labs and institutes. These applications are covering many fields of interest in both civilian and military purposes. Some of these practical applications are search and rescue, inspections, surveillance, photography, agricultural terrain mapping, mineral exploration, etc. Some prospective applications like medical cargo delivery because it will not rely on the normal road maps if they exist, or traffic constrains.

There are many types of UAVs categorized based on the geometry and designs like fixed wing, flapping wing, and rotor crafts, mentioned in more details in [43]. Figure 1.1 shows some of the current UAVs. From cheap and small toys that you can be bought anywhere as the RC planes, toy quadcopter, to big military projects such as the Global Hawk.

UAVs have recently reached a decline in their cost especially the quadcopters. In this thesis during the practical implementation part, quadcopters as one type of the rotor crafts will be used. The quadcopter used is Ar.drone 2.0 from the french company Parrot. It is off shelf drone

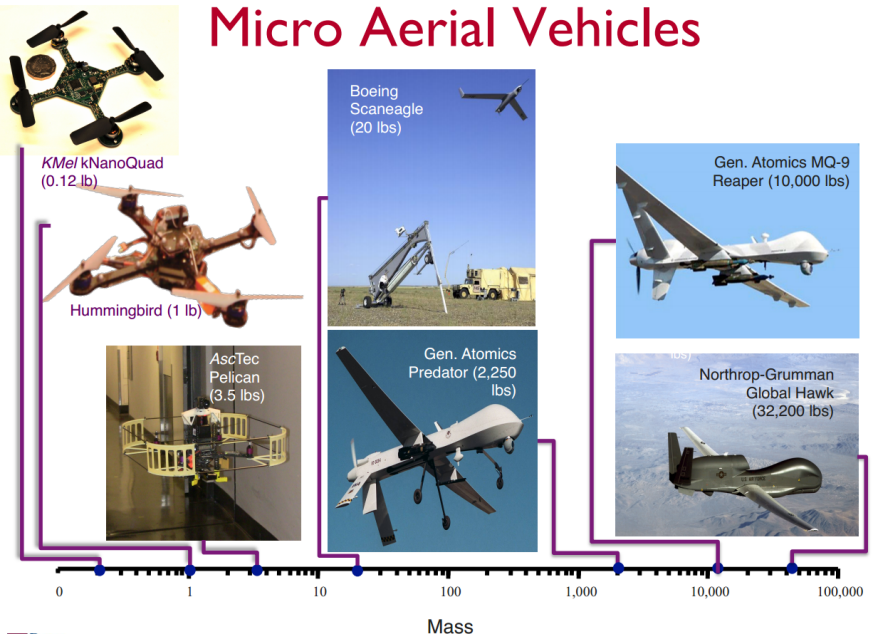



Figure 1.1: Micro aerial vehicle

that can be found on the market with comparable cheap price in range of 300 euros depending on the extras. It has been used several times in research labs and studied like in [5], [24],etc.

There are many advantages that make quadcopter as a specific type of UAVs, suitable for both indoor and outdoor applications. It takes off vertically, do not need a runway, hover in its place, comparably light weight, and small in size.

For simulation V-REP with a model of quadcopter available was used. Some modification in this model was introduced to cope with the problem being solved and will be discussed later in more details. In both the practical work and simulation, ROS packages were used, tuned and implemented to control the quadcopter.

Some of the previously mentioned applications require area coverage of outdoor or indoor mapping. Coverage path planning is an active research topic as the sub division of the general path planning problem that is studied by robotics domain for several decades till our day. It has been applied on many platforms in many areas where these platforms work. Normally the area to be covered is not a regular one. Most of the literature review that comes to the awareness to the author of this thesis are concerned with uniform areas and prior map with static or dynamic obstacles. Platforms here means the mobile robots like autonomous underwater vehicles(AUV), unmanned aerial vehicle(UAV), or ground vehicles.

Prior information about the environment is assumed to be given in the form of a rough map

for the area required to be covered. For this work, we model the set of coverage problems as arc routing problems. Although these routing problems are generally NP-hard, our approach aims for optimal solutions through the use of low-complexity algorithms in a branch-and-bound framework when time permits and approximations when time restrictions apply.

The final objective of this thesis research is to design a global optimization scheme allowing a cameras network to be self organized or to plan single trajectory of a flying robot equipped with a camera, according to fixed priority and constraints, in order to ensure a full coverage of a given scene. Applicable solution for both indoor and outdoor with ensuring coverage of the terrain while minimizing path repetition. Then building a mosaicking of the area covered and compare it with the original scene.

In the next subsection the objectives of this thesis will be discussed.

### 1.1.1 Objectives

- Study coverage path planning problem, implement the suitable one.
- Examine and validate the path planning in both simulation and real world with practical hardware.
- Construct a final mosaicked scene after the UAV acquire images.

### 1.1.2 Thesis Organization

This thesis is organized as follows. Chapter 2 discusses the background and shows the literature review done by the author. Then comes the description of the methodology which is written into two chapters; 3 and 4. In chapter 3, the methods of selecting the best waypoints is discussed. Chapter 4 contains the path planning methods developed and used. Also explanation of the combining several methods is provided. The whole pipeline is being presented by the end of this chapter too. Results and experiments in both real life and simulation is discussed and shown in chapter 5. Finally conclusion and future work are addressed in chapter 6.

# Chapter 2

# Background

The merit of this thesis in the coverage path planning context is splitting and decoupling the process of the coverage problem. The process will consist of: finding the proper pose and orientation of camera views that maintain area coverage, then apply path planning techniques to traverse these poses. So that the first part can be used as a solution by itself with positioning cameras in the obtained positions. The second merit is minimizing the overlapping paths and implementing the path planning taking into account the footprint of the robot and the camera, not relying on the assumption of robot representation as a point.

## 2.1 Related Work in path planning

### 2.1.1 Coverage Path Planning

There are two main surveys that this literature review is relying on in picking and comparing the path planning technique to be further implemented and tested on the quadcopter. These surveys are; a survey on coverage path planning for robotics [17] and coverage for robotics survey of recent results [8]. There are a lot of other work extensively explored from many research areas beside path planning like computer vision and graph theory. The problem can be studied from many sides like mapping, searching, patrolling, and applications. We focus here on the path planning and area coverage problems with slightly different approach than the common one. we take into consideration the footprint of the camera and one of the goals is to reduce the number of waypoints with maximum area coverage. Full area coverage in terms of 100% is not the constraint.

In the literature generally there is always a navigation pattern to pass through the whole space; the robot needs to cover. This pattern is done after decomposing the map into either

regular cells or irregular ones like Voronoi. These methods can be categorized as heuristic and approximate, partial-approximate and exact cellular decompositions as discussed in [8].

Coverage algorithms can be classified as heuristic or complete depending on whether or not they provably guarantee complete coverage of the free space. They can also be classified as online or offline based on when they are going to be applied. In the case of being applied to the robot while it is executing its path, then it is online. They are considered offline if a prior map is given and the path planning is obtained, and then the robot execute this path.

The heuristic approach that is based on a random path which is like the one equipped in some lawn mower and vacuum cleaner robots as mentioned in [8]. This approach can not be considered for aerial robots applications due to its expensive time and energy costs.

Coverage algorithms can rely on cellular decomposition with many techniques that an extensive comparison was given in [17]. Then an ox plowing motions (Boustrophedon manner) is used like the example in planning algorithm book [27]. As the approach used in this thesis will not rely on map decomposition, further investigation can be found in the cited papers and book.

Other approaches like Artificial Potential Fields (APF), node based (graph) algorithms and neural networks can be used instead of the map decomposition.

### 2.1.2 Coverage Path Planning for UAVs

There are various ways to obtain and optimize the path planning. CPP problem is a sub-field of path planning and well studied for ground vehicles like cases in [8, 17, 45].

In operations research, the CPP problem represents the environment as a graph. The problem can be represented as the travelling salesman problem(TSP) or postman problems to generate optimal solutions. In the graph representation, locations in the environment is represented as nodes in the graph and the paths between the locations are the edges. Each edge has a cost assigned to it. The cost can represent measurements such as distance(Euclidean,Mahalanobis,etc) between locations, terrain traversability, travel time or a combination of several metrics. These edges can be constrained or unconstrained with directions. More precisely undirected , directed graphs.

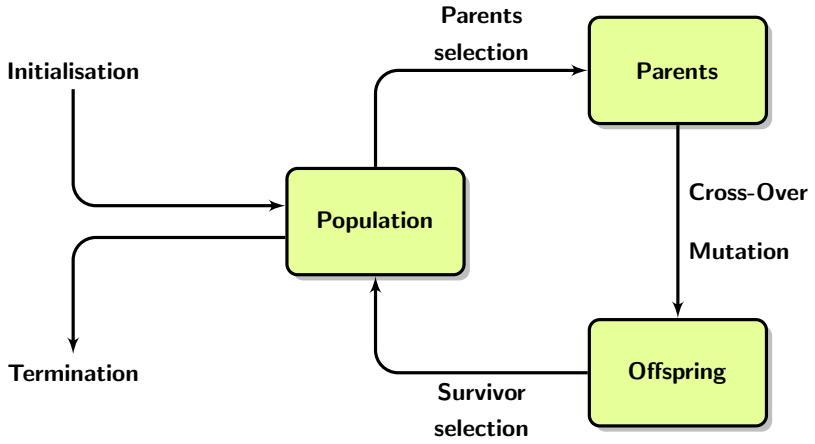
Voronoi diagram is also another possible solution as developed in [6] for decomposing the desired area and find the feasible solution from the starting point to the goal point through two-step path planning algorithm, but it doesn't take into consideration the repetition rate constraint or full area coverage. For these reasons considering problem representation as arc routing problem is most beneficial. Solving these problems can be achieved by using bio-inspired algorithms like neural networks or evolutionary algorithms.

TSP is considered one problem of a large class of problems known as combinatorial problems.

It is introduced in UAV path planning for several purposes like refueling depots in [40], multiple UAV cooperative reconnaissance in [42]. This is a non-deterministic polynomial-time hard (NP-hard) problem, that is sub-optimally solvable by many approaches as found in various research fields [1]. Evolutionary approaches specifically will be considered in this thesis.

#### Evolutionary algorithms

Evolutionary Algorithms(EA) as a sub branch of meta-heuristic optimization algorithms, imitates the biological process of evolution in nature. There are various algorithms that are branched from it like Ant colony optimization (ACO), particle swarm optimization (PSO),Genetic Algorithm GA and much more which can be studied from [11,39]. GA uses techniques of inheritances, mutations, selections and crossovers of chromosomes over several generations of possible solutions to find convergence to the most optimized solution. Explanation in the next chart is the general abstract view of GA, more details will be explained in the methodology chapters.




