Obstacle avoiding and path planning for UAV

Session 2013-2017

Author's

Registration No	Name
BScs-14201	Hasan Irtaza Mirza
BScs-14216	Muhammad Faisal
BScs-14217	Hafiz Awais Ahmad
BScs-14247	Izrar Ahmad

Supervisor: Miss Amna Khan

Submission Date: **23 October 2017**

Computer Science & Information Technology Department

Superior University (Lahore)

Project Registration

Project ID (for office use)						
Type (Nature of Project)	[ ] Development [ ] Research **[ ] R&D**					
Project Group Members						
Sr.#	Roll #	Student Name	CGPA	Email ID	Phone #	Signature
I	14201	Hasan Irtaza Mirza	3.3	Hasanirtaza@gmail.com	+923361404455	
II	14216	Muhammad Faisal	2.1	Ch.m.faisal990@gmail.com	+923377035055	
III	14217	Hafiz Awais Ahmad	2.4	Hafizawais.rajput@gmail.com	+923074320022	
IV	14247	Izrar Ahmad	2.5	Arshad11111arshad@gmail.com	+923354170310	
Name & signature of advisor (if student are eligible for FYP)						

Plagiarism Free Certificate

This is to certify that, I am Muhammad Faisal S/D/o Muhammad Shahid, group leader of FYP under registration no BSCS-14216 at Computer Science Department & Information Technology, Superior University, Lahore. I declare that my FYP proposal is checked by my supervisor and the similarity index is _____% that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix A.

Signature: _____

I _____ show my consent to supervise the project titled _____

_____

which consists of above listed students as group members

Date & Signature: _____

Designation: _____

Approval of FYP Committee

Committee Member 1: Name: _____ Designation: _____

[ ] Accepted [ ] * Deferred [ ] *Rejected Signature: _____

*Remarks: _____

Committee Member 2: Name: _____ Designation: _____

[ ] Accepted [ ] * Deferred [ ] *Rejected Signature: _____

*Remarks: _____

FYP Manager: Fahad Sabah

[ ] Accepted [ ] *Deferred [ ] *Rejected Signature:_____

*Remarks: _____

Abstract

World is getting advance day by day, the manned air crafts are now moving towards UAV. Since UAVS are ground operated there exists chances of mishandling and crashing the air vehicles. To overcome this scenario we came to a solution of obstacle avoiding and path planning of UAVs. At first Environment map is processed and obstacles are detected then path is planned using A* Algorithm. Then it is capable of deploying this planned path in UAV. We have used Quadcopter which is the most stable among UAVS.

Acknowledge

First of all thanks to ALLAH ALMIGHTY who has guided us through every step of the way and it is only because of His blessings that we gathered enough audacity and Will to accomplish this project. There are many people to thank in a project of this large.
But first and foremost we would like to express our sincere gratitude to Miss Amna Khan (Assistant Professor, Department of Computer Science & Information Technology, Superior University), our project supervisor, for showing us the path of learning. Naturally, we would also like to thank our families for giving us the gift of an Education. Each one of you kept faith in us throughout our college and university career and were there for us when we needed you the most. Your undying love and support is immensely appreciated and for that we are grateful to have such wonderful parents.
Finally, we would like to take this opportunity to express our thanks and gratitude to all the persons who have directly or indirectly availed us in guiding our project.

Table of Contents

List of Figure

CHAPTER 1
Introduction

1 Introduction

The term UAV is an abbreviation of Unmanned Aerial vehicle, meaning aerial vehicles which operate without a human pilot. UAVs are commonly used in both the military and police forces in situations where the risk of sending a human piloted aircraft is unacceptable, or the situation makes using a manned aircraft impractical. One of the predecessors of today's fully autonomous UAVs were the "aerial torpedoes", designed and built during World War One. These were primitive UAVs, relying on mechanical gyroscopes to maintain straight and level flight, and flying until they ran out of fuel. They would then fall from the sky and deliver and explosive payload. More advanced UAVs used radio technology for guidance, allowing them to fly missions and return. They were constantly controlled by a human pilot, and were not capable of flying themselves. This made them much like todays RC model airplanes which many people fly as a hobby. It is interesting to note that the government considers all aircraft UAVs, if they are unmanned and used by a government or business. After the invention of the integrated circuit, engineers were able to build sophisticated UAVs, using electronic autopilots. It was at this stage of development that UAVs became widely used in military applications. UAVs could be deployed, fly themselves to a target location, and either attack the location with weapons, or survey it with cameras and other sensor equipment. Modern UAVs are controlled with both autopilots, and human controllers in ground stations. This allows them to fly long, uneventfully flights under their own control, and fly under the command of a human pilot during complicated phases of the mission.[1]

1.1 Outcomes

This project is very versatile that it approximately covers the entire field related subjects studied so far. Following outcomes are achieved:

- Obstacle detection and avoidance using image segmentation.
- Implementation of A* algorithm for shortest path finding.
- Developed a simulation model in MATLAB.
- Implementation in QUADCOPTER (In future).

1.2 Tools

- MATLAB
- Visual studio
- Photoshop
- 3D-MAX

1.3 Methodology Used

Project contains a hybrid methodology of development which is a mixture of three basic types of methodologies namely iterative, incremental and evolutionary. Our methodology is discussed below in this chapter.

1.4 Highlights of Discussion of Carious Chapters of Reports

1.4.1 Chapter 1 (Introduction)

This chapter contains information about basic information of two module i.e. object detection and path planning in an unknown environment and a hybrid communication methodology that we have used. Moreover, the achievements and contributions we had throughout the year.

1.4.2 Chapter 2 (Problem Statement)

This chapter contains the information about the problems to be addressed as a solution of our project.

1.4.3 Chapter 3 (Functional requirements)

This chapter contains information about the use cases according to the real time and according to the simulator. Moreover, the section contains non-functional requirements information such as efficiency requirement and reliability requirement.

1.4.4 Chapter 4 (Design and Architecture)

This chapter contains information that is very useful in order to understand the system. E.g. Activity diagram and Flow Chart. This chapter elaborates the obstacles we faced in making the design and architecture of the project.

1.4.5 Chapter 5 (Testing and Validation)

This chapter contains information about the test we have done in order to achieve the maximum output. The basic theme behind this chapter is that every project can have many bugs. To remove those bugs we have tested each.

1.4.6 Chapter 6 (Tool Use)

This chapter contains information about tool, languages, Applications and libraries used in this projects.

1.4.7 Chapter 7 (Conclusion)

This chapter includes brief summary and conclusion drawn from this projects.

1.4.8 Chapter 8 (Lesson learnt and future work / Enhancements)

This chapter contains information about the current development, future scope and future enhancement of this project.

1.4.9 Chapter 9 (Bibliography)

This chapter contains all the reference materials used for this project including plagiarism report and appendices.

1.5 Relevance to courses

1.5.1 Introduction to programming (ICP)

We have studied the basics of programming in this course. Using the concepts of C language we implement A* algorithms.

1.5.2 Design and Analysis of Algorithms (DAA)

We studied how different algorithms and how to calculate their running time.

1.5.3 Data structure (DS)

Data structure helped us to develop a simulator based on stack and priority queue technique.

1.5.4 Object oriented programming (OOP)

A* algorithm was implemented using OOP concepts.

1.5.5 Software engineering (SE)

This course helped us to evaluate our development methodologies.

1.5.6 Artificial Intelligence (AI)

Path planning is always an issue in the field of A.I. We used A* algorithm for path planning.

1.6 Project Background

An extensive amount of research has been carried out in obstacle detection and path planning as they are critical components of many robotic applications. This project aims to develop a system for Vision-based Obstacle Detection and Fair Path Planning for the UAV drone.

We have to find the optimal path from the initial to the final desired point with in an environment (represented by an image of the room) avoiding collision. In this scope, optimal is defined as the shortest collision-free path between two points. Our image represents world coordinates and is divided into two parts free space and obstacle. We applied image processing techniques on input images to recognize obstacles in first phase of the project. In the second phase A* algorithm was used to plan a fair path for the UAV drone. The planned shortest path will then be used to define a trajectory path using 8-connected neigh boring in order to provide a smooth path for the motion of the drone.

