USING A USV TO EFFICIENTLY CLEAR UNCERTAINTY FROM AERIAL IMAGES FOR RESCUE BOAT PATH PLANNING IN FLOODED URBAN ENVIRONMENTS

A Thesis

by

MEHMET FATIH OZKAN

BE, Mevlana University, 2016

Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

Texas A&M University-Corpus Christi
Corpus Christi, Texas

May 2019

May 2019

USING A USV TO EFFICIENTLY CLEAR UNCERTAINTY FROM AERIAL IMAGES FOR RESCUE BOAT PATH PLANNING IN FLOODED URBAN ENVIRONMENTS

A Thesis

by

MEHMET FATIH OZKAN

This thesis meets the standards for scope and quality of
Texas A&M University-Corpus Christi and is hereby approved.

Scott A. King, PhD
Chair

Luis Garcia Carrillo, PhD
Co-Chair

Junfei Xie, PhD
Committee Member

May 2019

ABSTRACT

Mapping and path planning in disaster scenarios is an area that has benefited from aerial imaging and unmanned aerial vehicles (UAVs). However, the integration of an unmanned surface vehicle (USV) in flood rescue operations has not received much attention. We propose a novel map generation and path planning algorithm, which makes use of aerial imaging provided by a UAV in combination with surface level information provided by a USV. Since the aerial image is a 2D projection of a 3D world, some areas of interest could be uncertain, such as under trees. Despite this issue, a Probabilistic Roadmap (PRM) path planning algorithm can be applied to the image in order to find near-optimal paths for a rescue boat between initial and target locations. With the method proposed here, the preliminary PRM solution is further improved by means of an online feedback structure, where local information provided by the USV is incorporated to the overall map as soon as it becomes available, eliminating uncertainties. Simulation results demonstrate the effectiveness of the proposed approach for improving the time response in search and rescue operations in flooded areas.

ACKNOWLEDGMENTS

I would like to thank my chair Dr. Scott King and my co-chair Dr. Luis Garcia for guiding me and for being generous throughout any process of the research which I greatly appreciate. I would like to thank my committee member, Dr. Junfei Xie for the efforts she expended. I am deeply appreciated to Texas A&M University-Corpus Christi for this great learning opportunity, for the people I met and for the fond memories.

My sisters Esra, Tuba, Sule, my brother Fuat, my parents Sabit and Bakdisan: thank you for the deep love and support, I owe you all more than ever.

TABLE OF CONTENTS

CONTENTS PAGE

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Natural disasters have serious impacts on human lives in terms of both losses of life and property. One of nature's most turbulent years was 1998. Floods wreaked havoc in Asia, Europe, North America, and South America. Many people suffered from these heavy floods which had cost great casualties and losses. USVs and UAVs have seen recent use in helping with various rescue operations. There are many application areas of USVs which consist of environmental monitoring [1], hazardous spill detection [2], victim recovery [3], river mapping [4], surveillance of shipwreck survivors at sea [5] and object recovery on the sea surface [6]. In Hurricane Wilma, 2005, UAVs and USVs were used for the first time as a cooperation system for the recovery phase of disaster management by detecting damage to seawalls and piers, locating submerged debris and determining safe lanes for sea navigation [2]. However, the existing UAV and USV cooperation systems have not had much attention for flood relief operations, especially for victim search and rescue. There are many challenges for flood rescue operations with existing systems where the time response is critical. The environment is completely changing under flood conditions, especially when some chemical substances spill into the flood areas and it causes a dangerous environment for victims and as well as rescue operators. Rescue boats are widely used for searching and rescuing the victims in the flooded areas. However, rescue boats have a limited view while searching and rescuing the victims. By using UAVs, an aerial image can be taken off the flooded environment and this aerial image can provide global information such as the location of victims and landmarks. The rescue operations can be organized based on a generated flood environment map by using this global information. Since global information includes the 2D projections of a

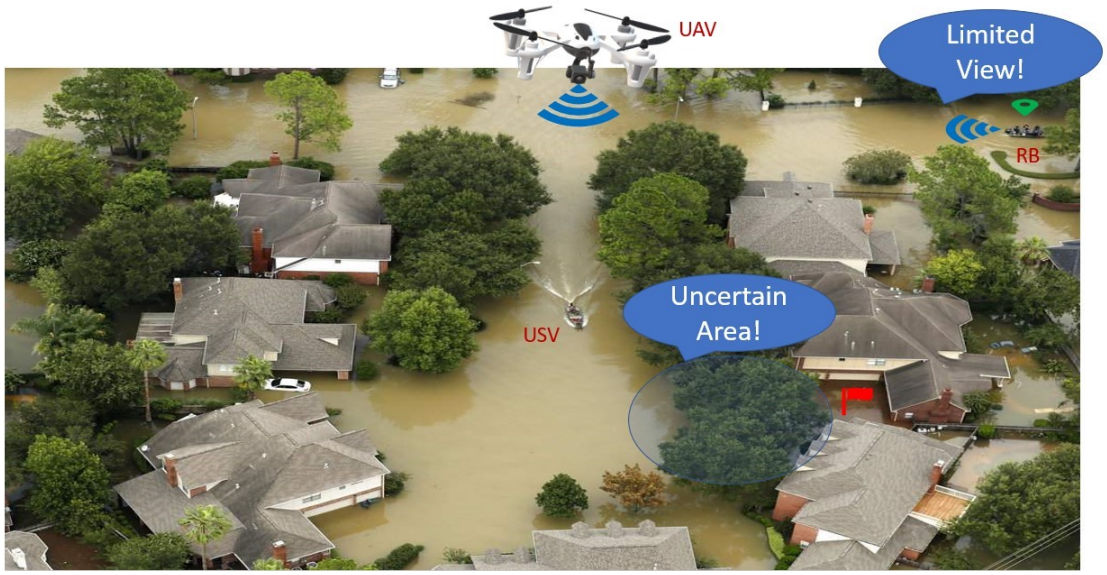


Figure 1.: Aerial view of a flooded environment

3D world, some areas of interest could be uncertain, such as under trees. By using the USV, the local information such as clearance of the trees can be inferred from the surface and then this local information can be helpful to reduce uncertainties to enhance the modeled map for rescue boat path planning. Figure 1 displays a typical urban flood scenario where a USV can work to help rescue operations. In this scenario, the trees occupy a large area, making it hard to detect the clearances. By using a USV, the clearance can be determined and then combined with global aerial information to cooperatively build a map model for path planning of a rescue boat. This research mainly aims at a cooperative map generation by using global information from UAV and local information from USV to help flood rescue operations by applying path planning and decision-feedback algorithms. To the authors' knowledge, this is the first known study of rescue boat path planning by using USV and aerial imaging in flooded urban environments.

1.1 Related Work

Using unmanned vehicles in rescue operations is not a brand new topic. Deng et al. [7] studied automatic ground map building and path planning in a UAV/UGV cooperative system for ground disaster rescue operations. More recently, Lakas et al. [8] introduced a framework for cooperative mission planning where a UAV and a UGV work cooperatively for a rescue task. Zhang et al. [9] introduced a new system which consists of a USV, a UAV, and a take-off and landing system. In 2017, Xiao et al. [10] present the first known implementation of a small UAV visually navigating a USV to solve flood rescue operations the rescuers victims in a more efficient manner.