Coverage path planning can be categorized into sampling based algorithms, node based algorithms, mathematic model based algorithms, bio-inspired algorithms. Then comes the category emerging lately in the next subsection.

### 2.1.3 Hybrid Algorithms for path planning

Having hybrid path planning algorithms by combining several algorithms together to achieve better performance. These combinations comes in terms of fusing several path planning algorithms in a layer by layer way. This will lead to better solutions to overcome the problems like local minima, long time to find the solution, sharp-edged paths. The algorithms fusion can be simultaneous or successive. For instance, Scholer et all [38] applied visibility graph, then Dijkstras algorithm to find an optimal solution for UAV path planning problem in 3D.

## 2.2 Waypoints Selection

Finding the waypoints to have the best coverage of the area is close to the problem of positioning cameras or sensors to cover an area efficiently. Sensor positioning problem has been investigated since a few decades, mainly for video surveillance [44]. Without any additional constraint, this problem is NP-Hard as stated in [12, 19, 29, 30, 44] for the Watchman Route Problem (which is very similar to the optimal positioning waypoint for UAV path). Two non-optimal solutions have been proposed. The first one is based on Art Gallery Problem (AGP) [29, 30] and the second one is based on the Wireless Sensors Networks [3, 18, 25, 35] trying to find the best position to design an efficient network which can collect data with any kind of sensors.

However, the solution proposed to the problem addressed the coverage problem but linked with additional and specific constraints, which are out of our scope.

One of the algorithms used is the Particle Swarm Optimization (PSO) as detailed in [22,34]. Zhou *et al.* [34], some experimental results are provided and one solution running in real time is proposed. However, the scene used in these experiments is rather small and many cameras are employed to fully cover it. On the other hand, [22] uses a cost function but the cost function is not only focused on the position for surveillance and coverage, but also handling resolution and lighting, which affect the final solution by not covering the under illuminated areas.Reddy *et al.* [22] also introduced the concept of an acceptable response, allowing non-optimal/sub-optimal solutions. If the coverage score is higher than a given threshold, the solution is accepted and not locked by the research of an optimal solution.

## 2.3 UAV Localization

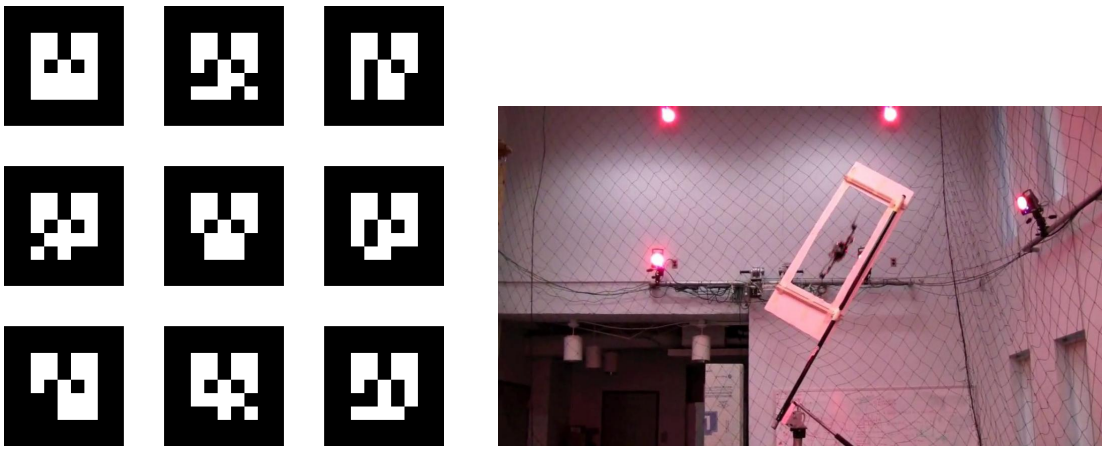
UAV localization problem is the challenge of finding the pose in terms of position and orientation of the aerial robot from the reference frame. This reference frame can be the initial point the robot is launched from, or a point on the map. There are two main categories for localizing the robot; either indoor or outdoor. For outdoor, GPS-based navigation system is often used to determine the UAVs absolute position. Also equipped with inertial measurement unit (IMU), UAV's orientation and acceleration can be estimated. State estimation of the UAV is obtained by coupling the previously obtained data with the UAV's or/and environment mathematical model. This information is essential for autonomous navigation and of significant importance to aid the pilot to achieve smoother navigation.

Unlike ground vehicles, just holding the robot's pose requires motor orders to withstand and hover its place, giving order continuously to the robot. This is particularly the case for an aerial vehicle - while for ground-based vehicles not moving typically is a trivial task. Holding a position in the air requires constantly counteracting minor randomly induced movements, which

in turn requires a method to detect these movements. While for indoor localization challenge; GPS data can not be reliable. Also employing the IMU data can provide relative location, over time small errors will keep accumulating and drift will happen and affect the localization. Additional information about global positioning should be available to recover for this drift using means of filtering and sensor fusion [4].

These alternative localization methods can be categorized as

- Rely only on the UAV's sensors IMU, onboard camera.
- External fiducial markers tracked by onboard camera.
- Using external tracking devices.



(a) AR tags (b) Motion tracking cameras

Figure 2.1: Localization techniques

An example of the available external tracking device is motion capture camera that allow the robot to measure its position with very high precision and accuracy as in [32]. Reflective markers like retroreflective dots attached to the robot, the cameras can estimate the position of each reflective marker and these cameras can do it at split second timing exceeding speeds of 100-200 times a second. This system will not be used as it requires setup up of several cameras which cost dozens of thousands of euros. The other external solution provided for example in [28] using external markers and detected using onboard camera. This solution could have been considered, but it is not the chosen one to be implemented in this thesis.

There is wide range of sensors can be equipped on the UAV to help its localization like laser range scanner, monocular camera, stereo camera, RGB-D sensor, or ultrasonic range sensors. The usage of the monocular camera is chosen to be the solution for this thesis. They provide

competitive advantages; as they are cheap, energy-efficient, small and light. For instance, the system of navigation in this paper [16] is based on vision optical flow and laser FastSLAM.

Simultaneous localization and mapping (SLAM) can be achieved by combining visual pose estimates from the monocular camera with additional sensor measurements available like the implementation of Tardif et al. in [41]. The tool developed by the computer vision group at the Technical University of Munich as mentioned in the papers [13, 14], is utilized on an AR.Drone 2.0 available in the lab with minor modification.

## 2.4 Mosaicking

Mosaicking is one of the active research areas in computer vision field. It is a technique used to blend several images together to form a larger image. The images captured by the UAV during flight are pieced together to construct a mosaicked image from overlapping photographs.

The problem of mosaicking can be solved by addressing three main problems:

- Correct geometrical deformations
- Image registration
- Eliminate seams from image mosaics

Image registration can be emphasised in research by finding the point-to-point correspondence in different views of the same scene. Plane surface under general camera motion can be related by planar projective transformation, called homography. The homography matrix is 3x3 8 degree of freedom (DOF). This can ensure mosaicking of images captured by the camera rotating around its field of projection in [46].

More details about introductory information can be found in [7, 31] .

# Chapter 3

# Optimized Waypoints Selection

## 3.1 Coverage

### 3.1.1 Context of experiment

The main purpose of our work is to estimate the position of $n$ camera waypoints for surveying a given area in order to maximize the visual coverage. Each camera provides a top view of the area similar to UAV views. The coverage area of each camera is defined by the projection of the visual field onto the ground. This way, the mosaic composed of the captured images should be very close to a complete top view of the area. In order to find the best coverage, many experiments have been done to compare PSO and also GA. PSO is easier to implement and runs faster but GA is more flexible and generic thanks to the many tunable parameters. The following subsections will give an overview our method, which is based on GA, and provide a comparison between PSO and GA that demonstrates the overall advantage of the latter on the former.

### 3.1.2 Cost function

Since the goal is to maximize the visual coverage of the camera network, a cost function has been chosen to quantify it, as follows:

$$\sum_{i=1}^{n} = \frac{cover(i)}{size(grid)}_{1 \leq i \leq n}, \tag{3.1}$$

where $n$ is the number of waypoints; *grid* represents the discretization of the ground plane (floor); *cover*( ... ) is a function which computes the area on the ground which is covered by at

least one camera ; $size(...)$ is the dimension of the full area which must be $covered(grid)$.

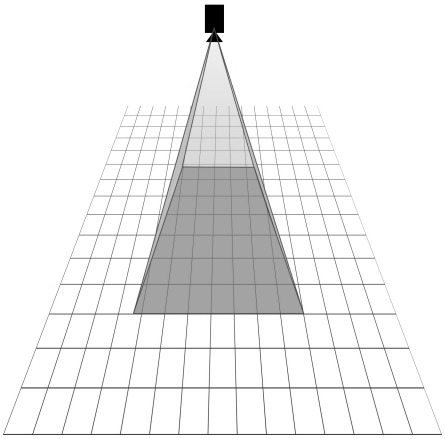


Figure 3.1: Projection of the camera on the ground.

Camera projection model is not explicitly taken into account, but the ground-projected visual field instead, as described in Fig.3.1.

### 3.1.3 Genetic algorithm

Motivated by Darwin's theory of evolution and the concept of survival of the fittest, GAs use processes analogous to genetic recombination and mutation to promote the evolution of a population that best satisfies a predefined goal [12]. Such kind of algorithms requires the definition of a genetic representation of the problem and of a cost function used to evaluate the solution. The candidate solution is represented by a data structure named chromosome, defined by Eq.(3.2) with $x, y$ and $z$ the cartesian coordinates and $\theta$ the rotation of the camera with respect to the optical axis: only two possible angles are allowed, 0° or 90° (portrait or landscape). To pass from an iteration (or generation) to the other a few steps are necessary (see Fig.3.2).

$$A = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ \theta_i \end{bmatrix}_{1 \leq i \leq n} \tag{3.2}$$

In our experiments, we empirically fixed the number of chromosomes to be 90, the mutation rate to be 0.001 and the crossover rate to be 0.919.