1.7 Software life cycle for this projects

It basically consists on iterative development, incremental model and evolutionary model as

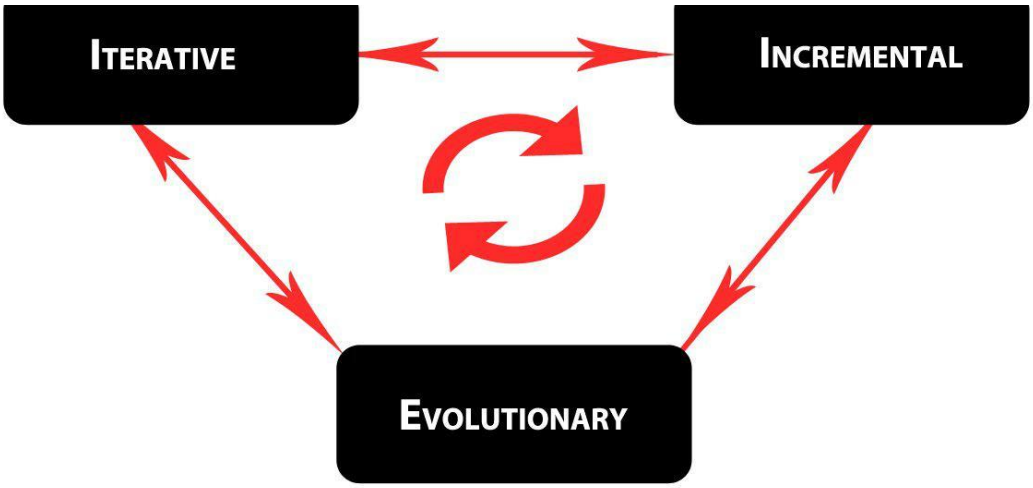


Figure 1: Software life cycle.

shown in Figure 1. The basic idea behind using all of these models is we develop one thing and then move to another but sometimes we feel that the previous unit should be changed in order to achieve our goal.

1.7.1 Iterative Model

An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model.

This project is basically research based. We were not sure in the beginning that through what ways we are going to achieve the goals. We started with some very basic techniques and design after completing/achieving those goals we review those and improvise them in next it iteration

1.7.2 Incremental

The incremental model is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements.

1.7.3 Evolutionary

The evolutionary model is a method in which software requirements are continuously evolving. The basic idea behind this is to meet the software requirements when the project completes. Usually the research based projects use this technique because new constraints and problems are popping up after every analysis of the requirement

CHAPTER 2
Problem Statement

2 Problem Definition

Drone can be successfully used in areas which are dangerous for human, such as buildings under fire, gas leakage in chemical plant or areas having explosive devices.

Putting human life in danger in an unknown environment is not a good idea. Human should at least have enough information about the environment so that he can take the security precautions.

2.1 Problem Statement

The work already done in this field is limited by constraints. The scope of those projects or researches is often limited by constraints like environments types, automation, or communication did work in this field like example [2]. Although, the systems was automated but it lose control when the drone went near heavy object like helicopter.

STARMAC Project used a quad-rotor for navigating the area, but they were also limited by environments and automation, as the quad-rotor could navigate only in on a pre-defined path, and also had problems in case of communication loss as it was controlled by a central server. Also, like many others, it created only a 2D map of the environment which is unable to provide full detail of a specific area or environment [3] .

A virtual view of any facility can give detail information about that area. In critical situations where human life risk is very high our system can help. An examples of such situations is:

2.1.1 Surveillance

Importance of surveillance drone is increasing as security is top priority in today's unpredictable society. With the advance in technology, researchers are able to build more drone to handle hazardous situations and save several lives. Research on building better surveillance systems is one of the highly funded projects these days. It can also be deployed in areas where hazardous materials are present due to a leakage [4].

UAV drone can play an important role in military battle, from patrol to dealing with potential explosives.

It can also help meeting challenges posed by the specter of urban terrorism. "Instead of having people get close to hazards such as unattended objects or car bombs, robots are used. If an operator concludes a dangerous object might explode, the drone could neutralize that object by shooting to detonate it," Goldenberg says. "Drone detect and explode in-ground mines or improvised explosive devices." These same drone systems are used for neutralizing or exploding forgotten ordnance and mines after conflicts cease [5].

2.2 Deliverable and Development Requirement

2.2.1 Deliverable requirements

- Creating 2D and 3D map.
- Detection of different objects.
- Follow a pre-defined path.
- Detect hurdle and avoid it.

2.2.2 Development Requirements

- Power supply. Laptop.
- Operating system Windows. C#.
- MATLAB.
- Unity 3D
- 3D Max

2.3 Feasibility of the system and management issues

There are few things we assume environment should have.

- Open/ Close 3D Environment
- System localizes itself upon a gird based ground system so there should be an initial & final point to drone
- Surface must be divided into gird.

CHAPTER 3

Requirement Analysis

3 REQUIREMENT ANALYSIS

3.1 Functional Requirements

3.1.1 Use Cases

Use Case 1:

Name	**UC1**
Summary	User will start the software
Pre-condition	Upload environment image
Basic Course of Event	Drone will initialize its components
Post condition	All components will start responding

Figure 2: Use Case 1. Start Software.

Use Case 2:

Name	**UC2**
Summary	User will give a picture
Pre-condition	drone system should be awaken
Basic Course of Event	Set task
Post condition	All components are working

Figure 3: Use Case 2. Give area to scan.

Use Case 3:

Name	**UC3**
Summary	Set the position on map
Pre-condition	drone system should be Awaken
Basic Course of Event	To have a starting position
Post condition	Localized form of drone in map

Figure 4: Use Case 3. Place drone and mark start point.

Use Case 4:

Name	**UC4**
Summary	Set the final point to see the behavior
Pre-condition	Starting point must be defined
Basic Course of Event	Have a final point
Post condition	Final point set

Figure 5: Use Case 4. Mark final point.

Use Case 5:

Name	**UC5**
Summary	This will show us how drone will move. Movement is same as in real world.
Pre-condition	Drone should have the starting point and destination.
Basic Course of Event	To analyze the movement
Post condition	Drone will explore its path and reach final point avoiding obstacles

Figure 6: Use Case 5. Start drone movement.

3.1.2 Real time scenario with full functionality

Figure 8 shows the scenario when some user interacts with drone system.

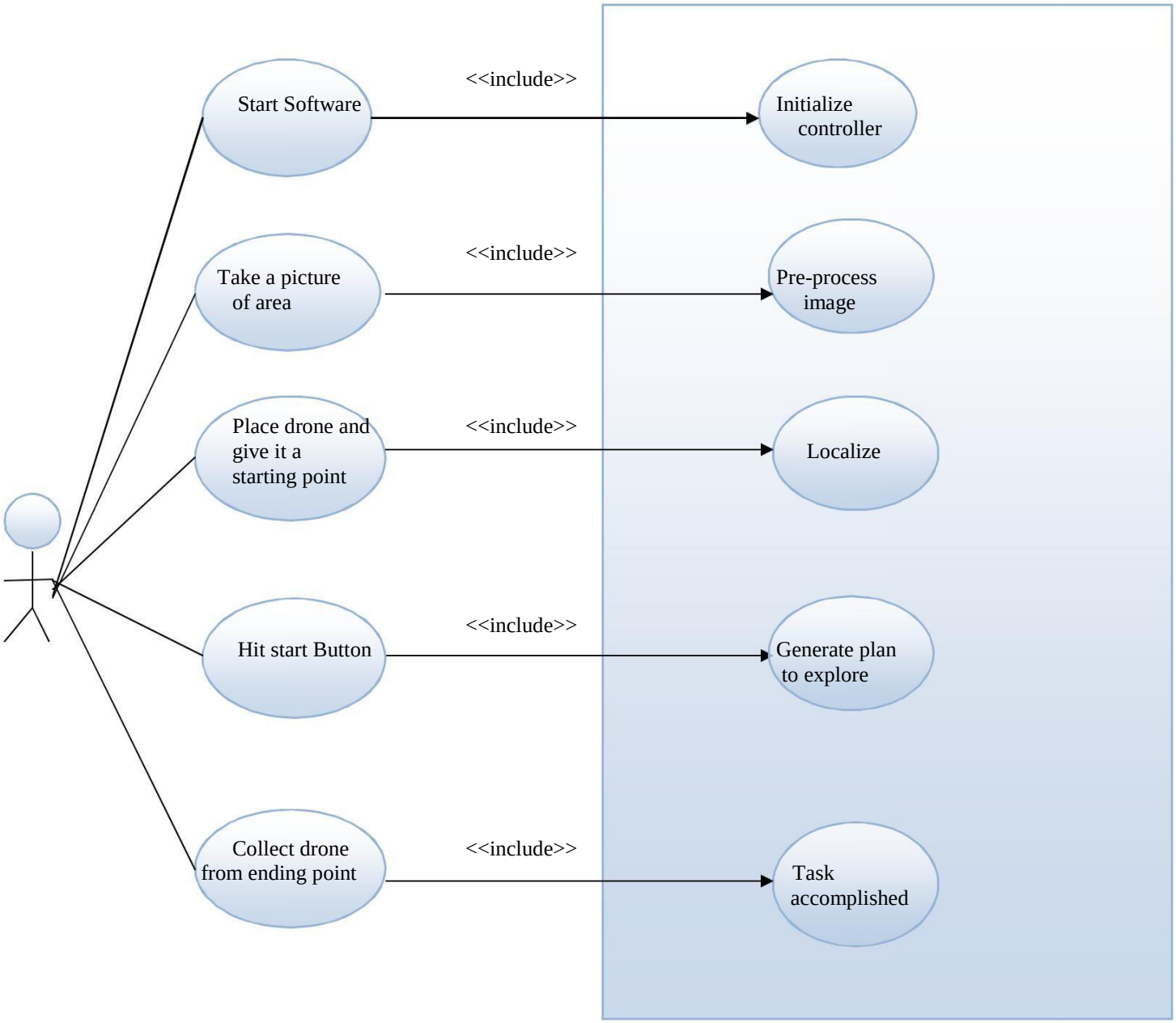


Figure 8: Real time scenario.