In the field of map building, many computer vision algorithms have been developed in the literature. Costea et al. [11] proposed a system for geo-localization from aerial images in the absence of GPS information. This research includes the development of computer vision algorithms for the recognition of road, intersections, buildings, and landmarks. Zhou et al. [12] present an efficient road detection and tracking framework in UAV videos is proposed. Li et al. [7] proposed a ground map construction by the aerial image from UAV which was processed with image denoising, correction and obstacle detection techniques in UAV/UGV cooperation system. A survey on computer-vision algorithms for obstacle detection in aerial images which are produced by UAV is analyzed in [13]. More recently, Gunasekaran et al. proposed a map generation in a static unknown environment by using a mobile robot [14].

In the field of path planning, it has become one of the fundamental study areas in UGV and USV systems. Cheng et al. [15] proposed an improved hierarchical A* algorithm to solve parking path planning issues of a large park. In [16], a heuristic-based method is proposed to search the feasible initial path efficiently to solve the

problem of dynamic environments. Li et al. [7] proposed a hybrid path planning method which consisted of genetic algorithm and local rolling optimization methods. Kurdi et al. [17] presented probabilistic roadmap (PRM) path planning method for UGV by using digital map of UAV/UGV cooperative system. However, the existing UAV and USV cooperation systems have not had much intention than UAV/UGV cooperation systems in the research area of path planning. Line-of-sight control is the most widely used control strategy for path planning of UAV/USV system. Nizami et al. [18] presented the first known implementation of rescue boat path planning by using A*, Dijkstra and Breadth-first algorithms in 2012. However, this proposed system has some limitations. For example, the environment is defined as a marine with fully certain islands but uncertain areas in the aerial image are ignored and a flooded urban environment is not tested with the proposed approach. Therefore, further study is needed to improve the performance of path planning to consider uncertainty areas in complex scenarios.

1.2 Contributions

In this research, we consider a flood rescue application where the main goal is to explore an uncertain environment by USV to build a real-time ground map and plan near-optimal paths for a rescue boat. The main contributions of this research are as follows:

1. Using the UAV, a ground image is obtained from above, and then processed to segment obstacles. The segmentation of the obstacles is done by using image processing algorithms such as image denoising and obstacle recognition techniques. This creates a ground map based on the locations of obstacles and the feasible paths. State of the art solutions on map generation does not take

into account the mapping uncertainty. In this study, we propose a novel map building approach where the clearance of obstacles is taken into account along with their corresponding uncertainty levels.

2. Based on the generated ground map, we developed a mission planning algorithm which consists of PRM path planning and decision-feedback building algorithms. The PRM algorithm is used for the rescue boat and as well the USV to achieve desired target points on the generated map. The decision-feedback algorithm is used to get local information from the USV's sensors while following the near-optimal paths to minimize uncertainties for improving the generated ground map.

3. Simulations are performed using a MATLAB based simulator to drive the USV, observe the range sensor readings and USV poses to validate the performance of the proposed approach.

CHAPTER 2

PROPOSED APPROACH

In this paper, we address the problem of mapping an unknown flood environment by recognizing obstacles and roads in the aerial images which are captured by a UAV. We consider a typical flood disaster scenario where the USV can explore the uncertain areas for assisting rescue boat to plan optimal trajectories for reaching the victims.

Figure 2 shows the general system design of the proposed approach. When the UAV transmits the aerial image to the USV, the first step is image denoising to recover the aerial image to remove present noise in the aerial image because the noise will affect the performance of the following steps. After image enhancement, obstacles are extracted as certain and uncertain obstacles and stored in a map structure. Based on the generated ground map, the rescue boat, as well as USV, can plan a path to reach destination points. The first step of path planning is applying the PRM path planning algorithm to find near-optimal paths on the generated ground map. Then, a decision-feedback algorithm is applied to decide which path needs to be followed by the rescue boat and whether USV needs to join rescue operations by removing uncertainty for the purpose of finding a better path for the rescue boat. If the USV needs to join rescue operations, a trajectory following algorithm is applied for trajectory tracking of USV and local information by USV's range sensor are used to enhance the initial generated ground map. The main processes of the proposed approach is shown in Algorithm 1. These processes are detailed in the following sections.

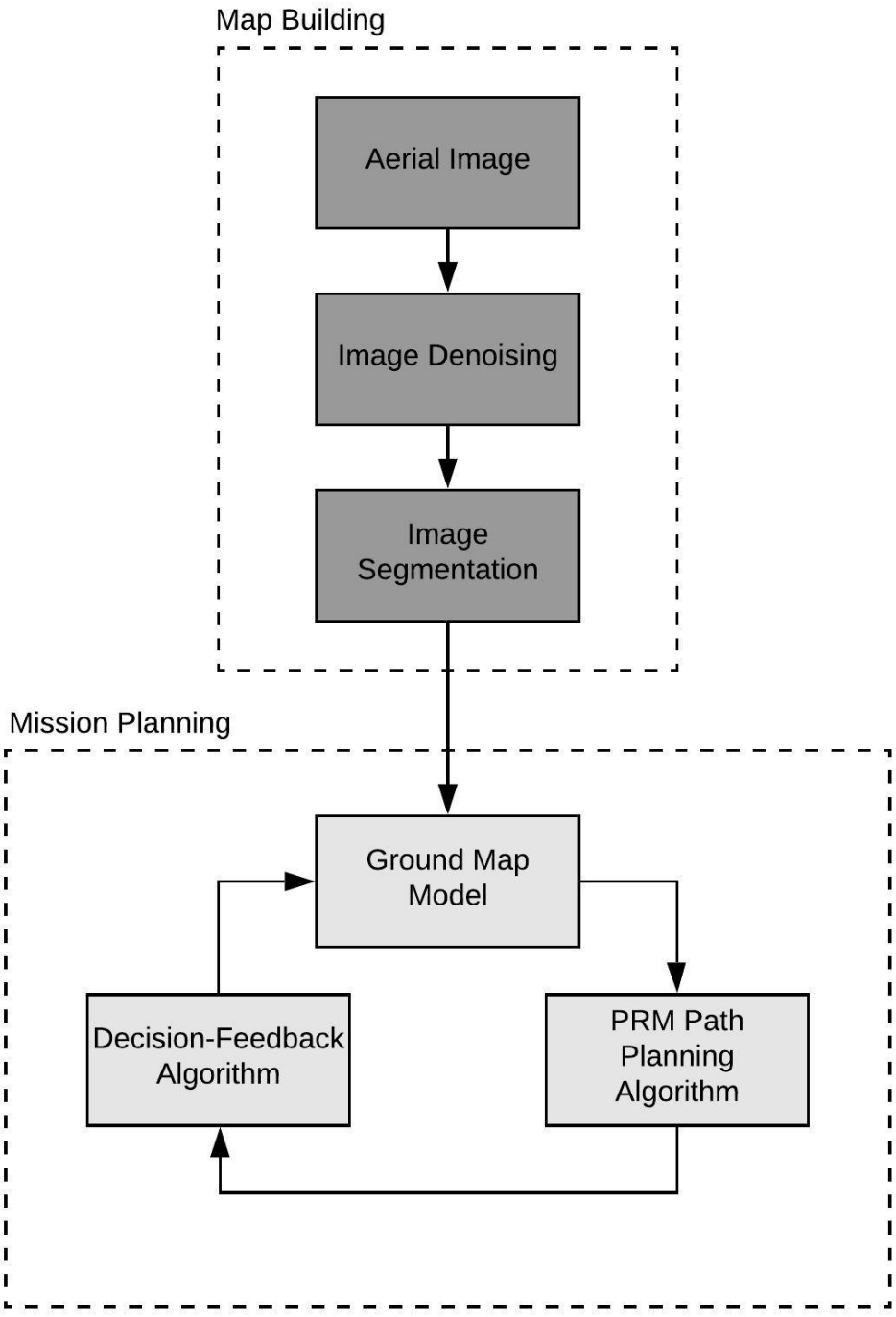


Figure 2.: General system design

Algorithm 1: The Main System Algorithm

Input : A flooded urban environment to perform rescue operations

Output: The near-optimal paths for RB to help rescue operations and an uncertainty cleared ground map

1 **Begin**

2 Record the ground image from aerial vision by UAV

3 Perform image denoising and segmentation algorithms

4 Apply Map Building Algorithm to model a ground map based on the uncertainty level of the obstacles

5 Define the initial and destination points in the ground map map

6 Apply PRM algorithm into the modeled aerial map to find preliminary near-optimal paths which can be followed by the RB and USV

7 Apply Decision-Feedback algorithms to decide whether or not USV needs to explore uncertain areas in the ground map to generate real-time cooperative ground map and to enhance the preliminary path for RB

8 **End**

2.1 Hardware and Software

In this section, we will provide more details about the hardware and software which are used for the USV for testing system algorithms.