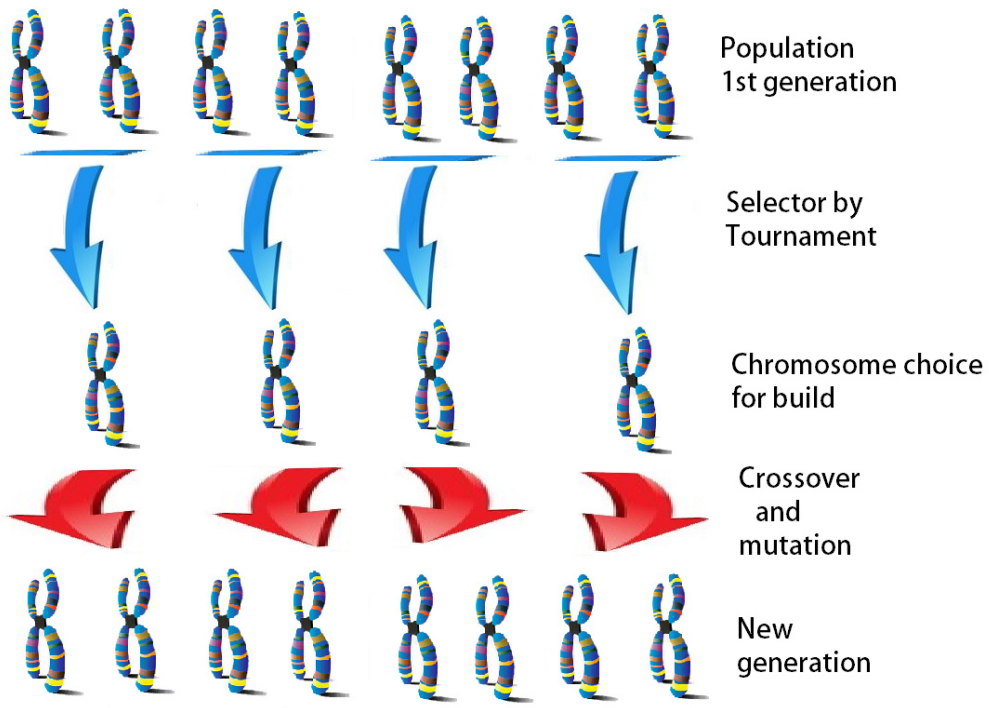



Figure 3.2: GA explanation, from one generation to the other.

### 3.1.4 Context of experimentation

In order to compare the two algorithms and evaluate their performances, we tested them on different scenarios depicted in Fig.3.3, with areas of different sizes and shapes, where:

- z is the height of the camera between (within the range $[1/z; z]$).
- Figure 3.3(a) is an area of size 120×80 (named Room).
- Figure 3.3(c) is an area of size 240×160 (named Big room)
- Figure 3.3(b) is an area of size 120×80 (named Room U)
- Figure 3.3(d) is an area of size 120×80 (named Room L)
- Figure 3.3(e) is an area of size 240×80 (named Big room L)

The design of experiments in Table 3.1 has been set up to identify the most efficient algorithm for the positioning of a set of waypoints with the maximum of coverage.

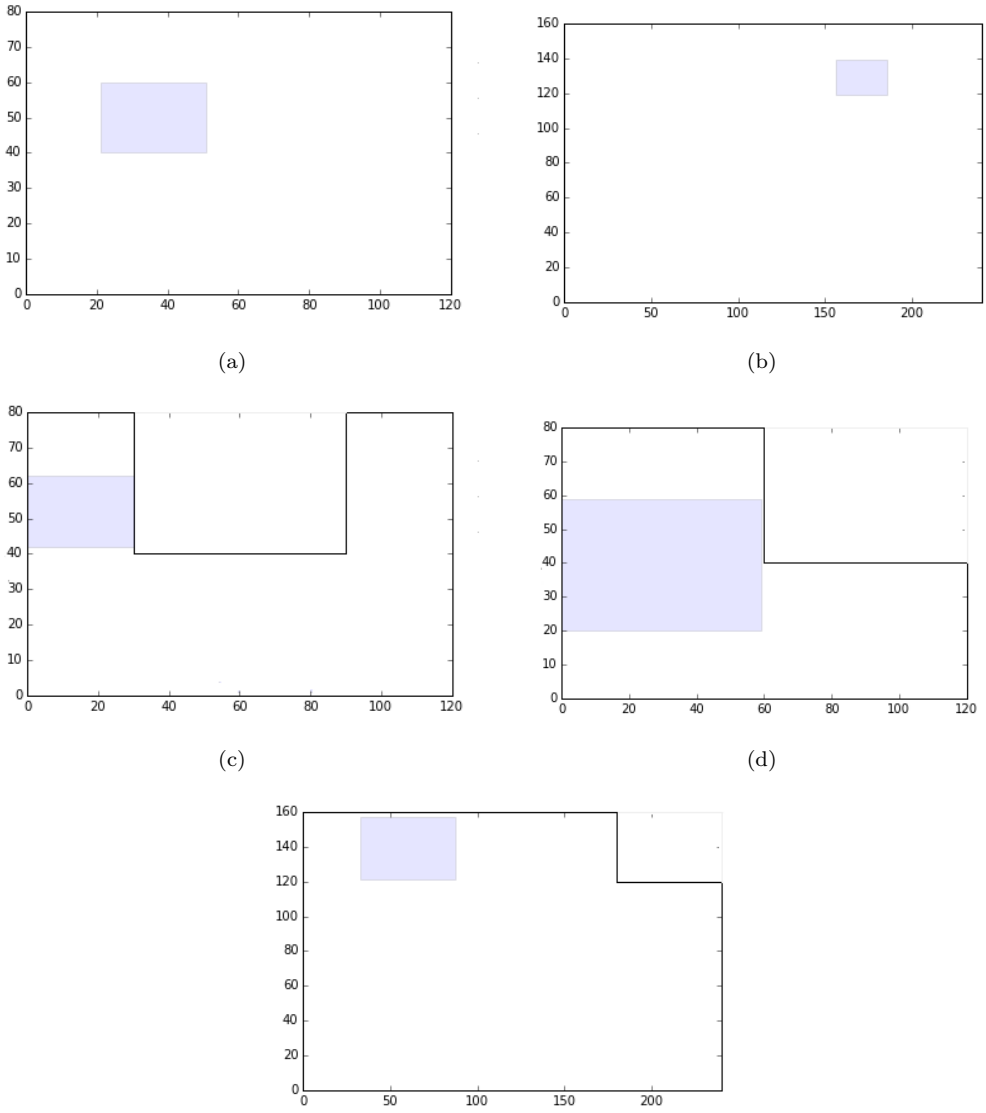



Figure 3.3: Scenarios used for the experiments: (a), (b), (c) are with z=1 and (d), (e) with z=2. The gray rectangle represents the field of view of one camera projected onto the ground.

The Ground Truth (GT) is the minimum number of waypoints required to fully cover a given area. The size of the area has been selected so that the GT can be easily estimated. NW is the maximum number of waypoints (or camera views) used for the experiments. At

| z=1 | | GA | | PSO | |
|---|---|---|---|---|---|
| | | GT | NW | GT | NW |
| Room | 120x80 | 16 | 20 | 16 | 20 |
| | 240x160 | 64 | 70 | 64 | 70 |
| Room U | 120x80 | 12 | 20 | 12 | 20 |
| z=2 | | GA | | PSO | |
| Room | 120x80 | 4 | 10 | 4 | 10 |
| | 240x160 | 16 | 20 | 16 | 20 |
| Room L | 120x80 | 3 | 10 | 3 | 10 |
| | 240x160 | 15 | 20 | 15 | 20 |

Table 3.1: Design of experiment for compare the efficiency of PSO and GA in different condition. (GT is Ground Truth and NW is Number of Waypoints).

each experiment a solution is computed for a number of waypoints from 1 to NW. In order to compare the different algorithms in similar conditions, only 10000 calls of the cost function is allowed for each set of waypoints.

### 3.1.5 Analysis of the result

After performing the design of experiments (see Table 3.1) it appears that GA and PSO algorithms are close in performance. In some cases GA is much more efficient (see in Fig.3.4) particularly in the case where the search space is large (big room and high number of cameras) as example in Fig.3.4). Instead, PSO is more effective to optimize small areas (see in Fig. 3.5 ). This efficiency can be explained by the small variation of the solution introduced by the PSO.

However, this small variation is not enough to find an optimized solution in a big search space which happens when a lot of cameras are required or when the local minima is deeper. Although the variety of solutions introduced by the GA allow escaping from local minima, it can affect negatively the accuracy of the solution, which may require a further optimization step to refine.

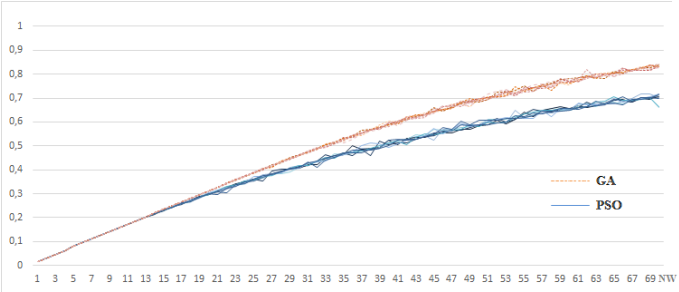



Figure 3.4: Comparison of eight solution given by GA, eight solution given by PSO algorithms with a Z equal to 1, in the big room 240x160 and ground truth equal to 64.

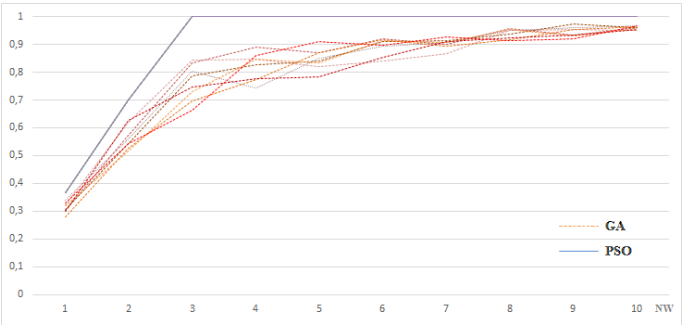



Figure 3.5: Comparison of eight solutions given by GA, eight solutions given by PSO algorithms with a Z between [1/2; 2], in the room with L shape 120x80 and ground truth equal to 15.

Following the comparison of the 2 algorithms, the GA seems more suited to find UAV waypoints especially if it navigates in a large room or outdoor scene. Furthermore, the comparative study demonstrated that the GA is less dependent on the shape of the covered area.

The number of UAV waypoints (camera pose) must be priorly known when using GA: the number of camera views is an input of the algorithm. However, in our application, it is far more interesting to assess the number of views with respect to the desired coverage rate of the area. To do so, a two-step procedure has been implemented. The first step is to find the minimum number of waypoints depending on the area to cover like formulated in the Eq.(3.3).

$$\frac{A_{room} - \sum_{i=1}^{n} A_{walli}}{A_{cam}} \times \text{Threshold Rate} = \text{NWayPoint} \tag{3.3}$$

- $A_{room}$ : area of the Room (length $\times$ width)

- $A_{Wall}$ : area of the obstacle like wall (length $\times$ width)

- $A_{Cam}$ : area cover by the camera in the maximum size of z

- NWayPoint : number of waypoints

- Threshold Rate : objective threshold rate

- $S$ : one solution of waypoints set

- $evalCost$ : cost function

The second step is to compute GA optimization until the threshold is reached, while increasing the number of waypoints like explained in the algorithm 1.