3.2 Non-functional requirement

3.2.1 Efficiency

Designing fair route for drone is a very important field now a day's as the technology is evolving and drone are performing the critical tasks of humans. Drone work in the environment where many humans being are working, cooperating with drone such as, factories, offices etc. In these environments, the collision-free path planning is one of the major problems to realize. As there are many obstacles in the environment, so drone should plan their own path that can avoid obstacles. This project contains object detection module which detect obstacles in an image and path planning module which is flexible enough to reach its final point avoiding obstacles and plan a shortest optimal path for drone.

3.2.2 Reliability

Navigating through an unknown environment is difficult for humans as well even we sometimes forget, from where we came and sometimes we could not see some objects in our path. Our navigation system is smart enough that drone can traverse any flat surface environment efficiently.

CHAPTER 4

Design & Architecture

4 System Architecture

Deployment diagram

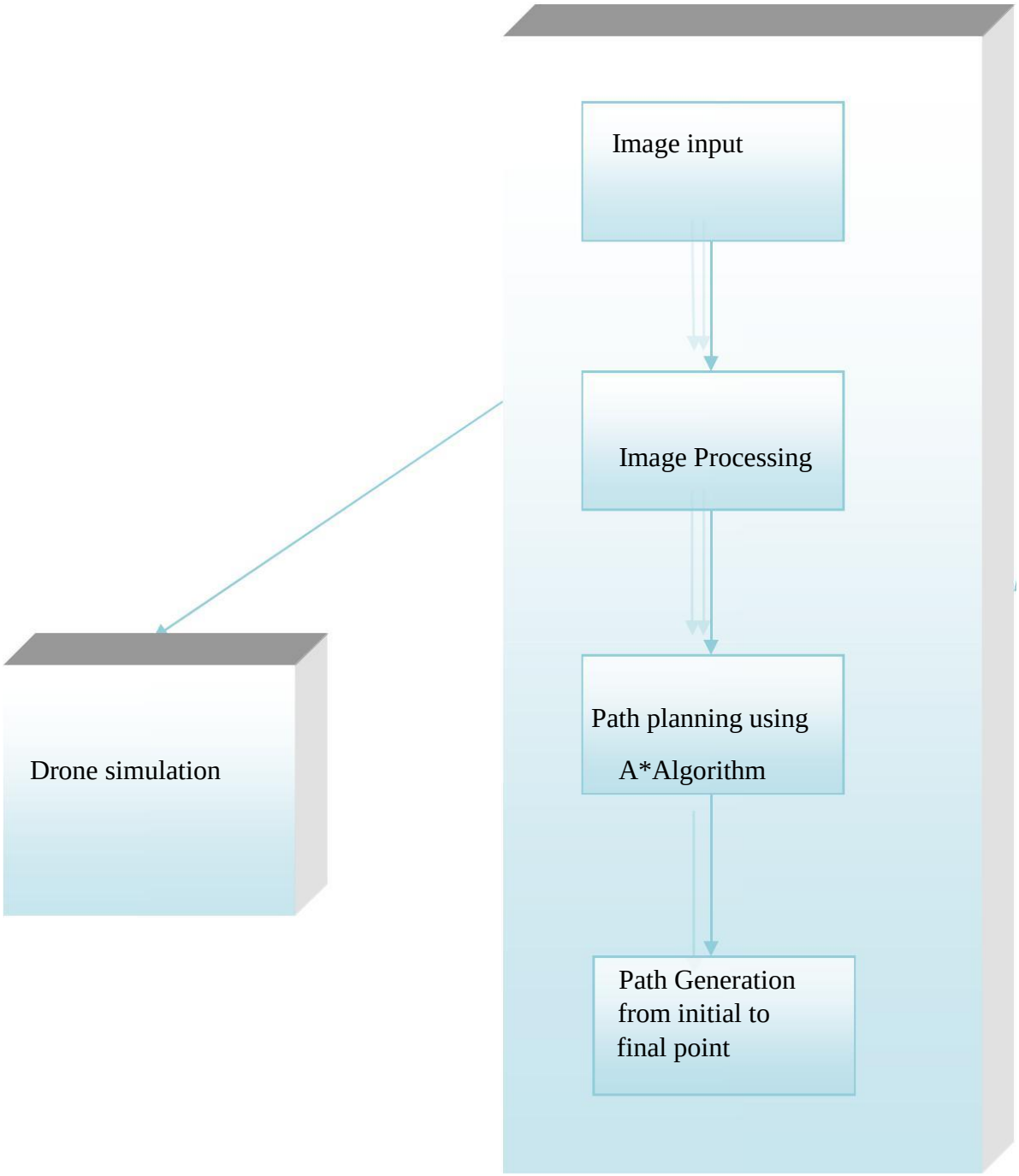


Figure 9: Deployment Diagram.

4.1.1 Computer

Computer is the main processing unit of the system. It contains several modules of the project in it. All the decision making processes are done here. Computer will process image and generate optimal path. The description of the modules it contains is given below.

a. Image input

In this module, image of a room has been captured to be used for obstacles detection.

b. Image processing

Image preprocessing techniques are being applied on the input image.

I. Resizing the image to maintain the standard size.
II. Conversion in to grayscale.
III. Complement of image.
IV. Thresholding.

The purpose of preprocessing is to improve the quality of the image being processed. It makes the subsequent phases of image processing like detection of obstacles easier.

- **Path planning using A***

  A* algorithm was used to plan a fair path for the Drone. The planned shortest path will then be used to define a trajectory path using 8-connected neighboring in order to provide a smooth path for the motion of the Drone.

- **Path Generation**

  Path has been generated from initial to final point using A* algorithm.

4.1.2 Drone

The basic task we are performing in order to move the system is just by giving some commands to the drone base to move from initial point to final point avoiding obstacles. Drone follows our command and performs the task accordingly.

4.2 Activity Diagram

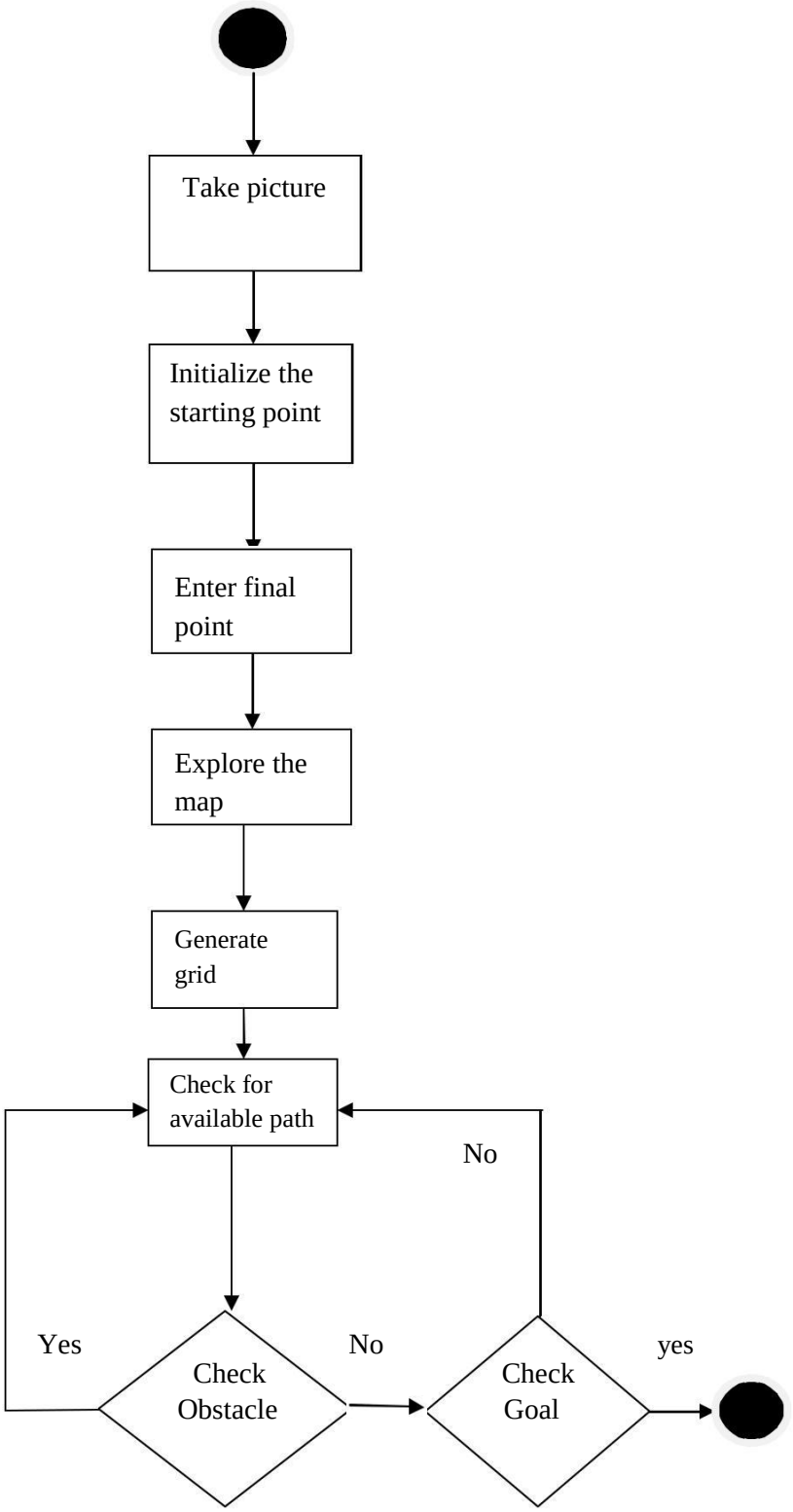


Figure 10: Activity diagram.