2.1.1 Differential (Nonholonomic) Drive USV

We adopted a differential drive robot in MATLAB based 2D Robot Simulator to USV Simulator to validate the performance of the proposed approach [19]. The USV Simulator uses the Pure pursuit path following controller to drive a simulated USV along a predetermined path. At each loop iteration, position of the USV (USV_{Pose} (x, y, theta)) is changing based on the velocity of the USV (S) and an arbitrary refresh period (dt) with respect to Pure pursuit controller which is shown in Equation (2.3), Equation (2.4) and Equation 2.5. The parameters and functions of differential drive USV are defined as follows:

1. USV_{Pose} (x, y, θ): Current pose of USV
2. Linear velocity (V): Current linear forward velocity of USV (m/s)
3. Angular velocity (ω): Current angular velocity of USV (rad/s)
4. Maximum linear velocity (v_{max}): Maximum allowable linear velocity for USV (m/s)
5. Maximum angular velocity (ω_{max}): Maximum allowable angular velocity for USV (rad/s)

$$dx = dt * V * \cos(USV_{Pose}(\theta)) \tag{2.1}$$

$$dy = dt * V * \sin(USV_{Pose}(\theta)) \tag{2.2}$$

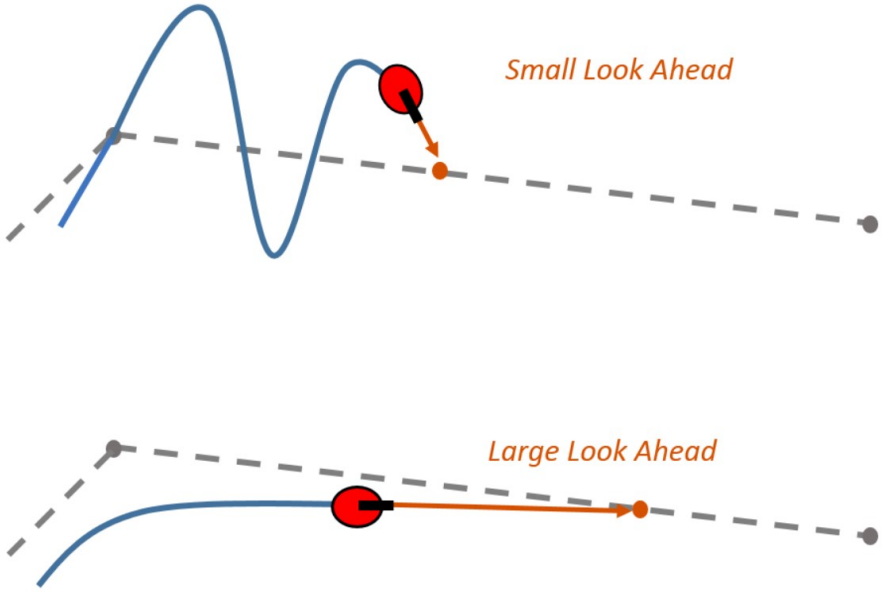


Figure 3.: Pure pursuit algorithm look-ahead distance comparison

$$USV_{Pose}(x) = USV_{Pose}(x) + dx \tag{2.3}$$

$$USV_{Pose}(y) = USV_{Pose}(y) + dy \tag{2.4}$$

$$USV_{Pose}(\theta) = \texttt{wrapToPi}(USV_{Pose}(\theta) + d\theta) \tag{2.5}$$

2.1.2 Pure Pursuit Trajectory Tracking Algorithm

The Pure Pursuit (PP) is a trajectory following algorithm where it computes the angular velocity which is needed for a non-holonomic vehicle to move from its current position to reach some look-ahead point in front of the robot [20]. The angular velocity is based on the robot's current position and the look-ahead distance which decides how far the look-ahead point is placed from the robot's current position. The look-ahead distance is the main tuning property of the PP algorithm. The aim of the look-ahead distance is how far along the path the robot should look from the current location to compute the angular velocity.

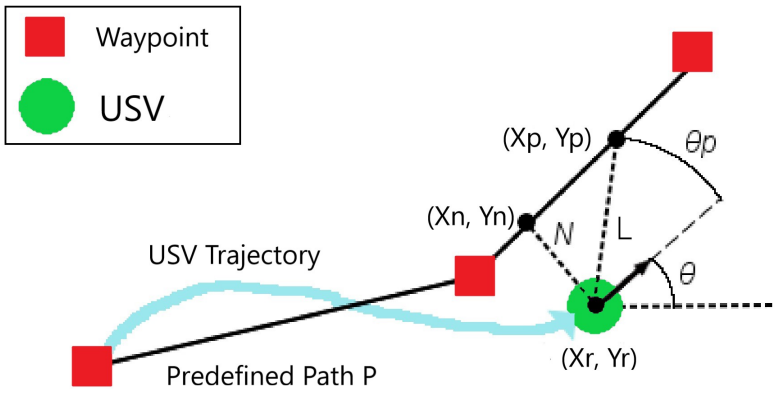


Figure 4.: Geometric explanation of Pure pursuit algorithm

The effect of changing look-ahead distance values can alter how the USV follows the path. A small look-ahead distance can result in oscillating behavior along the path. In order to avoid the oscillating, a larger look-ahead distance can be given but it might cause larger curvature near the corners which is shown in Figure 3 [21].

An overview of the Pure pursuit process is shown in Figure 4. The predefined path P is the set of straight-line sub-paths between waypoints. USV trajectory is the actual movement of USV with respect to P.

The Pure pursuit process is implemented as follows: [20]

1. Read USV's positions to estimate the USV's coordinates, (Xr, Yr), and angle θ with respect to the x-axis.

2. Find the point on the path nearest to the USV (Xn, Yn). This is the starting point to search the line for the furthest point on the line that is within the look-ahead distance. The line between the USV and the nearest point is called N.

3. Search the line for the furthest point (Xp, Yp) within the look-ahead distance. The line between the USV and this point is called L.

4. Compute the new angle, θ_p, needed to direct the USV toward (Xp, Yp).

5. Send θ_p to the USV.

6. Move in that direction for t seconds, where t is an arbitrary refresh rate. In this work, $t = 0.25$ to give the USV time to move before re-calculating and sending a new command.

7. Repeat process until the distance between the goal and the USV is within a specified threshold. This threshold is set because the USV need not touch the exact coordinates of the waypoint. Using pure pursuit, the USV would be continually overshooting the waypoint and retrying to touch the point.