Algorithm 1:

```
1: procedure N(a)
2:     S ← 0
3:     while evalCost(S) ≤ ThresholdRate do
4:         S ← GA(NWayPoint)
5:         NWayPoint ← NWayPoint + 1
6:     return NWayPoint
```

# Chapter 4

# Optimized Path Planning

As mentioned earlier in the previous chapter, the best waypoints to be traversed guaranteeing area coverage were chosen. Path planning algorithm should be implemented afterward to sort these waypoints in order to assure minimal traversal 3D euclidean distance. After sorting the waypoints, there are several methods to achieve the trajectory planning that the robot should follow in between them.

These processes of sorting the waypoints, then traversing in-between them; can be formally explained as 2 layer path planning consisting of; global and local planners. Autonomous navigation of an aerial robot based on the combination of evolutionary algorithms as a global planner. Followed by one choice of the approaches discussed in section 4.2 as a local planner. These approaches can be summed up as a linear piecewise function, spline piecewise function and artificial potential fields.

Normally path planning optimization techniques does not put the constraint of passing by the defined waypoints as a hard constraint in their solutions. But rather, tend to change the waypoints depending on the robot motion in relative to the environment. In this thesis's case hard constrains of passing by the waypoins are taken into account. A global path planner is good in producing a minimal distance path, but poor in finding intermediate waypoints and reacting to unknown obstacle.

## 4.1 Global Path Planning

There are several ways to globally solve and find the robot path within a map.

Some operations research methods, such as traveling salesman problem (TSP), Chinese postman problem and rural postman problem can be considered. In this thesis; TSP was taken into account.

### 4.1.1 Problem Formulation

#### Search Based Shortest Path Problem

The goal of this category of methods is to find a continuous curve (no need to be smooth) in a configuration space that begins from the start node $x_{init}$ to the goal end node $x_{goal}$. It is mature enough method in research field with several algorithms like Dijkstra, A*, D*, and many more.

Representing the waypoints chosen previously as a weighted graph. The nodes of the graph are the waypoints, vertexes are an intermediate path between waypoints. The weights of the vertices are the Euclidean distances between waypoints. Dijkstra is then implemented to find the minimum cost path.

Iteratively, Dijkstra will start from the initial node and search within the connected nodes for the one with the least weight. Choose this node to be its next initial node and repeat the process till all the required nodes are traversed and ranked.

It is wide known of its slowness in the topological representation. The resolution and amount of waypoints considered as goals lead to very slow and nonoptimal convergence for passing by all these waypoints, so another formulation is required to be tested. The formulation discussed in the next section shows promising results.

#### Traveling Sales Man Problem

The robot traversing the waypoints can be formulated as traveling sales man problem(TSP). The TSP can be shortly explained as a salesman has to visit several cities (or road junctions). The goal is to find the salesman's route of minimum length with the constrain of passing by by all the cities only once. Mathematically modeling the problem as a complete graph with n vertices, the salesman will make a tour or hamiltonian cycle [9].

Sorting the path can be addressed as an optimization problem. Every node which is a point in space (3D position) is represented as a city and the euclidean distances between the cities are calculated and used as the optimization cost function. The path is organized based on the minimum length of traversing over all the waypoints. To find an optimized solution GA is used. The privilege of TSP problem formulation and solving it using GA over the other shortest path algorithms like Dijkstra, is that it provides global complete solution traversing all the waypoints not finding a path from a starting node to a goal node. The GA approach for multiple waypoint path planning is more clarified and discussed by Trevor *et al.* in [10].

GAs generally consist of three operators: selection, crossover, and mutation. The evaluation function for the N cities two-dimensional Euclidean TSP is the sum of Euclidean distances between every pair of cities in the tour.

That is:

$$Costfunction = \sum_{i=1}^{N} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

Where, $x_i$, $y_i$ are the coordinates of city i and $x_N$, $y_N$ equals $x_0$, $y_0$.
The algorithm can be explained in abstract as :

1. Create a group of many random tours in what is called a population. This algorithm uses a greedy initial population that gives preference to linking cities that are close to each other.

2. Pick two of the shortest tours which represent better parents in the population and combine them to make 2 new child tours. Hopefully, these children tour will be better than either parent.

3. A small percentage of the time, the child tours are mutated. Mutation of a tour is done by multiple ways like shuffling or swapping the route. Shuffling the route is done by randomly moving one of the cities from one point in the tour to another. Mutation is done to prevent all tours in the population from looking identical.

4. Crossover is done by picking two parents. Then take subset from the first parent and complete the child formation from the second parent.

5. The new children tours are inserted into the population replacing a number of the longest tours. The size of the population remains the same in every generation.

6. New children tours are repeatedly created until the desired goal is reached.

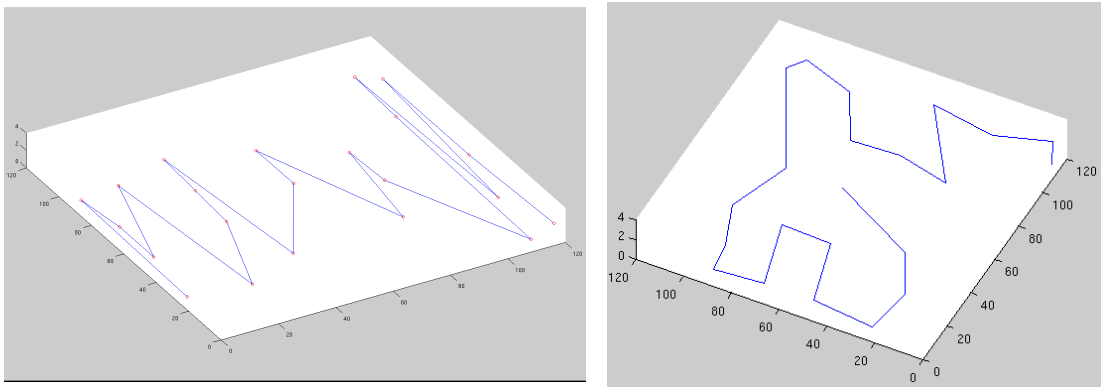



Figure 4.1: Path using Dijkstra vs GA

The distance covered by a path that is planned by GA is 513 meters which is shorter by the factor of 1.8 compared to the distance covered by Dijsktra multi-goal approach which is 963 meters.

## 4.2 Local Path Planning

After succeeding in getting the main waypoints sorted in a way to grantee shortest path length. Forthwith comes the issue of generating the mid waypoints that the robot should traverse to generate the trajectory.

Three main ways have been implemented and tested. These ways will be mentioned in the next three subsections.

### 4.2.1 Linear Piecewise Interpolation

The previous process will provide a list of sorted waypoints co-ordinates. These waypoints are dealt as graph nodes, that need to be connected to have a path, so the problem is formulated as a linear or spline piecewise function.

Hereby the used linear piecewise function used in Eq.(4.1).

$$f(x) = \begin{cases} 0 & \text{if} \quad x \leq a \\ \frac{x-a}{b-a} & \text{if} \quad a \leq x \leq b \\ \frac{c-x}{c-b} & \text{if} \quad b \leq x \leq c \\ 1 & \text{if} \quad c \leq x \end{cases}, \tag{4.1}$$

Where the line segment between points $a$ and $b$ will be the line between every waypoint and the consecutive one. The third point is $c$ and depending on the number of waypoints variables will be added in the formula. Discretizing this point up to an extent will act as the sampling the path.

### 4.2.2 Spline Piecewise interpolation

The goal of the spline interpolation is to find the shortest smooth path through consecutive waypoints. This smoothness is advisable in order to eliminate the sudden change in the first derivative or the slope of the function (the line between the waypoints). It is very important as, robots specially aerial ones, require a portion of time and space to change direction and rotate.

When the interpolation is assumed to be piecewise linear, it is relatively easy. However, if the curve is to be a spline, perhaps interpolated as a function of chordal arc length between the points, this gets a bit more difficult. A nice trick is to formulate the problem in terms of

differential equations that describe the path along the curve. Then the interpolation can be done using an ordinary differential equation solver. For instance, the robot used in this thesis is a quadcopter; which is considered to be holonomic. Its kinematics give hovering capabilities. This feature permits to relax turning constraints on the path (which represents a crucial problem for fixed-wing vehicles). The implemented interpolation as shown in the results section 5 is a cubic spline. This cubic spline of the $i^{th}$ spline function, in general, can be written as:

$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \tag{4.2}$$

For n data points, there are n-1 intervals and thus 4(n-1) unknowns to evaluate to solve all the spline function coefficients. $a_i$, $b_i$, $c_i$, and $d_i$ must be determined for each $i$. More investigation about spline interpolated path planning can be found in [21].

### 4.2.3 Artificial Potential Field

It was promising idea when Artificial Potential Field (APF) was first introduced in the robotics field by Oussama Khatib in 1986 [23]. With simple rules defined in the paper as a potential function between free space and obstacle space. The goal position to be reached is an attractive pole for the robot and obstacles are repulsive surfaces.

The procedure involves the following steps:

1. Establish a potential field function.
2. Locate a minimum potential point using any optimization algorithm.
3. Navigate an object toward the minimum potential point.
4. Repeat steps 2 and 3 until the object reaches the goal position.

$$\begin{aligned} U(q) &= U_{att}(q) + U_{rep}(q) \\ U_{rep}(q) &= \sum (U_{obstacles}(q)) \end{aligned} \tag{4.3}$$

$$F(q) = -\nabla U(q) \tag{4.4}$$

Where $U_{att}$ is the attractive potential and $U_{rep}$ is the repulsive potential. F(q) is the local force that will let the robot move by small steps. There are various functions to model the potential field. The parabolic one is used in this context. Following an iterative gradient decent can be a simple way to go to the bottom where the goal point is available.

$$q_{i+1} = q_i + \delta_i \frac{F(q)}{\|F(q)\|}$$

As obstacle avoidance was the original goal of this algorithm, potential fields come with the risk of getting stuck in local minima and not being able to reach the goal. There are several solutions to this inherent characteristic. one of them is to methods like RRT to plan the path in the generated potential field map like what stated by Latombe et al. in [26,27]. In our specific case, the global path planner set the initial point and the goal point provided to the APF with a guarantee to have a short distance which should not lead to a local minimum. Here APF is tweaked to only avoid the obstacles that may appear in the map, not planning the whole path.