Figure 10 shows that how system moves from various stages to achieve a task. Drone initially is at idle state and requires some input to move. After giving input generated optimal path planned through A* to drone, it will move from initial point to final avoiding obstacles.

4.3 Project diagram

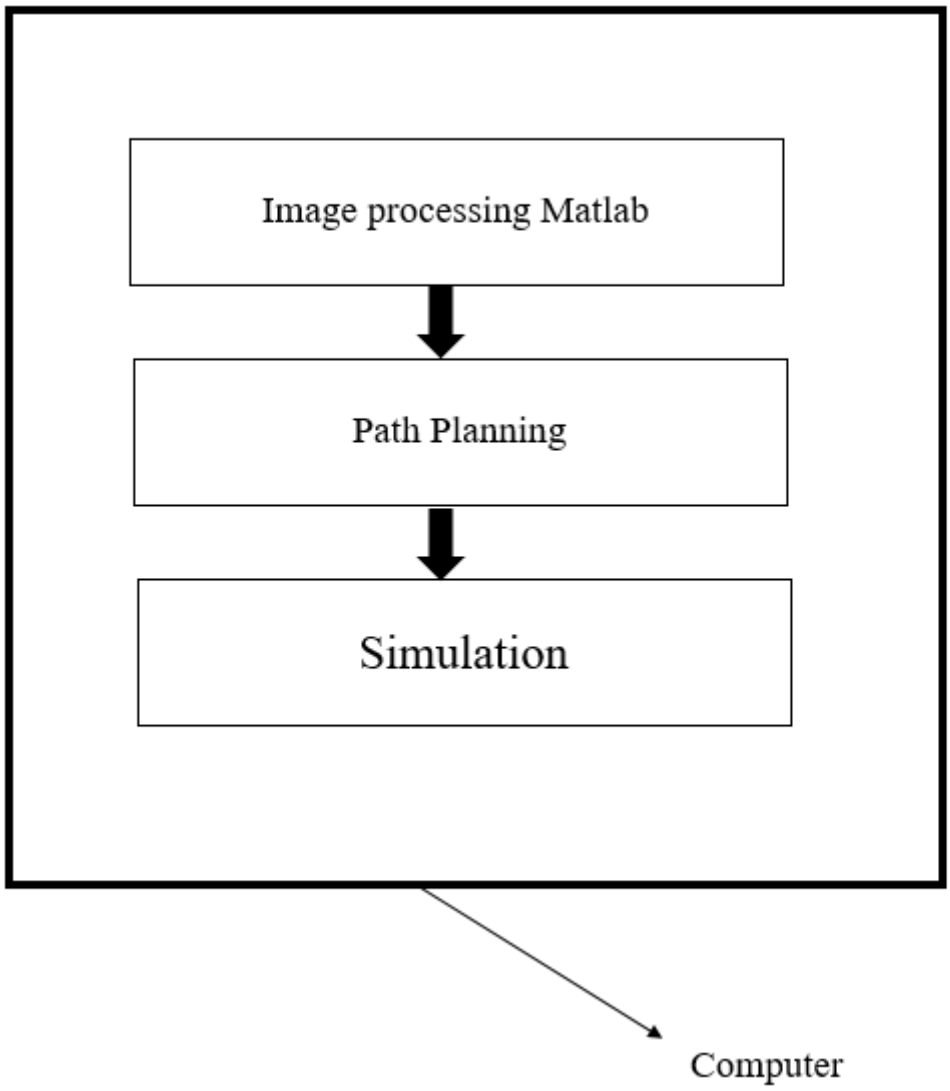


Figure 11: Project diagram.

A* Flow Chart

The main idea of A* algorithm is that, it avoids expanding paths that are already expensive. A* Evaluation function f(n) = g(n) + h(n).

g(n) = exact cost so far to reach n.
h(n) = estimated cost to goal from n.
f(n) = estimated total cost of cheapest path through n to goal.
n = node number