2.1.3 Particle (Holonomic) Drive USV

We proposed a particle drive USV in MATLAB based 2D Robot Simulator to USV Simulator to analyze the performance of the proposed approach. The USV simulator does not use any controller to drive a simulated USV along a predetermined path in contrast to the differential drive USV system. At each loop iteration, position of the USV (USV_{Pose} (x, y)) is changing based on the speed of the USV (S) which is shown in Equation 2.6 and Equation 2.7. The parameters and functions of particle drive USV are defined as follows:

1. USV_{Pose} (x, y): Current position of USV

2. Heading (theta): Current heading of USV

3. Speed (S): Speed of USV (m/iteration)

$$USV_{Pose}(x) = USV_{Pose}(x) + S * \cos(\theta) \tag{2.6}$$

$$USV_{Pose}(y) = USV_{Pose}(y) + S * \sin(\theta) \tag{2.7}$$

2.1.4 Range Sensor Implementation

One of the key elements in any search scenario is to choose a proper sensor for USV. In this research, we decided to use an omni-directional range sensor for USV to explore the environments in a more efficient manner. The omni-directional range sensor has the capability of 360 degrees to sense the environment. Therefore, we decided to mount the omni-directional range sensor at the center of USV. The sensing area coordinates which are OS(x) and OS(y) while the USV exploring are defined in Equation (2.9) and Equation (2.10) below.

$$OS(\theta) = 0 : \frac{\pi}{50} : 2 * \pi \tag{2.8}$$

$$OS(x) = USV_{Pose}(x) + r * cos(\theta); \tag{2.9}$$

$$OS(y) = USV_{Pose}(y) + r * sin(\theta); \tag{2.10}$$

2.2 Map Building

In this paper, we study a novel cooperative map building system where the obstacles will be taken into account, along with their corresponding certainty and uncertainty levels. Map building is the foundation of path planning, which is critical for a rescue boat and USV to reach their destinations accurately. For the rescue boat and USV, the map will be changed dynamically when the vehicle is moving. By contrast, for a UAV, the ground environmental information is nearly unchanging. Therefore, after

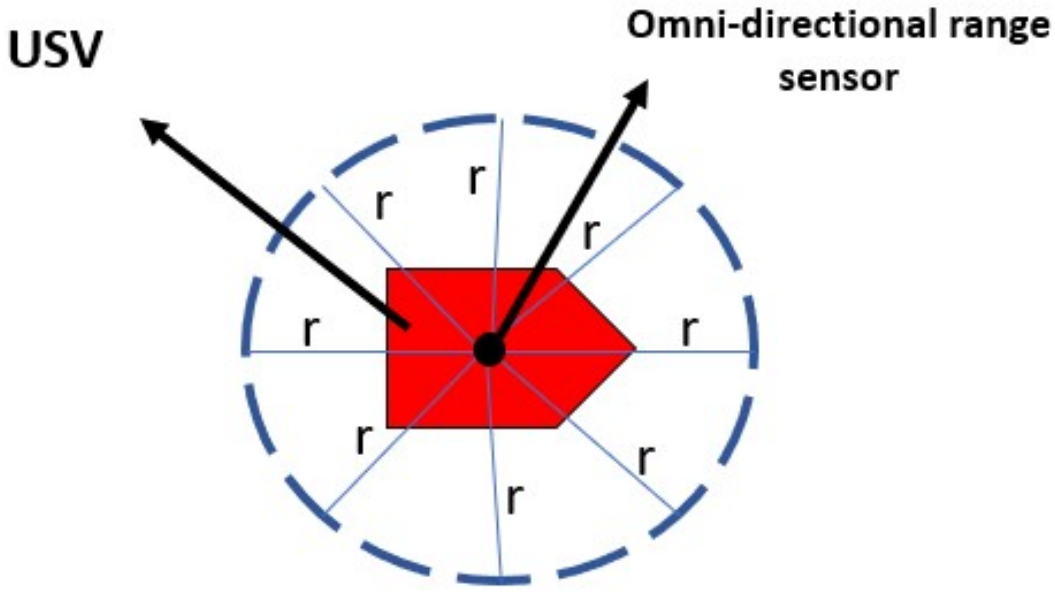


Figure 5.: Omni-directional range sensor

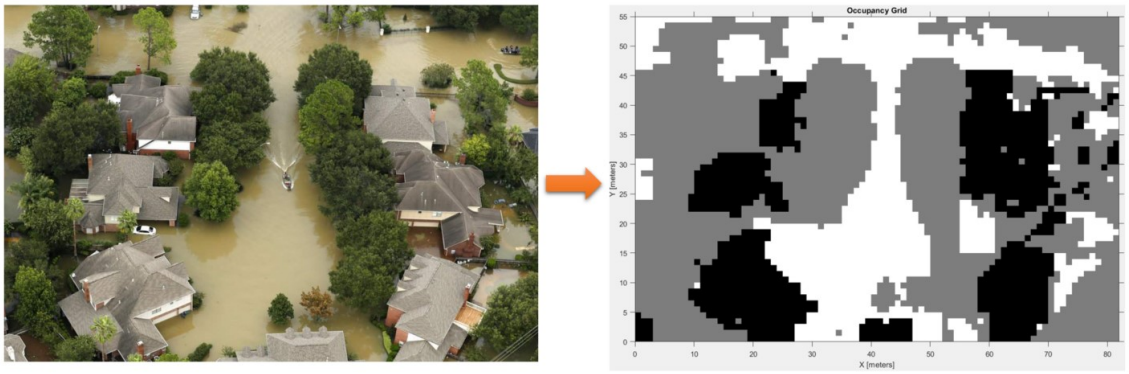


Figure 6.: A ground map model generation

the UAV has collected the ground image, image processing is necessary to build a ground map. Firstly, the median filtering technique is applied to filter possible noise in the aerial image. Then, image segmentation techniques are applied to build a ground map by using the MATLAB Image Segmenter App tools such as flood fill. In the aerial image, certain obstacle areas such as buildings and landmarks are marked as black, collision-free areas are marked as white and uncertain areas such as trees and bridges are marked as gray. After image segmentation, the segmented aerial image is transferred to the occupancy grid map structure. In occupancy grid map, cell value of 1 indicates occupied, 0 indicates collision-free and -1 indicates uncertain which is shown in Table I. Figure 6 shows how a ground map structure looks like after applied image processing techniques into the aerial image.

Environment Definition	Occupancy Grid Representation
Occupied	1
Collision-Free	0
Uncertain	-1

Table I.: Occupancy grid representation of ground map

2.2.1 Map Building Benchmark Algorithm

In this section, we describe a benchmark map building algorithm to compare with our proposed map building algorithm. In our proposed map building algorithm, the obstacles are taken into account with certainty and uncertainty levels. In contrast, traditional image segmentation techniques are taking into account the uncertainty of the obstacles. In this purpose, we analyzed existed image segmentation algorithm to generate a binary ground map where the occupied and collision-free areas are represented as 1 and 0 respectively. Figure 7 shows the general process of image processing for the benchmark map building algorithm. The process of building the ground map by using image processing algorithms illustrated in the following main steps:

1. Image Denoising

   In the step of image denoising, Gaussian filtering technique is applied to filter possible noise in the aerial image.

2. Image Segmentation

   In this step, the filtered image is converted to grayscale and then the canny edge detection technique is applied to extract contours accurately by transforming the grayscale image to binary image. After extracting the contours, erosion and dilation morphological operation techniques are applied to model a ground map by using square structuring elements.

Figure 8 shows the general process of a ground map model generation in the flooded urban environment image by using the benchmark map building algorithm.