## 4.3 Summary

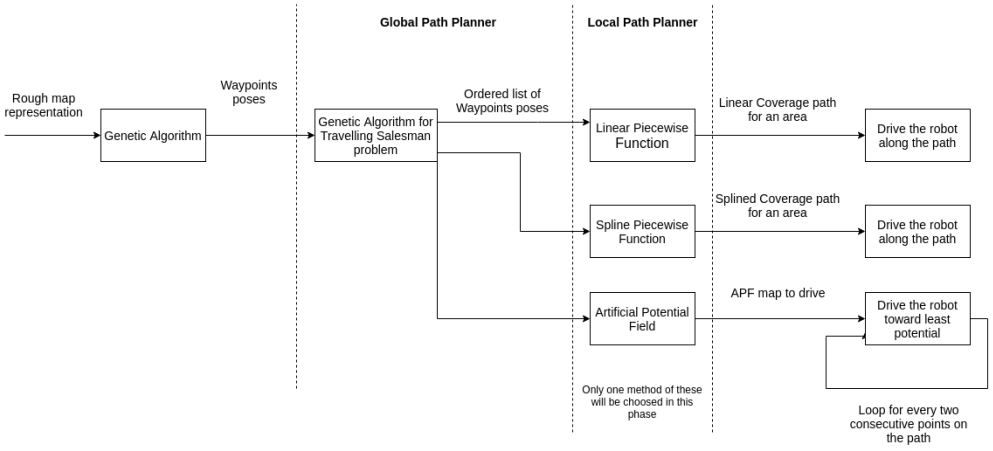



Figure 4.2: Pipeline of the methodology

# Chapter 5

# Experiments and Results

As mentioned earlier, the work presented in this thesis is prepared on simulation and validated on real micro UAV robot. More details about the environment of the simulation is found in 5.2, and about the real robot in 5.3. An abstraction for the real robot's hardware and software specifications are mentioned in 5.3.1. Robotics Operating System (ROS) has been used because of its advantages that will be discussed in section 5.1.

## 5.1 Robotics Operating System

In the past years ROS has proved its value and grown great attention in both research and industry since it is first introduced by Andrew Ng. et al. in 2009 [36]. It is an open source message oriented middleware with publisher subscriber message passing process. Collaborators from many companies and research laboratories develop packages and tools with the concept of being reusable.

There are now mature enough tutorials, that can give extensive and deep view of ROS like for example technical books that show step by step development using ROS as in [20, 33]. It is chosen and preferred because it can be connected to various simulators and hardware.

## 5.2 Simulation

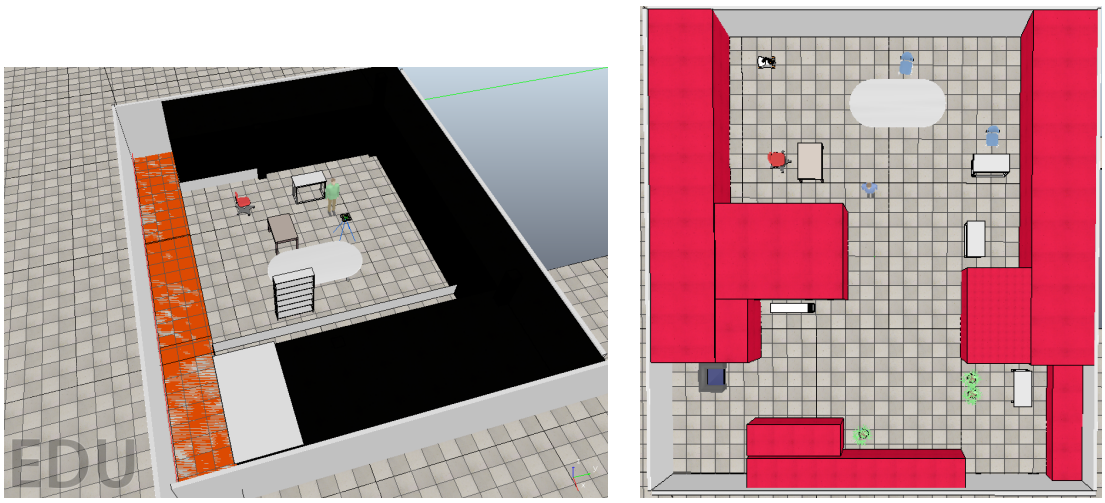
There are variety of robotics simulations available in the field nowadays like MORSE, Gazebo, V-REP,Stage, etc. Available comparison between them can be found examined by Gonalo Nuno in [2].

### 5.2.1 VREP

Virtual Robot Experimentation Platform (V-REP) is introduced in 2013 by E. Rohmer et al. in [15]. It has great advantage in speeding up the process of algorithm's development visualization and validation. The robot simulator V-REP, model can be individually controlled via an embedded script, a plugin, a ROS node, a remote API client, or a custom solution. This makes V-REP very versatile and ideal for multi-robot applications. Controllers can be written in C/C++, Python, Matlab, and more.

#### Room representation

Several rooms were built in V-REP to act as different room scenarios. In the figure 5.15(a) the black and orange parts are not areas of interest to cover. while in figure 5.15(b) the red blocks are the areas of least interest.



(a) Small Room (b) Big room

Figure 5.1: Room Coverage

The size of the big room is 15*14 $meters^2$ .

### 5.2.2 UAV Simulation

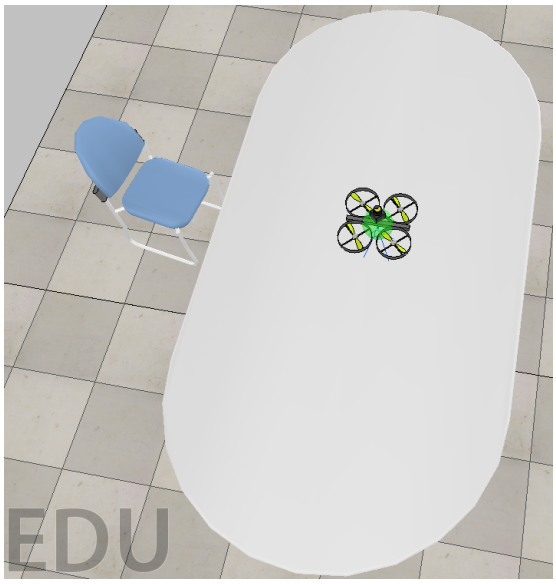


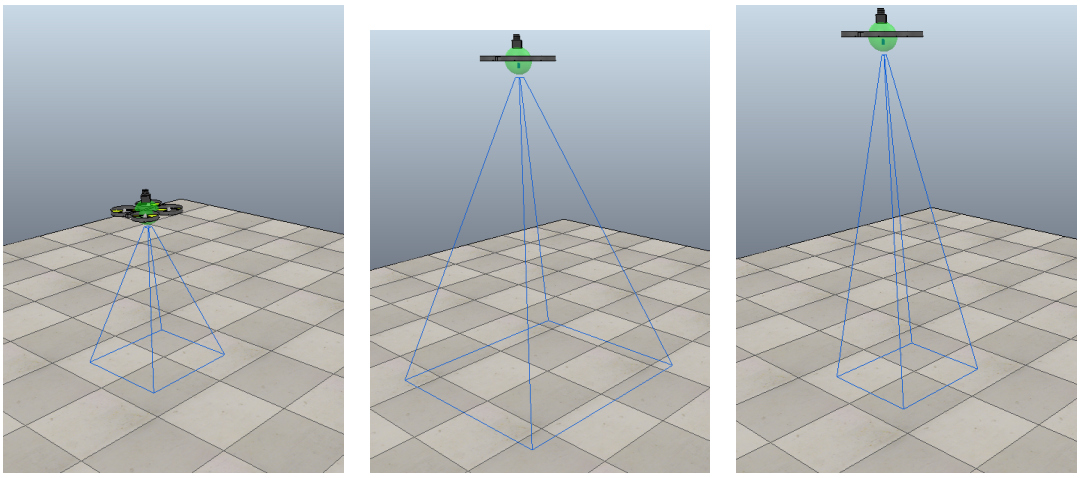

Figure 5.2: Drone available in V-REP with some modifications

#### UAV model

The UAV used either in simulation or real world experimentation is of micro aerial quadcopter drone like in figure 5.2. Obviously from the name comes quad which is implying to have 4 blade rotors. The quadcopter is controlled by adjusting the angular velocities of the rotors which are spun by electric motors. It has simpler structure compared to the other rotor crafts like helicopter. It can easily take off and land with vertical take-off and landing (VTOL). Some modifications is done on the provided model by V-REP like adding LIDAR sensor over the robot. So that in the future work, dynamic obstacle avoidance will be taken into consideration. Also a proportional-integrator-differentiator (PID) control is provided for the motors, in order to have more precise motions.

#### Mounted Camera

There are two cameras mounted on the quadcopter available with V-REP, unfortunately their images can not be transmitted over the package used in ROS to transport and process images. For this purpose editing the available model by mounting what is called according to V-REP, vision sensor. This vision sensor is giving many properties to be edited like clipping plane distance, perspective angle, resolution and more.



(a) 1m height, wide angle (b) 2m height, wide angle (c) 2m height, small angle

The previous figure is showing the change in the field of view and its effect on the captured image. This is also affecting the number of waypoints as mentioned before.

Figure 5.3: Camera projection on the ground

### 5.2.3 Simulation Results

In this subsection the results generated from GA solving the TSP followed by either the linear, or spline piecewise function. Afterwards the artificial potential field results are shown.

Results generated by using 18 waypoints, providing 79% of area coverage.

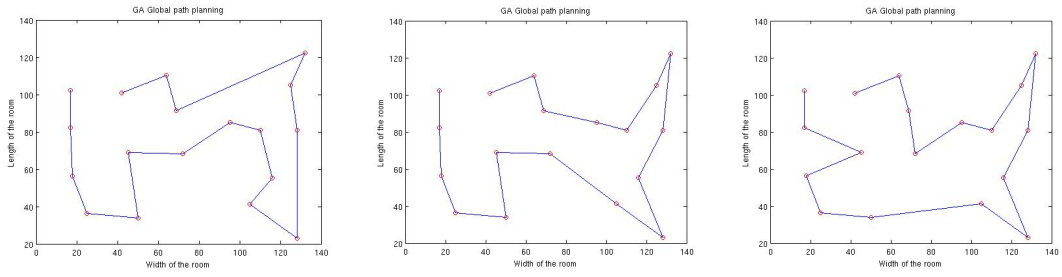



Figure 5.4: GA path planning followed by linear piecewise of 18 waypoints to cover 79% of an area

The results shown in the previous paths are generated with the same parameters specification of the GA with letting the starting node to be anywhere inside the arena. The left path length is 486 meters, the middle path is 463.4714 meters, while the right one is 474 meters for the same

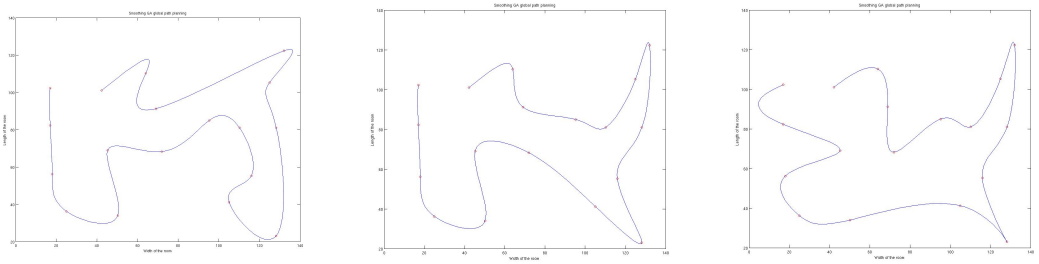


Figure 5.5: Smoothing GA path planning of 18 waypoints to cover 79% of an area

area coverage percentage. There are variations of results because GA acquires randomness in finding the solution. This randomness appears in the evolution of the populations ( proposed tours in the iterations) of the program.