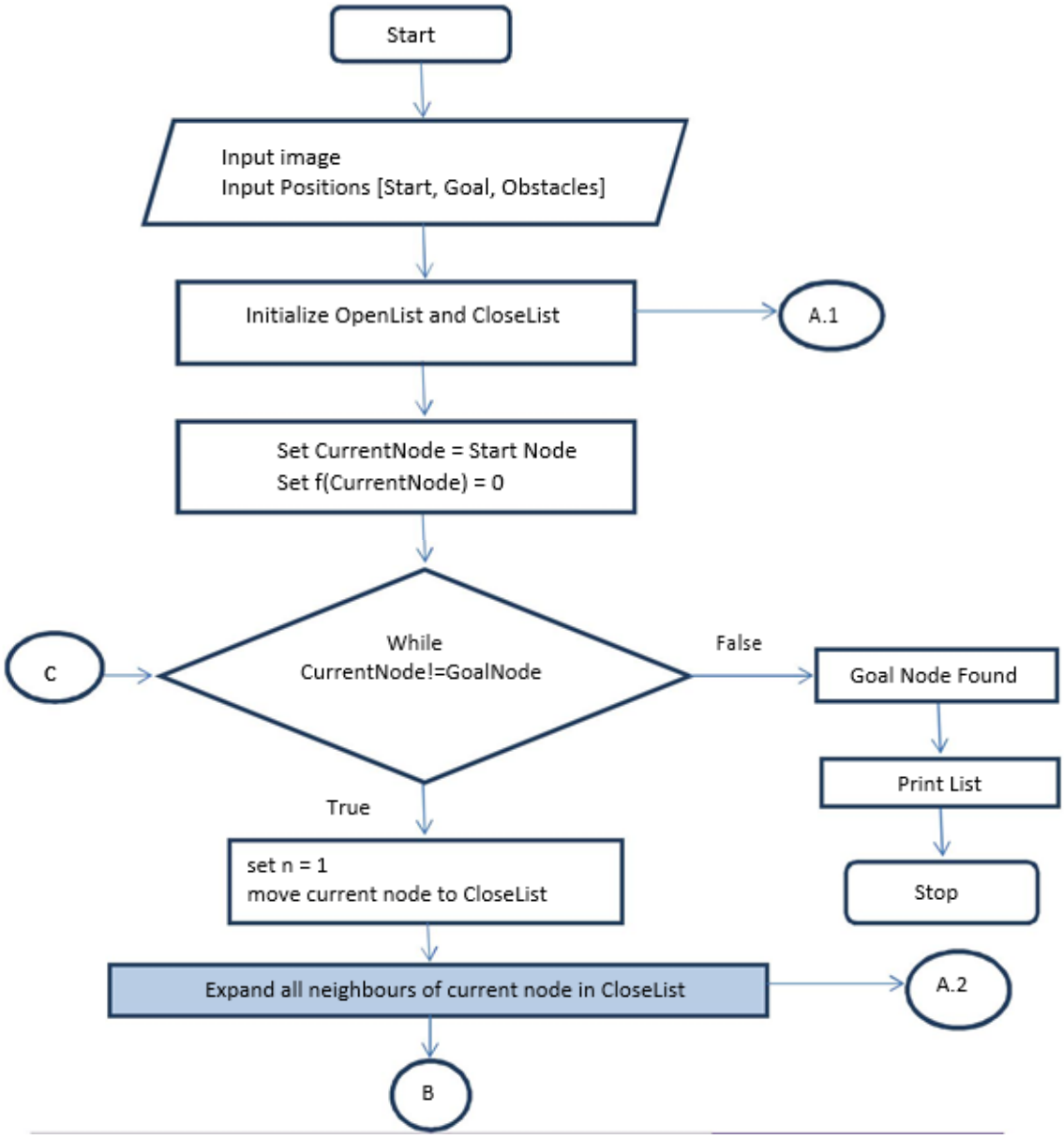
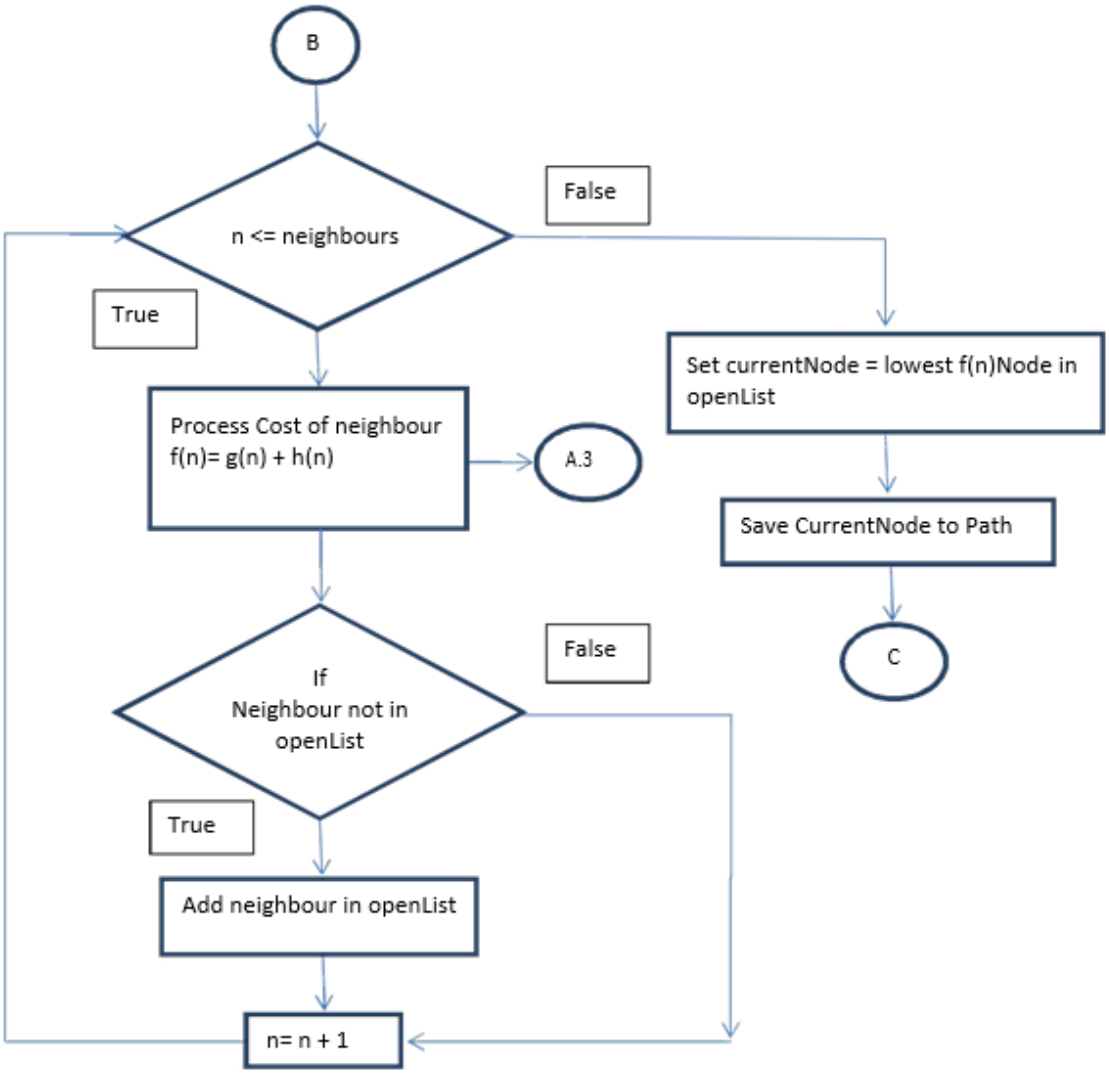
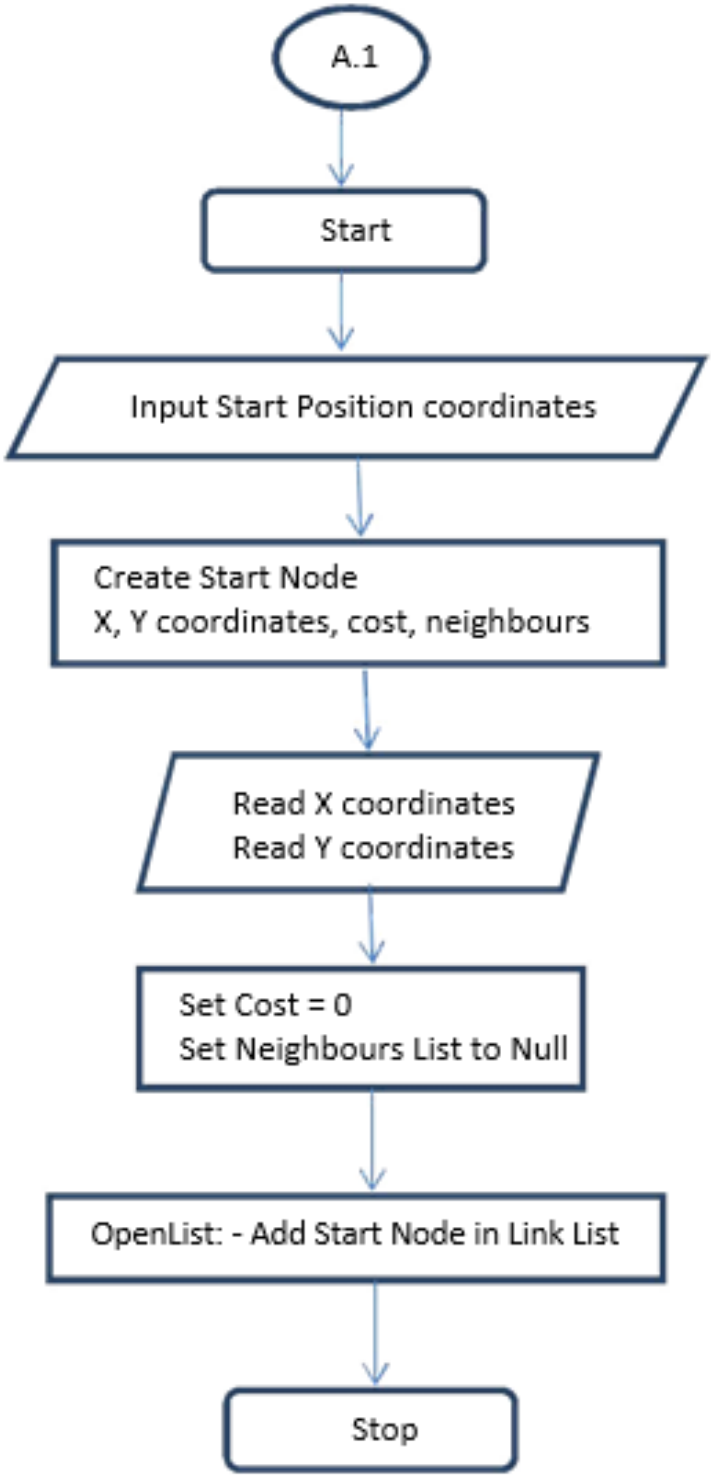
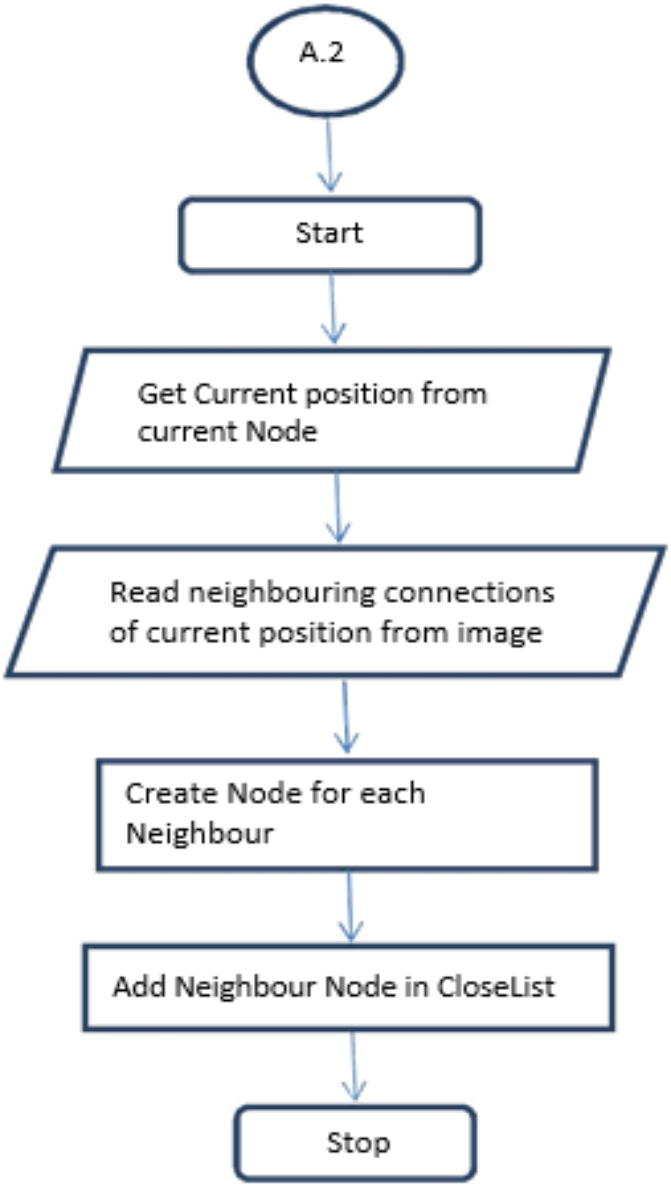
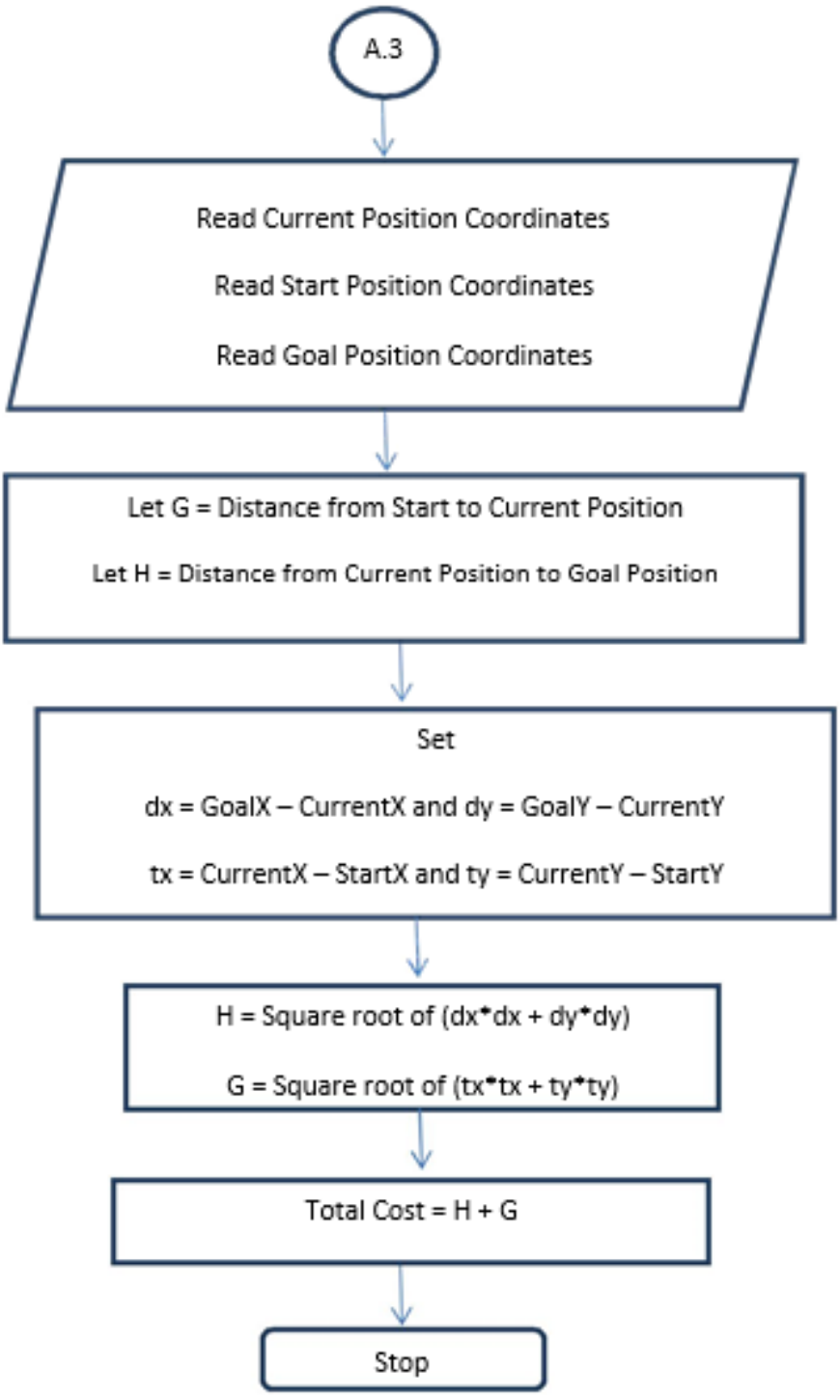