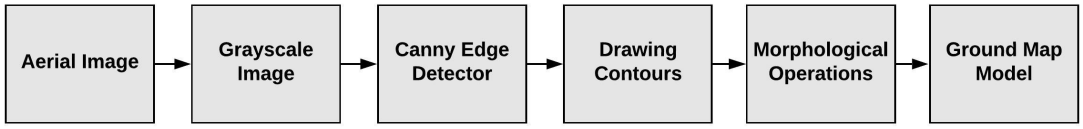


Figure 7.: A ground map model building process

2.3 Path Planning

In this section, we utilize the generated maps and introduce path planning and a trajectory following algorithms. In flood rescue application, one critical issue is making sure rescue vehicles will not collide with the obstacles in the path. In our system, rescue boat and USV should avoid collision with the obstacles while performing their tasks. In order to solve the obstacle avoidance problem, we applied PRM path planning to find collision-free paths. If USV needs to explore uncertain areas in the ground map, USV needs a trajectory following algorithm to visit destination points by following the waypoints in the ground map. In order to solve the trajectory tracking problem of USV, we used Pure pursuit trajectory following algorithm for USV to follow the collision-free paths.

2.3.1 Probabilistic Roadmap Path Planning Algorithm

Probabilistic roadmap path planning algorithm is a sampling-based path planning technique which consists of two stages: a construction and a query stage [22]. The goal of the construction stage is to randomly draw a graph (roadmap) across the environment. All edges and vertices of the roadmap should be collision-free so that the RB and USV can use the roadmap for their task planning. The PRM selects a number of random nodes in the work-space as the vertices where the nodes must not lie inside of the obstacles. If the number of nodes high, the results would be better

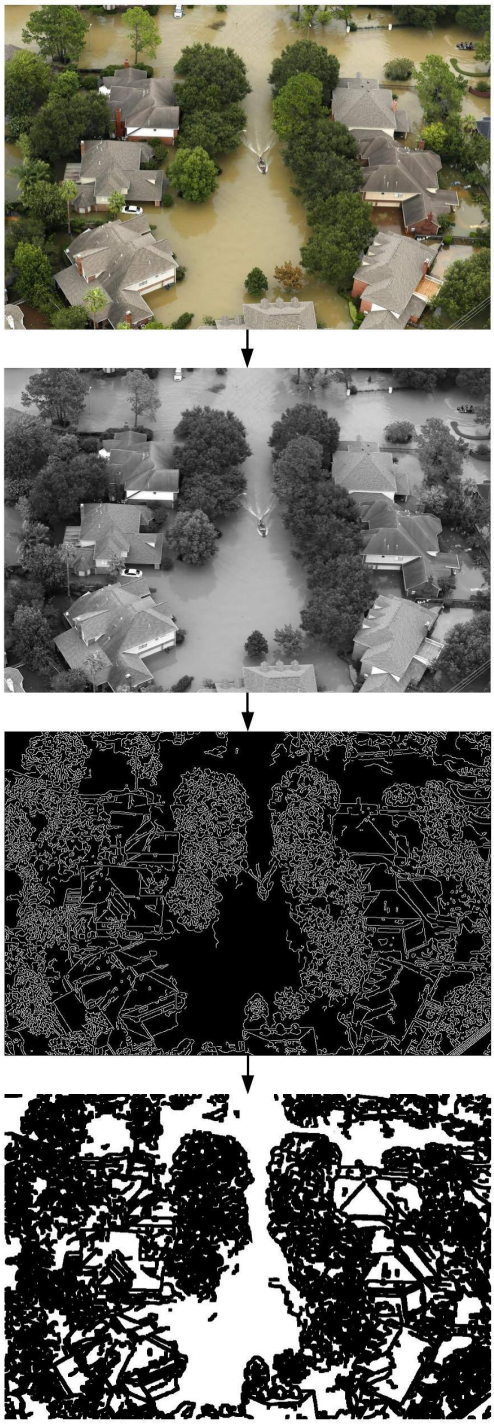


Figure 8.: A ground map model generation by benchmark algorithm

in contrast, the computation time would be high. Then, the algorithm connects all pairs of randomly selected vertices. If any two vertices can be connected by a straight line, the straight line becomes an edge which is shown in Figure 9.

The goal of the query stage is to use the roadmap which is developed earlier for finding the shortest path for the rescue boat and USV. The distance between each node and the position of the nodes should be considered to find the shortest path. Therefore, a cost function which is the same as in the graphical search algorithms such as A* algorithm should be applied in this purpose. [23]. The cost function consists of heuristic and historic functions. The heuristic function, H, stores the weights of the edges, which are taken as the Euclidean distance between the connecting points, as shown in Equation (2.11):

$$H(x_i, y_i, x_j, y_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2.11}$$

The historic nearness function, N, which determines the nearness of the point to the goal while finding the Euclidean distance between a point and the goal, as shown in Equation (2.12):

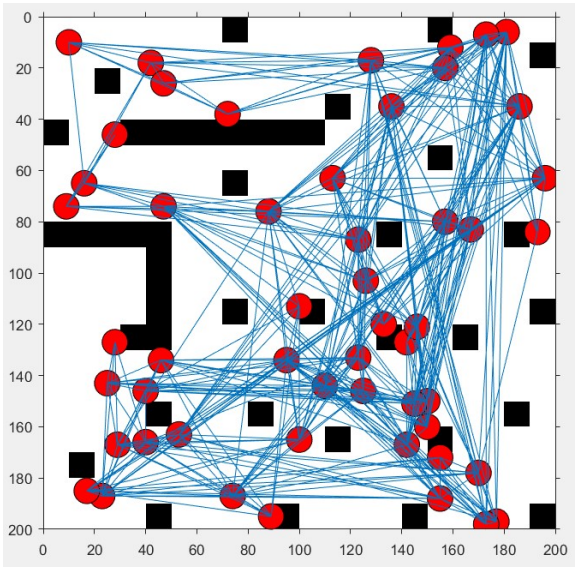
$$N(x_i, y_i, G_{x_j}, G_{y_j}) = \sqrt{(x_i - G_{x_j})^2 + (y_i - G_{y_j})^2} \tag{2.12}$$

The cost function, C, is summation of heuristic and historic functions which is show in Equation (2.13):

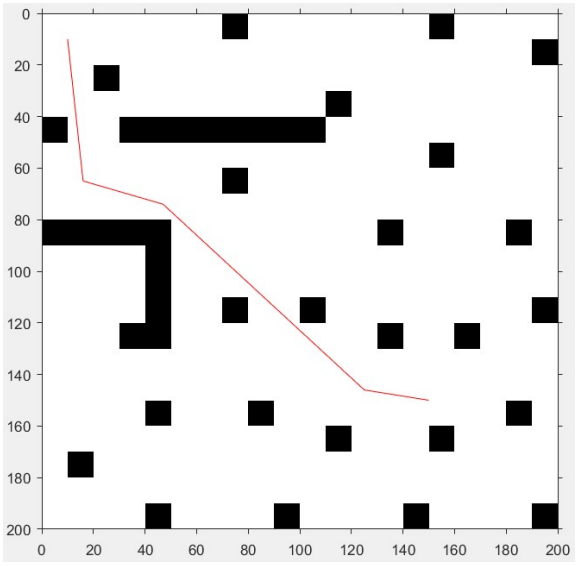
$$C = H(x_i, y_i, x_j, y_j) + N(x_i, y_i, G_{x_j}, G_{y_j}) \tag{2.13}$$

2.3.2 Benchmark Algorithms

In this section, we describe the benchmark algorithms for comparisons with the PRM path planning algorithm.