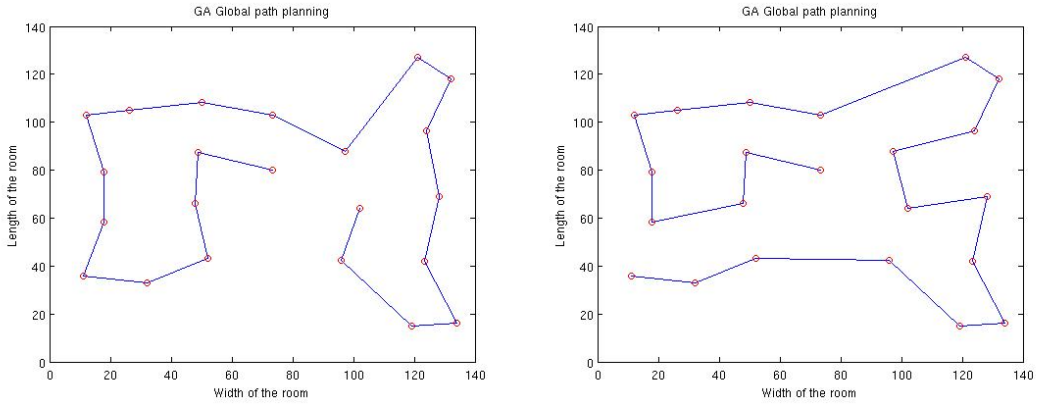



Figure 5.6: GA path planning followed by linear piecewise of 22 waypoints to cover 90.68% of an area

The left path length is 511 meters, while the right one is 547 meters for the same area coverage percentage.

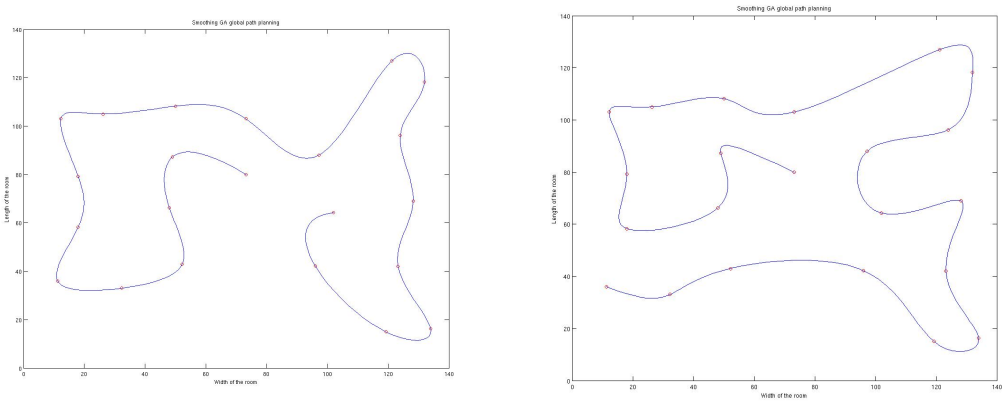



Figure 5.7: Smoothing GA path planning of 22 waypoints to cover 90.68% of an area

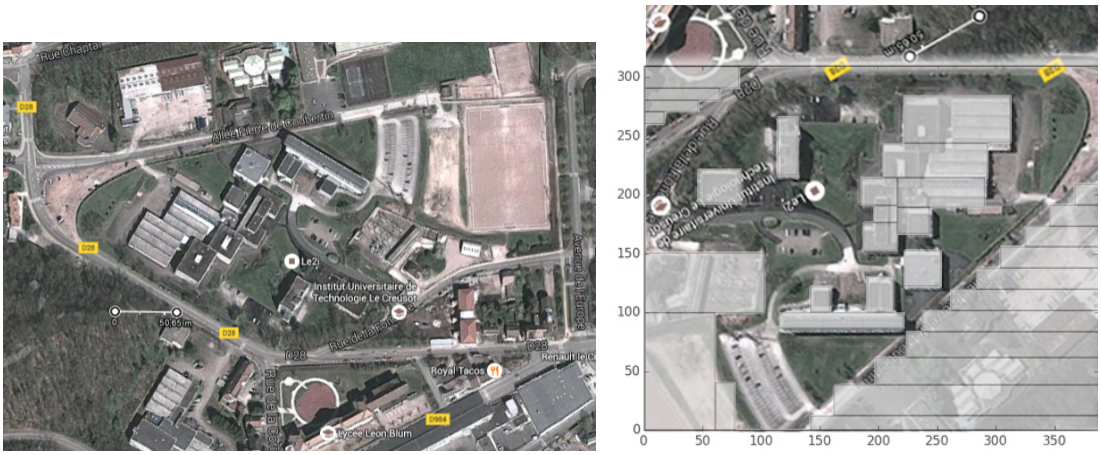
The time elapsed to generate these paths range from 0.475 seconds to 0.507 seconds on a computer with specifications discussed in subsection 5.5

| Parameter | Corresponding Values | |
|---|---|---|
| Area Coverage | 79% | 90.68% |
| Number of waypoints | 18 | 22 |
| Distance in meters | 486 | 511 |
| | 463.47 | 547 |
| | 463 | |
| | 474 | |

Table 5.1: Different results of the path generated by GA

### 5.2.4 Outdoor Simulation Data

Aerial map of the laboratoire electronique,informatique et image (LE2I) and its surroundings is captured. Following to the capture step, comes the discretization of the area. Also blocks covers the areas that are not required to be passed by.




(a) Aerial image as seen on map

(b) unwanted areas to be covered are contoured with rectangles

Figure 5.8: Aerial Arena

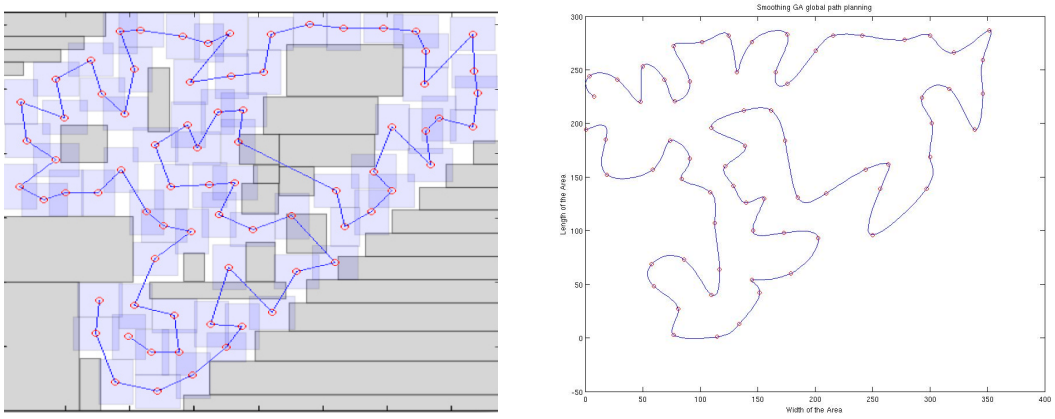



Figure 5.9: Imitated area coverage by aerial robot

Just a reminder that passing over the rectangular blocks is not a forbidden constraint. This area has no interest to be covered, but there is no harm for the robot to pass over it.

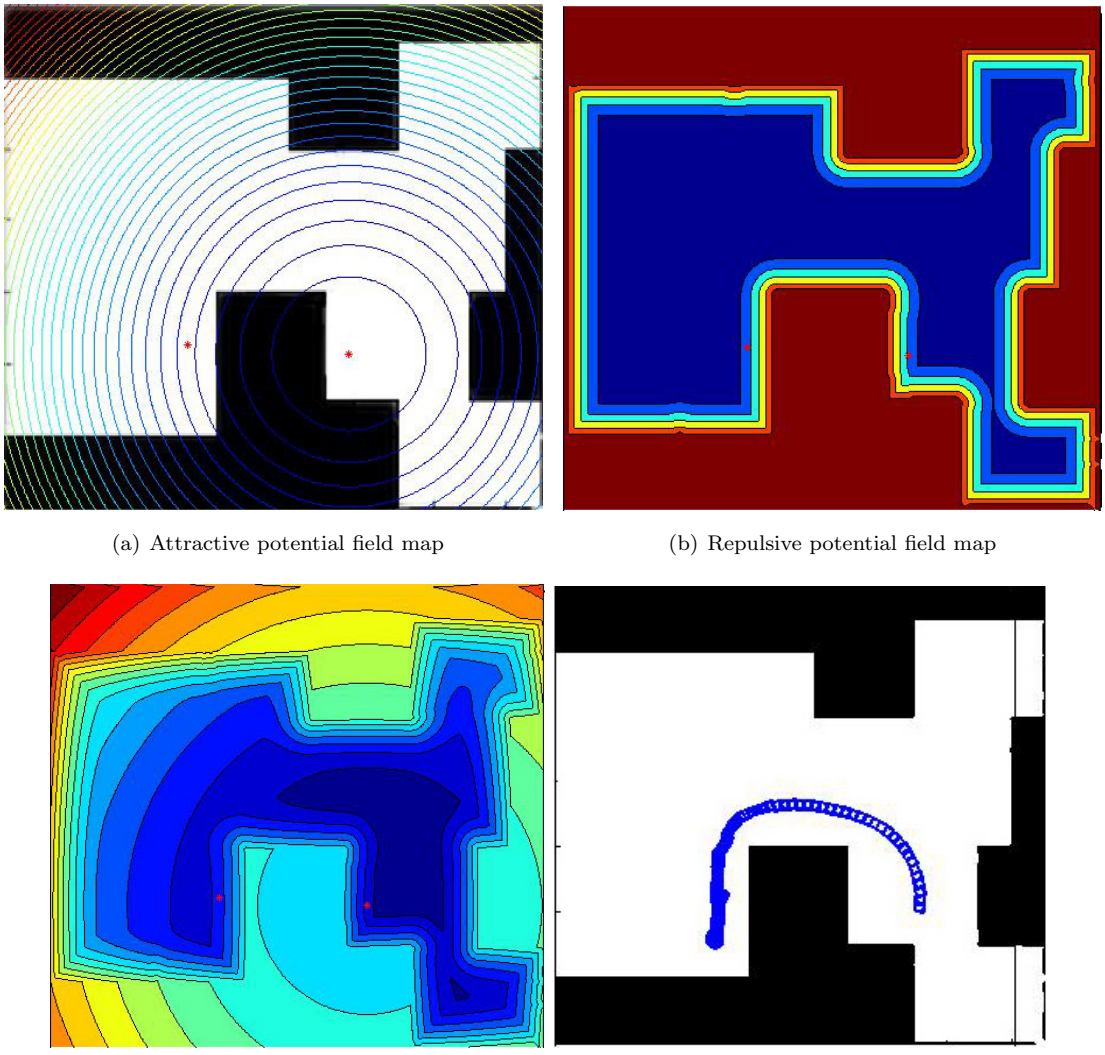
### 5.2.5 Artificial Potential Field

In case of, passing by the unwanted areas became a constraint. Artificial potential field can be a good solution. It is also offering very good results to avoid any appearing obstacles, but with

the limitation of falling in local minima.