Figure 12: A Flow Chart.*


B
n <= neighbours
False
True
Set currentNode = lowest f(n)Node in openList
Process Cost of neighbour f(n)= g(n) + h(n)
A.3
Save CurrentNode to Path
C
If Neighbour not in openList
False
True
Add neighbour in openList
n= n + 1


A.1
Start
Input Start Position coordinates
Create Start Node
X, Y coordinates, cost, neighbours
Read X coordinates
Read Y coordinates
Set Cost = 0
Set Neighbours List to Null
OpenList: - Add Start Node in Link List
Stop


A.2
Start
Get Current position from current Node
Read neighbouring connections of current position from image
Create Node for each Neighbour
Add Neighbour Node in CloseList
Stop


A.3
Read Current Position Coordinates
Read Start Position Coordinates
Read Goal Position Coordinates
Let G = Distance from Start to Current Position
Let H = Distance from Current Position to Goal Position
Set
dx = GoalX – CurrentX and dy = GoalY – CurrentY
tx = CurrentX – StartX and ty = CurrentY – StartY
H = Square root of (dx*dx + dy*dy)
G = Square root of (tx*tx + ty*ty)
Total Cost = H + G
Stop

CHAPTER 5

Testing and Validation

5 TESTING AND VALIDATION

5.1 Verification

The system was tested using different parameters.

- When the right parameters are given the systems performs the desired tasks.

  The system was given complete area parameters X and Y coordinate points. Start and goal, and the map creation process started.

- In case of wrong values the system notified the error.

  The system was provided the wrong information like instead of giving integer point the system was provided character, string or special character, in that case the system notify an error.

  The same happened when incomplete area information was provided.

5.2 Validation

The system was tested with different parameters to check the validity of its input.

- Different coordinate points were given with different grid positions of the world and they were placed accordingly in the global map.
- When provided the coordinate point where there is obstacle, the system notifies that change location, goal cannot be reached.
- The system was provided with a grid with unknown paths and obstacles, and a path was calculated during exploration.

5.3 Usability Testing

The system was tested by different users for Path Planning algorithms, and with the suggested changes the algorithms were optimized.

5.4 Integration Testing

The system was tested in the following groups.

- A* Simulation
- using Matlab Path Planning.
- Obstacle Detection.
- Localization
- Autonomous movement Drone
- Wireless communication between Drone and XBEE Manual control of drone using arrow keys.

5.5 System Testing

The system was completely tested in a control environment in which grid was properly adjusted and the environment was static. The area information was provided accordingly.
After starting the system, the drone moved to the desired recording position while being assisted with the A* Path Planning algorithm, base on the start and destination point the drone find his path while avoiding obstacles.
After the world scan was completed the system stopped, and the complete map of the specified world was created.

5.6 Acceptance Testing

The System testing phase verified the completion and working of all the desired features in the system.

The testing in the control environment gave maximum percentage of the expected result and passed the test.

5.6.1 Output of original image:

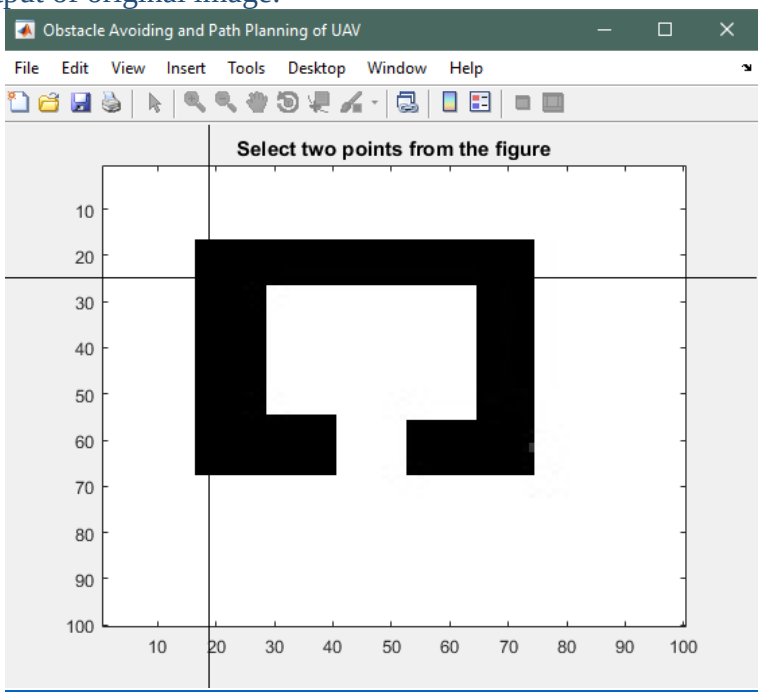


Figure 13: Obstacle detection.