(a)



(b)

Figure 9.: (a) Roadmap (b) Path Derived

2.3.3 Path Planning using A* Algorithm

A* path planning algorithm is a standard graph search based technique. The A* algorithm takes a graph as input and explores all the regions to find the shortest path from the initial point to the destination points in the explored regions. The A* algorithm is heuristic based and works hierarchically which means that all the near regions are explored before the further ones while the exploration is also biased towards the regions closer to the destination points [23].

The A* algorithm would take a very large computation time on a high-resolution map. Therefore, the map resolution is needed to be reduced by the A* algorithm, while the resolution is an input parameter of the algorithm. However, the higher resolution map displays better results while it leads to excessive computation time.

In general, a graph consists of vertices and edges. Each pixel of the reduced resolution map is taken as a vertex in this algorithm and each vertex has a number of connections which act as edges. In Figure 10 [24], the possible connections are given for any general position of the vehicle which is represented as a matrix as shown.

In the matrix, the current position of the vehicle is marked as '2'. There needs to be only one current position of the robot which means that only one '2' should be in the matrix. All possible moves are represented by 1 and all impossible moves are represented by 0.

The connection matrix is an input parameter of the A* algorithm and we can create our own matrices to test the efficiency of the system. There are three typical connection matrices which are shown in Figure 11 [24]. Figure 11(a) only allows the vehicle to take linear moves such as up, down, left and right. Figure 11(b) allows the vehicle to take four different diagonal moves together with the four linear moves.

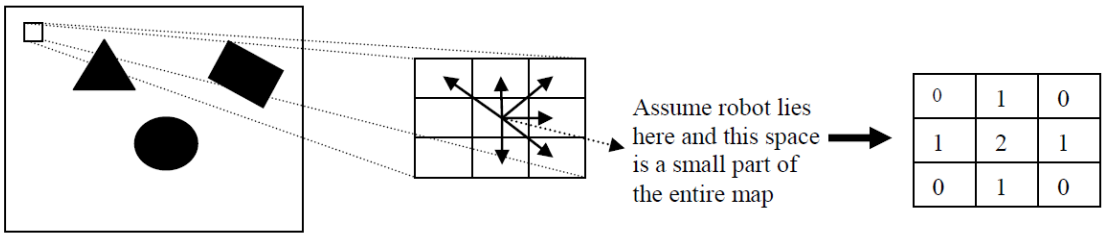


Figure 10.: A* algorithm connection matrix

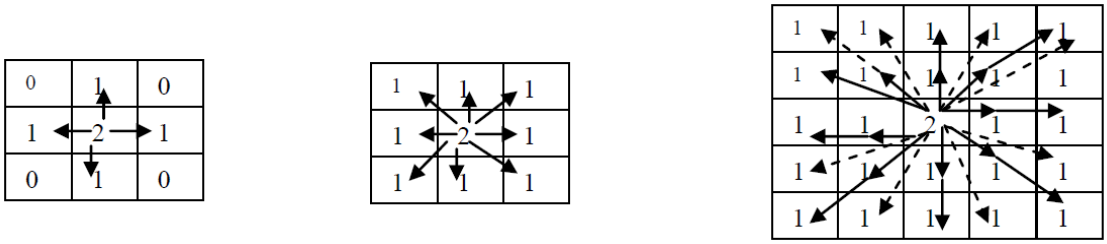


Figure 11.: A* algorithm connection matrices

Figure 11(c) allows the vehicle to make more moves while adding connections between the diagonal moves. Allowing more movements for the vehicle by adding more connections can help to generate a better path. However, adding more connections may result in excessive computation time. Another design specification of the algorithm is a cost function. The cost function stores the weights of the edges, which are taken as the Euclidean distance between the connecting points which are nearness points to the destination.

There are some disadvantages and advantages of using A* algorithm in the path planning purpose. The A* algorithm is simple and efficient. However, it has excessive computation time in large scenes and it generally creates unnatural motions such as sharp turns.

2.3.4 Path Planning using GA Algorithm

GA is a meta-heuristic search algorithm which applies Darwin's principle of natural selection to model path planning problems. In problem modeling, we need a fitness function and specification of variables of that fitness function. A path can be characterized by a fixed number of points on the map. In order to make some path from this set of waypoints, we start from the source and connect it to the first waypoint by a straight line. The first waypoint is connected to the second waypoint by a straight line, and so on. At the end, the last waypoint is connected to the goal. Distance minimization is computed using the Euclidean distance (Equation 2.14) between each pair of waypoints to measure the paths length (Equation 2.15) is the summation of each sub-path between adjacent waypoints.

$$D(x_i, y_i, x_j, y_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2.14}$$

$$Path_D = \sum_{i=1}^{n-1} DIST(< x_i, y_i >, < x_{i+1}, y_{i+1} >) \tag{2.15}$$

A heavy penalty (P) is added if any part of the path lies inside the obstacle, while the penalty is proportional to the length of the path inside the obstacle defined which is S_{INF} as show in Equation 2.16. The fitness function is comprised of the distance minimization and obstacle avoidance functions as shown in Equation 2.17.

$$Path_O = S_{INF} * P \tag{2.16}$$

$$Fitness_{path} = Path_D + Path_O \tag{2.17}$$

The locations of each of these fixed number of points in x and y coordinates are the optimization variables. The variable bounds are such that the waypoint lies inside the map (lower bound 1 and upper bound as the length/width of the map

for the X/Y axis). All points put one by one makes the genetic individual used for optimization.

Each waypoint in the path marks a waypoint of turn. The total number of points is an algorithm parameter and should be equal to the maximum number of turns a robot is expected to make in the robot map. Setting this number too high would result in very large computational requirements. If the algorithm is not allowed a large computational time, random results may be the output. Setting a large value in simple scenarios will result in useless turns and hence a high path length. A too small value of the parameter may not give enough flexibility to the algorithm to model the optimal path, thus resulting in collision-prone paths.

2.3.5 Decision-Feedback Algorithm

The main idea of proposing the decision-feedback algorithm for ASV in USV purpose is to check whether or not SP is updated while the USV is searching the uncertainty areas other than reaching the goal autonomously while reducing the uncertainty. Based on the generated aerial map, uncertain areas are needed to be searched for reducing uncertainty by USV if it is necessary. In the decision-feedback building algorithm, the generated ground map is divided into two different map structures which are Uncertainty-blocked map where the all uncertain areas are assumed as obstacles and Uncertainty-free map where the all uncertain areas area assumed as collision-free where the example is shown in Figure 12. Then, we compare two different collision-free path options which are generated by the path planning module to decide or not USV needs to explore uncertain areas in the generated map. The Safest Path (SP) which is generated from Uncertainty-blocked map where the rescue boat should avoid from obstacles and uncertain areas and the Most Efficient Path (MEP) which is generated from Uncertainty-free map where the rescue boat should

avoid from only obstacles between defined starting and destination points where the example is shown in Figure 13-14.