The context of this experiment is done by picking the waypoint and its consecutive one. Then generate a potential field map and let the robot navigate to it.

The points shown in this experiment are the ones that the linear and spline piecewise shows cutting over the unwanted area to be covered. This solution comes with the price of slightly increasing the length of the coverage path.



(a) Attractive potential field map

(b) Repulsive potential field map

(c) Summary potential field map

(d) Path from one point to the consecutive

Figure 5.10: Potential field map

## 5.3 Real World Experiments

### 5.3.1 Quadcopter Hardware

The Quadcoptor used to do this experiments is an offshelf commercially found drone from the French company Parrot which is AR.drone 2.0. It was first introduced in 2010 as a toy for Augmented Reality games. Shortly research labs, universities make use of its comparably cheap and low weight in their experiments. Hardware and software specification with sample research usage can be found in [5, 24]. Equipped with two cameras; one of them is pointing downwards makes it clearly very useful for the application of area coverage. This camera runs at 60 fps with a resolution of 176 x 144 pixels, covering a field of view of only 47.5° x 36.5°. Concerning the software, there are mobile applications to control the drone for both IOS and Android. SDK is avilable for computer development. The embedded software that is running onboard is not directly accessible, but sending and transmitting data with telnet shell communication channel is possible. Transmitting navigation data and receiving the sensor readings to the drone is done after connecting a workstation computer to it through ad-hoc wireless LAN network. External ROS packages used to control the quadcopter will be discussed in more details in the next subsections.

Figure 5.11: Ar.Drone 2.0

### 5.3.2 Experiment Environment

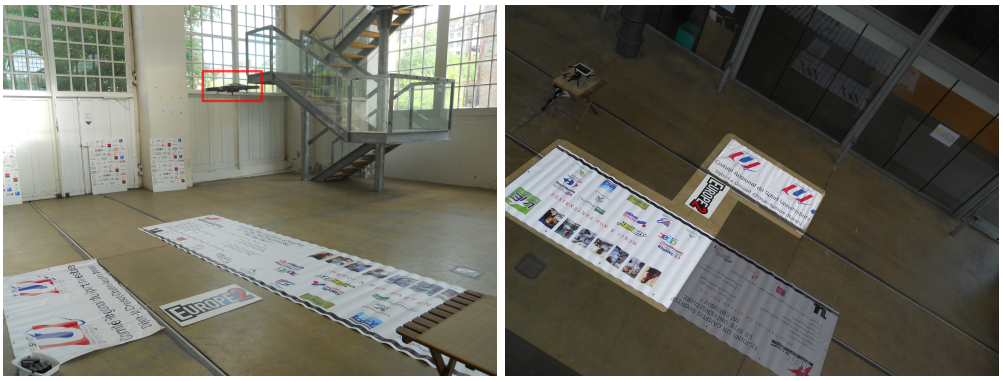
As mentioned in 2.3 the wide scenarios of generating real experiments with aerial robots, localization problem is crucial. Using the monocular SLAM package developed for AR.Drone by Engel et al. in [13, 14] is chosen to solve this challenge. This package proved by practical experimentation its reliability in localizing the robot without any environment modification except putting a panel full of figures to give rich features. Flying arena overview is shown in figure 5.12 .

### 5.3.3 Robot Localization

The tum_ardrone ROS package enables the drone to localize itself and navigate in an unknown and potentially GPS-denied environments without learning the environment or mapping it previously. Monocular SLAM to compute a visual pose estimate within this map for each video frame. The main contribution of this package is to estimate the scale of map from inertial and altitude measurements by formulating the problem statistically, and deriving a closed-form solution for the maximum likelihood (ML) estimator of the unknown scaling factor. Proportional-integral-differential (PID) control is applied to control the pose of the drone, and fly to and hold a desired target position [14].

### 5.3.4 Practical Results

A plug-in to the tum_ardrone package is developed for the practical work. This plug-in takes as inputs the waypoints from the path planning module and also the relative position of the drone. The relative position is of the drone current position to the zero position on the map used to generate the path. The plug-in calculates the mid path in between the main waypoints. It is also responsible for toggling the camera of the drone when it reach the waypoints. Then it gives the order of image acquisition.



(a) Drone and flag to be covered (b) Top view of the flying arena

Figure 5.12: Workspace for the experiment

The highlighted area in figure 5.12 (b) is the area needed to be covered by the robot. Then mosaicking techniques are applied to construct the final image output.

The Path generated from a simulated area that imitate the desired part wanted to be covered.

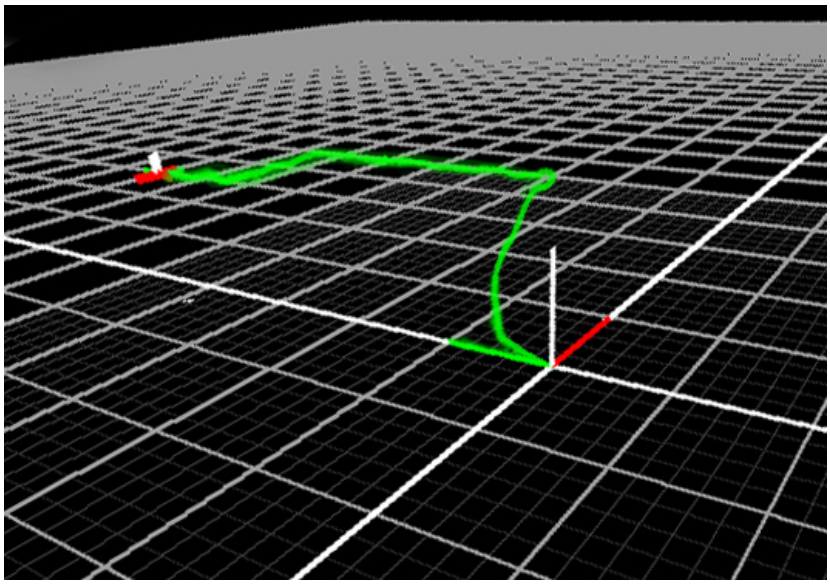


Figure 5.13: Path Planning visualization on PTAM

A video showing the robot navigation, path execution, frontal camera scene and mosaicking generation; can be found in this link [31].

## 5.4 Mosaicking

### 5.4.1 Simulation

The ortho-mosaicked output image composed by the captured images is very close to a complete top view of the desired area to be covered. Mosaicking in our case is simply computed using the position and orientation coordinates of the known waypoint. Then stitch the images together to validate the optimum choice of the positions where the drone captured these images as shown in Fig.5.15 for the simulated room in V-REP.

Otho-mosaicking also proved its validity in mosaicking aerial outdoor images as mentioned by Yahyanejad in [37].

The coverage threshold rate is 90% of the useful area in the room without taking the red block parts into consideration to cover.

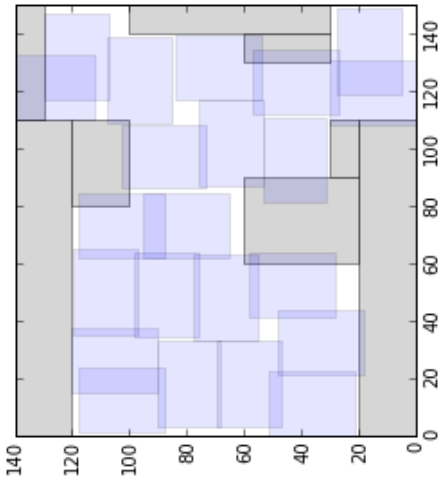
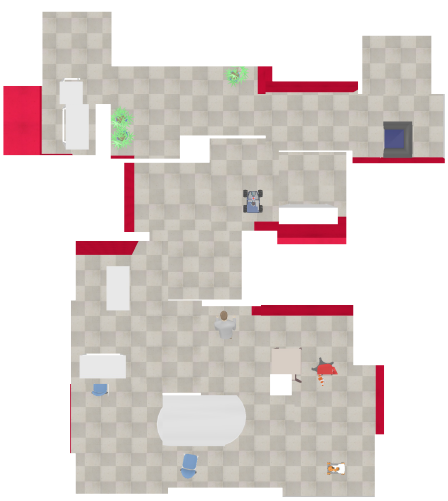
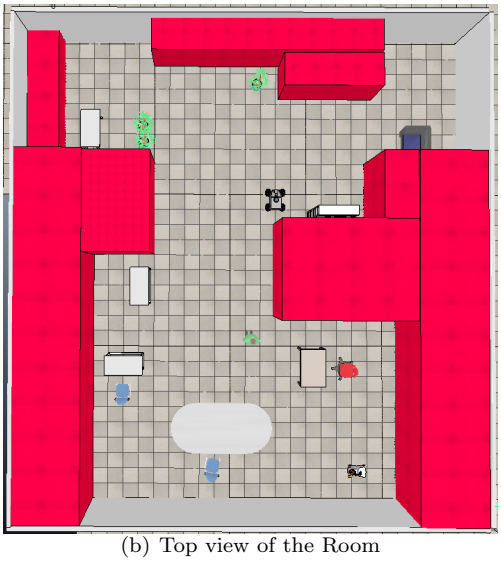



Figure 5.14: Poses of every image captured in the room



(a) Room mosaicing



(b) Top view of the Room

Figure 5.15: Room Coverage

### 5.4.2 Practical

The mosaicking process of the practical experiment is done after the flight. The acquired images used to generate the figure 5.16 are 32 images. The several image on the same waypoint is shoot, so that the images with best correspondences will be used. This result is not by the mosaic

developed by the author for the simulation. Indeed it is done by a well known software online; provided by Microsoft Research for free of use. It is called Image Composite Editor.