5.6.2 Output with obstacles:

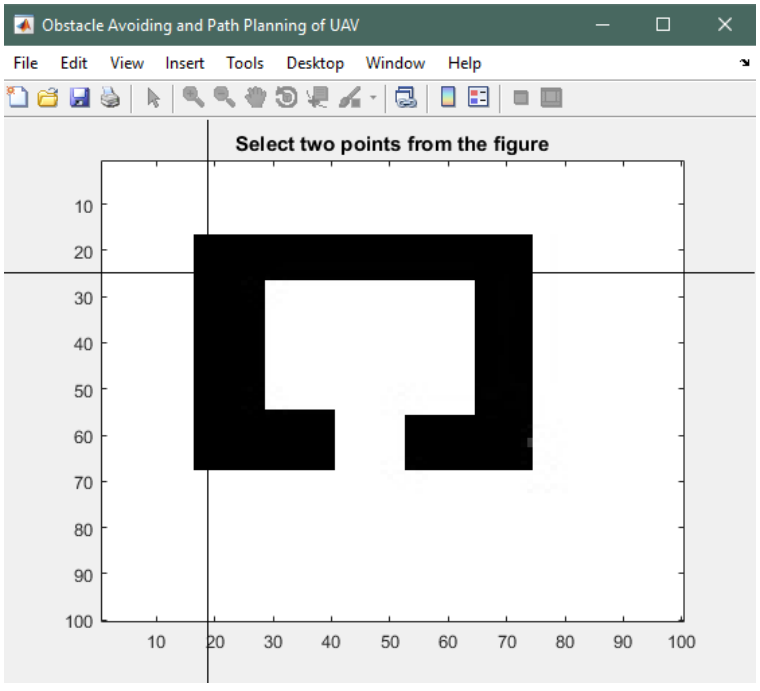


Figure 14: Output.

5.6.3 start and goal points:

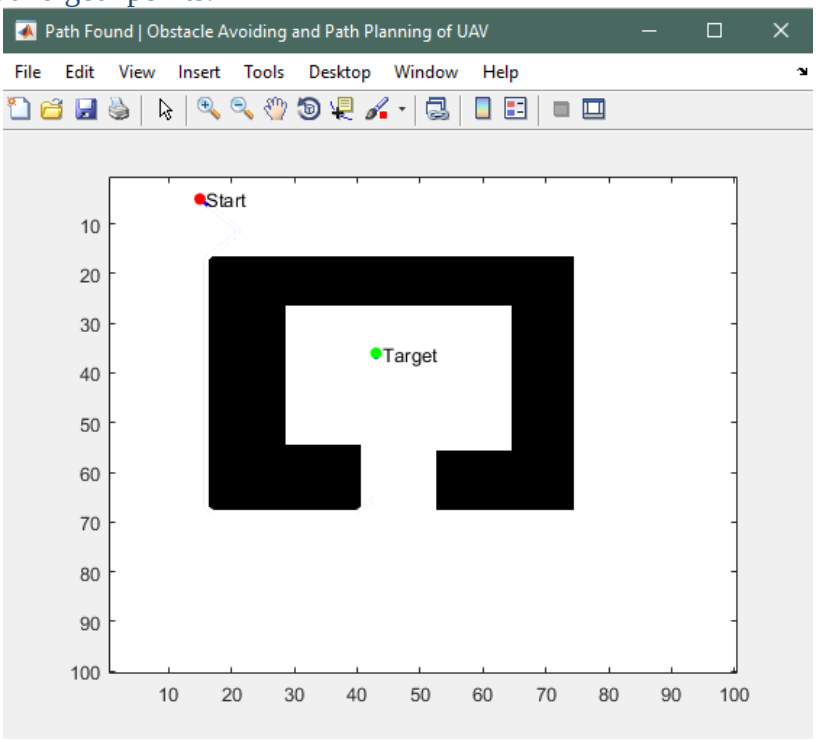


Figure 15: start location.

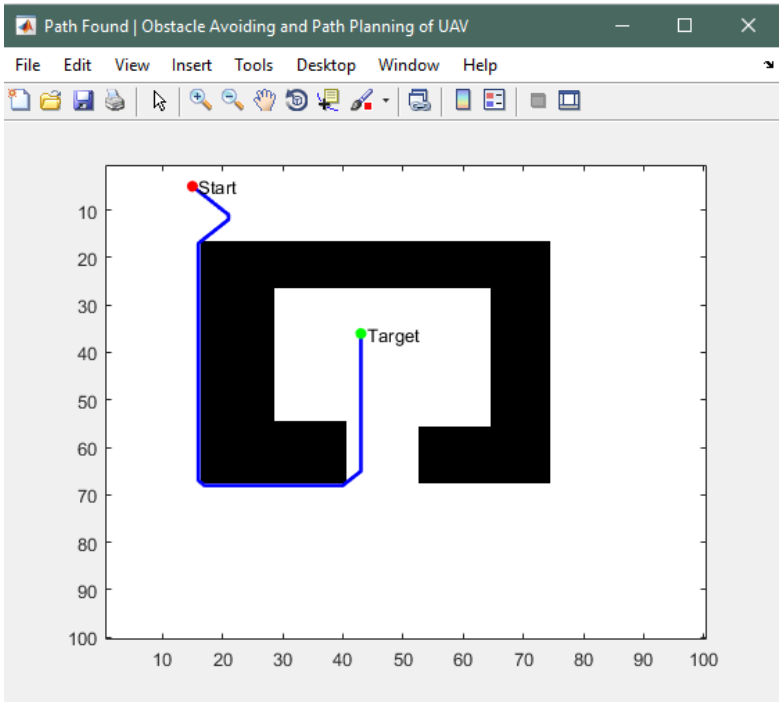


Figure 16: goal location.

Figure 15 and 16 shows the worst case scenario of our A* algorithm. In figure 15 the start points are given to the drone but due to the obstacle in that location the path is found between the starting point and ending point. The same message is also displayed in figure 16 when a start points are given to drone but there is obstacle in that location.

CHAPTER 6

Tools Used

6 Tool Use

6.1 Languages

Following are the languages used in this project

1. Matlab
2. C#

6.2 Tools

Following are the tools used in this project

1. MATLAB
2. Visual studio
3. Photoshop
4. 3D-MAX
5. Unity 3D

CHAPTER 7

Conclusion

7 Conclusion

UAV drone are pilot less flying vehicle. As drones are now automating these days it is necessary to completely automate in path planning and obstacle avoiding so it can reach its destination safely. We have finalized a program after research and development of path planning and obstacle avoiding of UAV drone. This program operates by fetching the environment map and planning path from starting to destination and simulate drone on its path to reflect the path planning in real scenarios. The aim of this project was to design and fabricate a drone to avoid obstacles in its path, and reach a fixed goal position. We accomplished the task of object detection using image processing and path planning using A* (A star) algorithm to design a fair path for drone, in known environment. This project successfully finds optimal path from the initial to the final desired point with in an environment. In this scope, optimal is defined as the shortest collision-free path between two points. The planned shortest path will be used to define a trajectory path using 8-connected neighboring in order to provide a smooth path for the motion of drone.

CHAPTER 8

Future Work

8 Future Work

- Designing and path planning of UAV by adding further features such as night vision etc.
- Moreover, we will take a step to get a full detailed view of environment by capturing 3D images.
- We can also add the control of UAV through satellite.
- Implementing simulation in real drone.
- Fully automated intelligent drone with path planning and obstacle avoiding advance algorithm.

CHAPTER 9

Bibliography

9 Bibliography

9.1 Roles & Responsibilities

Sr#	Name	Role & Responsibilities
1	Muhammad Faisal	Image Processing & Documentation
2	Hasan Irtaza Mirza	Path planning & Documentation
3	Hafiz Awais	Software Integration
4	Izrar Ahmad	Research & Data Collection

9.2 WBS (Work Breakdown Structure)

Sr	Tasks	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
1	**Research on Drones**										
2	**Collection of Drone Parts and assemble**										
3	**Documentation**										
4	**Research on path planning**										
5	**Research on Image Processing**										
6	**Image 2D to 3D Conversion**										
7	**General Research on Project**										
8	**Develop Path planning Module**										
9	**Develop Image processing module**										
10	**Project Documentation**										
11	**Final Modules integration and project compilation**										

9.3 Scope Statement

UAV drone are pilot less flying vehicle. As drones are now automating these days it is necessary to completely automate in path planning and obstacle avoiding so it can reach its destination safely. We have finalized a program after research and development of path planning and obstacle avoiding of UAV drone. This program operates by fetching the environment map and planning path from starting to destination and simulate drone on its path to reflect the path planning in real scenarios.