In this algorithm, we calculate Predicted Enhancement of SP (PE_{SP}) based on the length of SP (L_{SP}) and MEP (L_{MEP}) which is shown in Equation (2.18):

$$PE_{SP}(L_{SP}, L_{MEP}) = \begin{cases} \left|\frac{L_{SP}-L_{MEP}}{L_{SP}}\right|, & \text{if } L_{SP} \neq 0 \text{ and } L_{MEP} \neq 0 \\ 0, & \text{if } L_{SP} = 0 \text{ and } L_{MEP} = 0 \\ 1, & \text{if } L_{SP} = 0 \text{ and } L_{MEP} \neq 0 \end{cases} \tag{2.18}$$

We defined a threshold value whether or not USV needs to start searching uncertain areas. In this purpose, the threshold value is defined as 0.25. If the PE_{SP} is smaller than 0.25, it means that USV does not have to start or continue exploring uncertain areas. If PE_{SP} is larger than 0.25 or the SP is not found while the MEP is found, USV should start exploring the uncertain areas to enhance the ground map. Then, we define collision-free waypoints (WPs) on the map which are around the intersection points (IPs) between MEP and the uncertain areas in the sense of USV omni-directional range sensor (OS) which is shown in Equation (2.19):

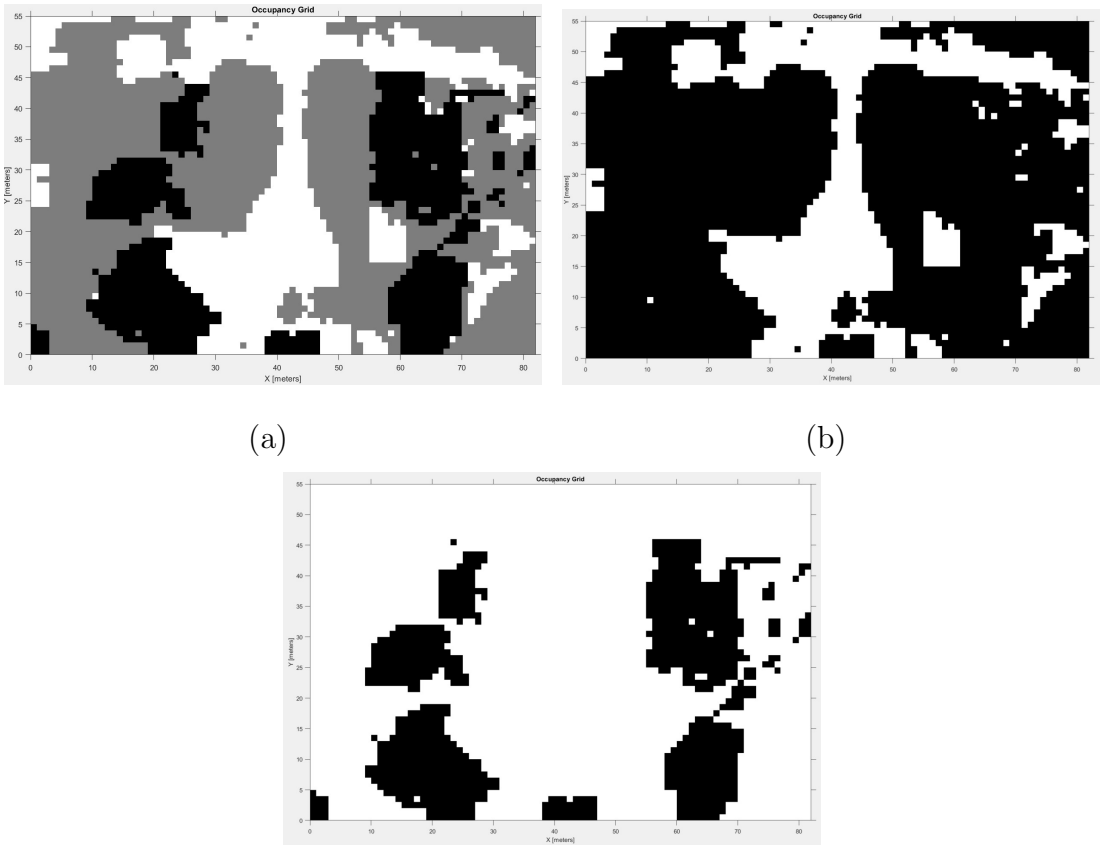
$$WPs(x, y) = IPs(x \mp OS(x), y \mp OS(y)) \tag{2.19}$$

While the USV reaches the WPs, the ground map updates by omni-directional range sensor's local information in the real time as shown in Equation (2.20). When the USV arrives the first WP, then we need to check whether or not SP is enhanced in the updated ground map.

$$Ground_{MAP}(O(x), O(y)) = GroundTruth_{MAP}(O(x), O(y)) \tag{2.20}$$

If the uncertainty is cleared as obstacles or collision-free, the MEP and SP need to be

updated between initial and destination points in the updated ground map. Then, we need to recalculate PE between the updated SP and MEP. If the updated PE is smaller than 0.25, it means that USV does not have to continue its mission because the threshold value is achieved. If the PE is larger than 0.25, then USV needs to visit new waypoints which are around the intersecting points between updated MEP and the uncertain areas in the range of USV sensor. The example scenario in the step of assigning WPs and updation PE during the USV simulation is shown in Figure 15-21. The main flow of the decision-feedback map building algorithm for USV is shown in Figure 22.



(a) (b)

(c)

Figure 12.: (a) Ground map, (b) Uncertainty-blocked map, (c) Uncertainty-free map