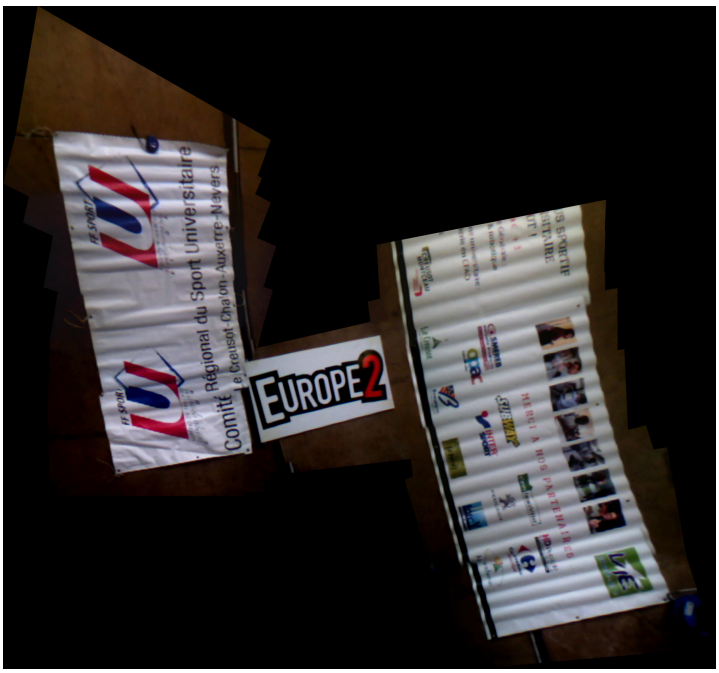



Figure 5.16: Flags Mosiacing

## 5.5 Computation

The computer used to generate these experiments is specified with 8 GB of memory (RAM). Processor is Intel Core i7-2720QM CPU @ 2.20GHz 8. The OS is ubuntu 14.04 LTS. Codes developed is developed by C++11, python 2, Matlab 2013B. V-REP version is 3.3.x. The ROS package is tested on ROS distribution Indigo.

# Chapter 6

# Conclusion

## 6.1 Conclusion

In this thesis, the coverage path planning is addressed with a new approach. This approach is dealing with area coverage problem as two different challenges. The first challenge is the optimized choice of poses that guarantee maximum area coverage. The second challenge is path planning of these chosen poses.

This thesis is considering nonregular areas. Evolutionary algorithms specifically genetic algorithm (GA) is used and compared with other methods like particle swarm algorithm to solve the first problem of area coverage. GA approach efficiently cover 90% of a given area.

Several path planning approaches were tested. The design of having the multi layer of path planning is taken into account. Three algorithms were implemented,tested and compared results were presented. In all the three algorithms, there are two layers of path planning. The first layer is formulating the robot poses as cities and connect them to form a graph which resembles the travelling sales man problem (TSP). GA is used to solve TSP and generate a list of poses of the cities that will insure least length of tour.

Then the second layer is different in the three algorithms. The first algorithm, takes the list of ranked poses and generate linear piecewise function linking all the poses. The second algorithm is using spline piecewise function instead of linear which generate smoother paths. Last but not least, the third approach, uses artificial potential field (APF) as the second layer of map representation and path planning. It is used mainly to avoid obstacles that appear on the map. Static obstacles are considered in this thesis. Efficiently traversing the map without the known falling in the trap of local minima.

One of the drawbacks of APF is the vast amount of parameter tuning needed before having

the efficient results. This tuning is dependent on many factors like; a scaling factor of both the attractive and repulsive potentials, the current heading of the robot. It is also dependent on the shape of the map. The initial and final goal location being traversed are influencing the potential field too.

Some of the work presented in this thesis lead to a publication in URAI 2016 and will be presented in China on August 2016.

## 6.2 Future Work

- Impose dynamic obstacles in the map to validate the artificial potential field.
- Re-planning algorithms in real time can be tested.
- RRT can be a good alternative to being tried instead of the artificial potential field.
- The down camera of AR.Drone 2.0 is of low resolution. So thinking of attaching a camera to the drone and testing it will be of great importance for better results.

# Bibliography


[1] Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5):753–782, 1998.

[2] Gonçalo Nuno dos Santos Augusto. Robotteamsim–3d visualization of cooperative mobile robot missions in gazebo virtual environment. 2013.

[3] Song B., Soto C., Roy-Chowdhury A. K., and Farrell J. Decentralized camera network control using game theory. *In Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on (pp. 1-8)*, 2008.

[4] Alessandro Benini, Adriano Mancini, and Sauro Longhi. An imu/uwb/vision-based extended kalman filter for mini-uav localization in indoor environment using 802.15. 4a wireless sensor network. *Journal of Intelligent & Robotic Systems*, 70(1-4):461–476, 2013.

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

[6] Scott A. Bortoff. Path planning for uavs. *American Control Conference Proceedings of the 2000. Vol. 1. No. 6. IEEE*, 2000.

[7] David Capel. *Image mosaicing and super-resolution*. Springer Science & Business Media, 2004.

[8] Howie Choset. Coverage for roboticsa survey of recent results. annals of mathematics and artificial intelligence 31.1-4. pages 113–126, (2001).

[9] George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410, 1954.

[10] Trevor Davies and Amor Jnifene. Multiple waypoint path planning for a mobile robot using genetic algorithms. In *Computational Intelligence for Measurement Systems and Applications, Proceedings of 2006 IEEE International Conference on*, pages 21–26. IEEE, 2006.

[11] Kalyanmoy. Deb. *Multi-objective optimization using evolutionary algorithms. Vol. 16.* John Wiley & Sons, 2001.

[12] Packer E. Computing multiple watchman routes. *In Experimental Algorithms (pp. 114-128). Springer Berlin Heidelberg*, 2008.

[13] J. Engel, J. Sturm, and D. Cremers. Camera-based navigation of a low-cost quadrocopter. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[14] J. Engel, J. Sturm, and D. Cremers. Scale-aware navigation of a low-cost quadrocopter with a monocular camera. *Robotics and Autonomous Systems (RAS)*, 62(11):1646–1656, 2014.

[15] Rohmer Eric, Singh Surya PN, and Freese Marc. V-rep: A versatile and scalable robot simulation framework. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1321–1326. IEEE, 2013.

[16] WANG Fei, CUI Jin-Qiang, CHEN Ben-Mei, and H LEE Tong. A comprehensive uav indoor navigation system based on vision optical flow and laser fastslam. *Acta Automatica Sinica*, 39(11):1889–1899, 2013.

[17] Marc Carreras Galceran Enric. A survey on coverage path planning for robotics. robotics and autonomous systems 61.12. pages 1258–1276, 2013.

[18] Ma H., Yang M., Li D., Hong Y., and Chen W. Minimum camera barrier coverage in wireless camera sensor networks. *In INFOCOM, 2012 Proceedings IEEE*, pages 217–225, 2012.

[19] Zhao J., Cheung S. C., and Nguyen T. Optimal camera network configurations for visual tagging. *Selected Topics in Signal Processing, IEEE Journal of, 2(4)*, pages 464–479, 2008.

[20] Lentin Joseph. *Mastering ROS for robotics programming.* Packt Publishing Ltd, 2015.

[21] Kevin B Judd and Timothy W McLain. Spline based path planning for unmanned air vehicles. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–9. Montreal, Canada, 2001.

[22] Reddy K. K. and Conci N. Camera positioning for global and local coverage optimization. *In Distributed Smart Cameras (ICDSC), 2012 Sixth International Conference*, 2012.

[23] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.

[24] Tomáš Krajník, Vojtěch Vonásek, Daniel Fišer, and Jan Faigl. Ar-drone as a platform for robotic research and education. In *Research and Education in Robotics-EUROBOT 2011*, pages 172–186. Springer, 2011.

[25] Liu L., X. Zhang, and Ma H. Optimal node selection for target localization in wireless camera sensor networks. *Vehicular Technology, IEEE Transactions on, 59(7)*, pages 3562–3576, 2010.

[26] Jean-Claude Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.

[27] Steven M. LaValle. *Planning algorithms*. Cambridge university press, 2006.

[28] Meier Lorenz, Tanskanen Petri, Heng Lionel, Lee Gim Hee, Fraundorfer Friedrich, and Pollefeys Marc. Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, 33(1-2):21–39, 2012.

[29] Erdem U. M. and Sclaroff S. Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. *J Computer Vision and Image Understanding*, pages 156–169, 2006.

[30] Moeini M., Krller A., and Schmidt C. Une nouvelle approche pour la resolution du probleme de la galerie dart. 1998.

[31] Mark Bastourous. video of the practical experiment. `https://onedrive.live.com/redir?resid=7D510DC6427EBCFD!2442&authkey=!ADaFBcw-AiQErOI&ithint=folder%2cwmv`, 2016. Online; accessed 29-May-2016.

[32] Michael Nathan, Mellinger Daniel, Lindsey Quentin, and Kumar Vijay. The grasp multiple micro-uav testbed. *Robotics & Automation Magazine, IEEE*, 17(3):56–65, 2010.

[33] Jason M O'Kane. A gentle introduction to ros, 2014.

[34] Zhou P. and C. Long. Optimal coverage of camera networks using pso algorithm. *In Image and Signal Processing (CISP), 2011 4th International Congress on*, 4:2084–2088, 2011.

[35] Wang Q., J. Wu, and C. Long. On-line configuration of large scale surveillance networks using mobile smart camera. *In Distributed Smart Cameras (ICDSC), 2013 Seventh International Conference on*, pages 771–779, 2013.

[36] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5, 2009.

[37] Yahyanejad Saeed, Wischounig-Strucl Daniel, Quaritsch Markus, and Rin Bernhard. Incremental mosaicking of images from autonomous, small-scale uavs. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 329–336. IEEE, 2010.

[38] Flemming Schøler, La Cour-Harbo, Morten Bisgaard, et al. Generating approximative minimum length paths in 3d for uavs. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 229–233. IEEE, 2012.

[39] Dan Simon. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

[40] Kaarthik Sundar and Sivakumar Rathinam. Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots. *Automation Science and Engineering, IEEE Transactions on*, 11(1):287–294, 2014.

[41] Jean-Philippe Tardif, Yanis Pavlidis, and Kostas Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2531–2538. IEEE, 2008.

[42] Jing Tian, Lincheng Shen, and Yanxing Zheng. Genetic algorithm based approach for multi-uav cooperative reconnaissance mission planning problem. In *Foundations of Intelligent Systems*, pages 101–110. Springer, 2006.

[43] Piegl LA Valavanis K, Oh P. *Unmanned aircraft systems.* Springer, New York, 2008.

[44] Chin W. and Ntafos S. Optimum watchman routes. inform. process. lett. 1988.

[45] Gabriely Yoav and Elon Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence 31.1-4*, pages 77–98, 2001.

[46] Jiang Yu Zheng and Saburo Tsuji. Panoramic representation for route recognition by a mobile robot. *International Journal of Computer Vision*, 9(1):55–76, 1992.