Annex IV: Meeting Performa

Department of CS & IT Superior University, Lahore

Final Year Project – Minutes of Meeting Progress Report

Project Title: _____

Supervisor: _____

Student name	Progress of tasks assigned in previous meeting	New tasks	Date	Signatures

I, _____, being Project Supervisor of this group, certify that above mentioned information is true as per my knowledge and concern. I allow the group to present its Final Year Project to the Panel.

_____

Signature of Supervisor

Code

MainProjectAstarDrone.m

```
function [] = MainProjectGridBasedAlgos()
currentfolder = pwd ;
mapName = 'Map (1)'
fileName = strcat(currentfolder ,'\InputImages\',mapName,'.jpg');
rgbI = imread(fileName);
I = rgb2gray (rgbI);
[w , h ] = size(I)
a=figure('Name', 'Obstacle Avoiding and Path Planning of UAV',
'NumberTitle','off'), imagesc(1:h, 1:w, rgbI),title ('Select two points from
the figure');
hold on
[y,x] = ginput (2);
xs = x(1);
ys = y(1);
xs = ceil ( xs );
ys = ceil ( ys);
xd = ceil ( x(2));
yd = ceil ( y(2));
str = [' start position ', num2str(xs), ',' ,  num2str(ys)];
disp(str)
str = [' end position ', num2str(xd), ',' , num2str(yd)];
disp(str)
P =[ xs ys ; xd yd];
    text(yd+1 ,xd,'Target');
    text(ys+1 ,xs,'Start');
scatter(P(1,2), P(1,1),'Filled','ro')
scatter(P(2,2), P(2,1),'Filled','go');
hold on
if ( xs == xd && ys == yd )
    disp('invalid data : same start and end positions, program terminated');
    close;
else
    CostMatrixofImageModified = ones( w,h);
    CostMatrixofImage = ones( w,h);
    level = graythresh( I );
    binaryImage = im2bw( I, level);
    finalImage = binaryImage;
    twoWayImage = imcomplement( finalImage);
    twoWayImage = im2uint16( twoWayImage);
    finalImage = im2uint16(finalImage);
 if ( (finalImage (xs, ys ) == 65535 ) && ( finalImage (xd, yd ) == 65535))
        doubleImage = ones(w,h);
        doubleImage = im2double ( finalImage);
        sedilate = strel('square',5);
        dilateImage = imerode(doubleImage, sedilate);
```

```
        tempImage = dilateImage ;
        dilateImage = im2uint16(dilateImage);
        tic
        [PathAstar , Found , CostMatrixofImageModified] =
AstarSearch(imcomplement(binaryImage),[xs,ys],[xd,yd]);
      toc
  close(a);
    figure('Name', 'Path Found | Obstacle Avoiding and Path Planning of UAV
', 'NumberTitle','off'), imagesc(finalImage);
    hold on
    plot( PathAstar(:,2), PathAstar(:,1), 'LineWidth',2, 'Color','b');
    P =[ xs ys ; xd yd];
    text(yd+1 ,xd,'Target');
    text(ys+1 ,xs,'Start');
scatter(P(1,2), P(1,1),'Filled','ro');
scatter(P(2,2), P(2,1),'Filled','go');
    colormap gray
      hold off
     end
    end
```

AstarSearch.m

```
function [PathTake, Found, ExpansionGrid]=AstarSearch(grid,init,goal)

cost=1;
Found=false;
Resign=false;
mcells = 0;

init;
goal ;


Heuristic=CalculateHeuristic(grid,goal);

ExpansionGrid(1:size(grid,1),1:size(grid,2)) = -1;

ActionTaken=zeros(size(grid));

OptimalPath(1:size(grid,1),1:size(grid,2))={' '};


delta = [-1,  0;
          0, -1;
          1,  0;
          0,  1;
```

```
          -1,1;
          1,-1;
           1,  1;
          -1, -1];
for i=1:size(grid,1)
     for j=1:size(grid,2)
         gridCell=search();
         if(grid(i,j)>0)
            gridCell=gridCell.Set(i,j,1,Heuristic(i,j));
         else
             gridCell=gridCell.Set(i,j,0,Heuristic(i,j));
         end
         GRID(i,j)=gridCell;
         clear gridCell;
     end
 end
Start=search();
Start=Start.Set(init(1),init(2),grid(init(1),init(2)),Heuristic(init(1),init(
2)));
Start.isChecked=1;
GRID(Start.currX,Start.currY).isChecked=1;
Goal=search();
Goal=Goal.Set(goal(1),goal(2),grid(goal(1),goal(2)),0);
 OpenList=[Start];
ExpansionGrid(Start.currX,Start.currY)=0;
small=Start.gValue+Start.hValue;
count=0;
 while(Found==false || Resign==false)
     small=OpenList(1).gValue+OpenList(1).hValue+cost;
for i=1:size(OpenList,2)
        fValue=OpenList(i).gValue+OpenList(i).hValue;
        if(fValue<=small)
            small=fValue;
             mcells = mcells+1;
            ExpandNode=OpenList(i);
            OpenListIndex=i;
        end
  end
 OpenList(OpenListIndex)=[];
  ExpansionGrid(ExpandNode.currX,ExpandNode.currY)=count;
    count=count+1;
  for i=1:size(delta,1)
        direction=delta(i,:);
        if(ExpandNode.currX+ direction(1)< 1 ||
ExpandNode.currX+direction(1)>size(grid,1)||ExpandNode.currY+ direction(2)<1
|| ExpandNode.currY+direction(2)>size(grid,2))
            continue;
        else
```

```
NewCell=GRID(ExpandNode.currX+direction(1),ExpandNode.currY+direction(2));
           if(NewCell.isChecked~=1 && NewCell.isEmpty~=1)

GRID(NewCell.currX,NewCell.currY).gValue=GRID(ExpandNode.currX,ExpandNode.cur
rY).gValue+cost;
                GRID(NewCell.currX,NewCell.currY).isChecked=1;
                OpenList=[OpenList,GRID(NewCell.currX,NewCell.currY)];
                ActionTaken(NewCell.currX,NewCell.currY)=i;
             end
      if(NewCell.currX==Goal.currX && NewCell.currY==Goal.currY &&
NewCell.isEmpty~=1)
                Found=true;
                Resign=true;
                disp('Search Successful');
                GRID(NewCell.currX,NewCell.currY).isChecked=1;
                ExpansionGrid(NewCell.currX,NewCell.currY)=count;
                GRID(NewCell.currX,NewCell.currY);
                break;
            end
        end
    end
  if(isempty(OpenList) && Found==false)
         Resign=true;
         disp('Search Failed');
         break;
     end
 end
 PathTake=[];

 if(Found==true)
     Policy={'Up','Left','Down','Right','Diag Down','Diag Up','Diag
Down','Diag Up'};
     X=goal(1);Y=goal(2);
     OptimalPath(X,Y)={'GOAL'};
     while(X~=init(1)|| Y~=init(2))
         x2=X-delta(ActionTaken(X,Y),1);
         y2=Y-delta(ActionTaken(X,Y),2);
         OptimalPath(x2,y2)=Policy(ActionTaken(X,Y));
         PathTake=[PathTake;[X,Y]];
         X=x2;
         Y=y2;
     end
     PathTake=[PathTake;[init(1),init(2)]];
 else

     disp('No Path to Display');
     h = msgbox('No path to display');
     Total_Elapsed_Time=toc
```

```
 end
end
```

CalculateHeuristic.m

```
function [Heuristic]=CalculateHeuristic(grid,goal)

Heuristic=zeros(size(grid));

for i=1:size(grid,1)
    for j=1:size(grid,2)
        Heuristic(i,j)=sqrt((i-goal(1))^2+(j-goal(2))^2);
    end
end


end
```

Search.m

```
classdef search

    properties
        gValue;
        currX;
        currY;
        isEmpty;
        isChecked;
        hValue;
    end

    methods
        function obj=search()
            obj.currX=0;
            obj.currY=0;
            obj.gValue=0;
            obj.isEmpty=0;
            obj.isChecked=0;
            obj.hValue=0;
```

```
        end

        function obj=Set(obj,X,Y,EmptyStatus,heuristic)
            obj.currX=X;
            obj.currY=Y;
            obj.isEmpty=EmptyStatus;
            obj.hValue=heuristic;
        end

    end
end
```

References

[1] Dragon Fly inc, An Introduction to Unmanned Aerial Vehicles (UAVs)

http://www.draganfly.com/blog/introduction-to-unmanned-aerial-vehicles-uavs/

[2] Civilian drone crashes into Army helicopter, Danielle Furfaro

http://nypost.com/2017/09/22/army-helicopter-hit-by-drone/

[3] Tomlin, "Starmac project.," 2009.

[4]K. Kaur, "Night Surveillance Robots,"
http://www.azorobotics.com/article.aspx?ArticleID=119#4
.

[5] Applications of robots, "Industrial-Robotics-Industry-Insights," *Robotics-in-Security-and-Military-Applications.*