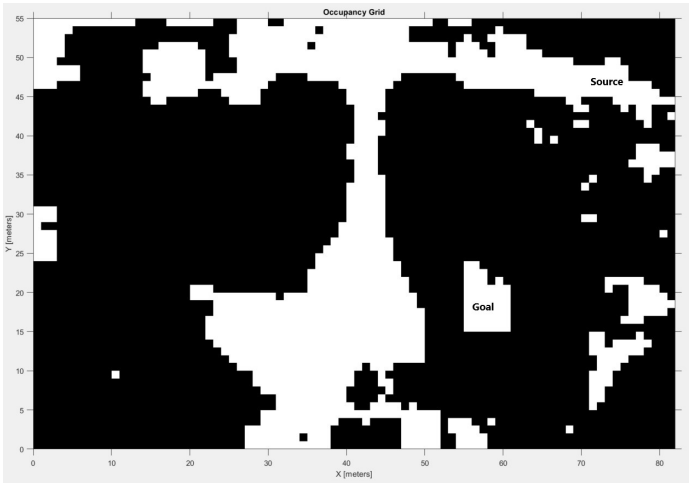


Figure 13.: SP is not found in the Uncertainty-blocked map

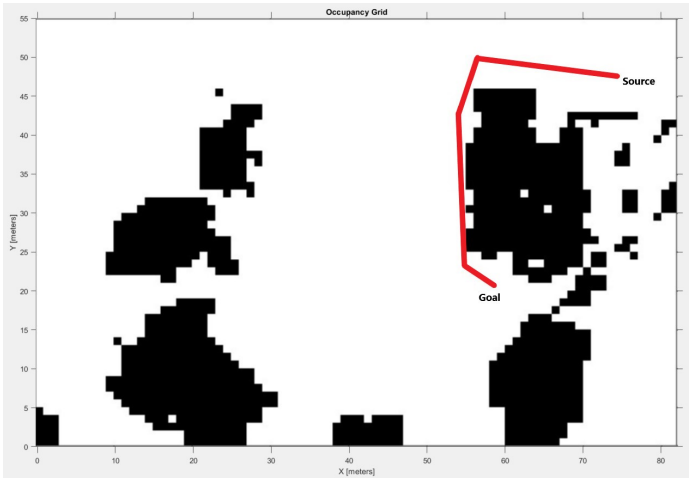


Figure 14.: MEP is found in the Uncertainty-free map

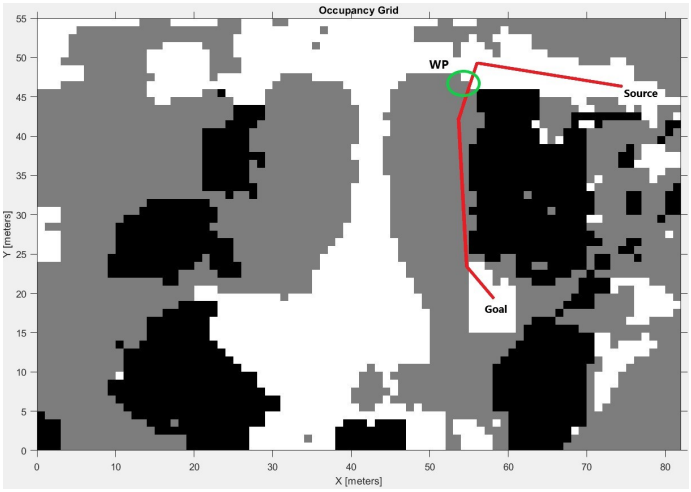


Figure 15.: Assigning WP 1

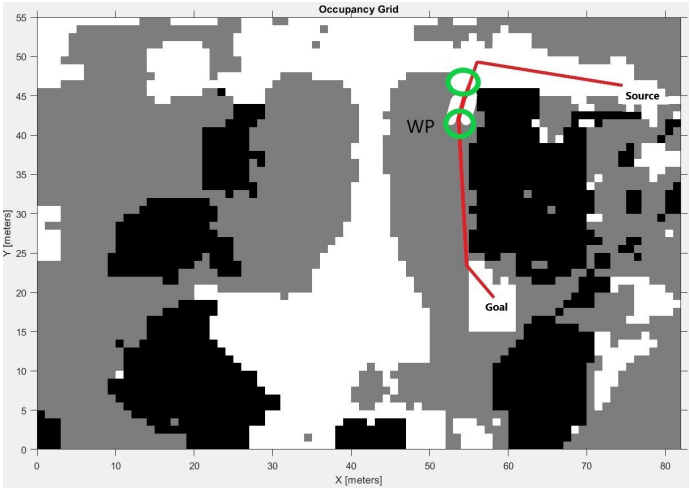


Figure 16.: Assigning WP 2

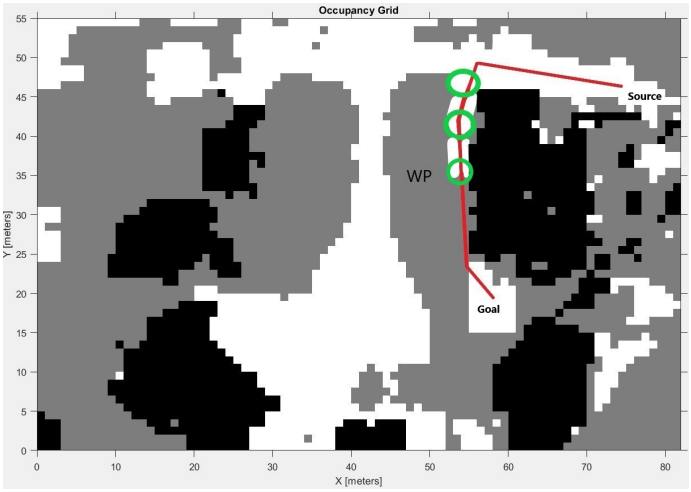


Figure 17.: Assigning WP 3

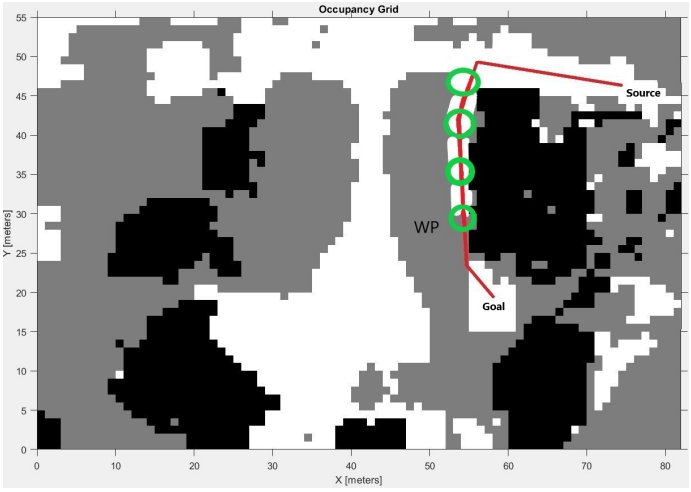


Figure 18.: Assigning WP 4

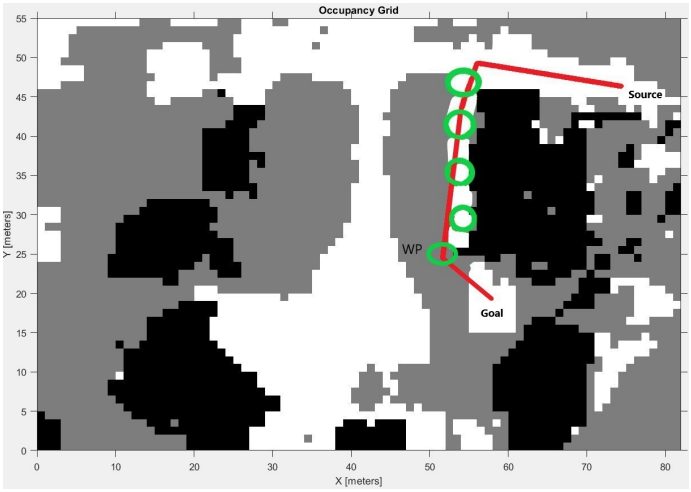


Figure 19.: Assigning WP 5

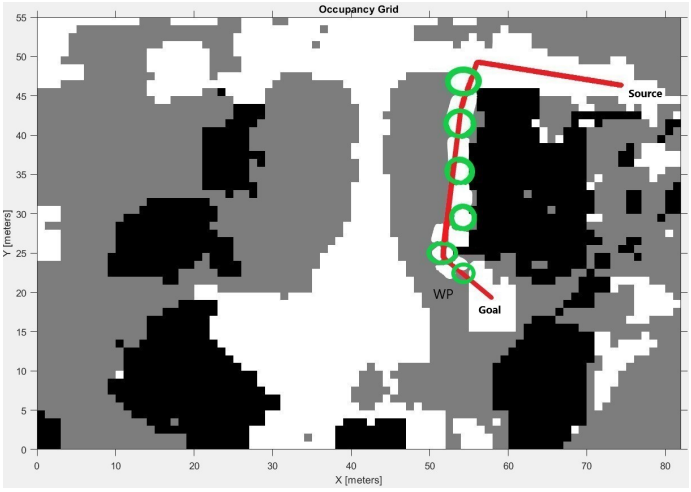


Figure 20.: Assigning WP 6

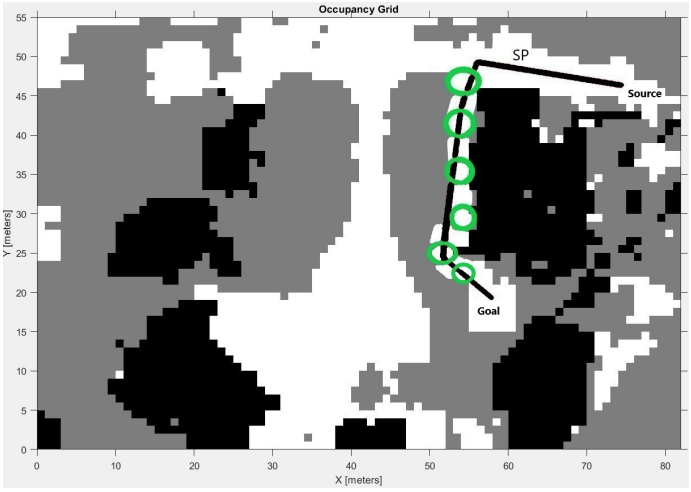


Figure 21.: Updating SP

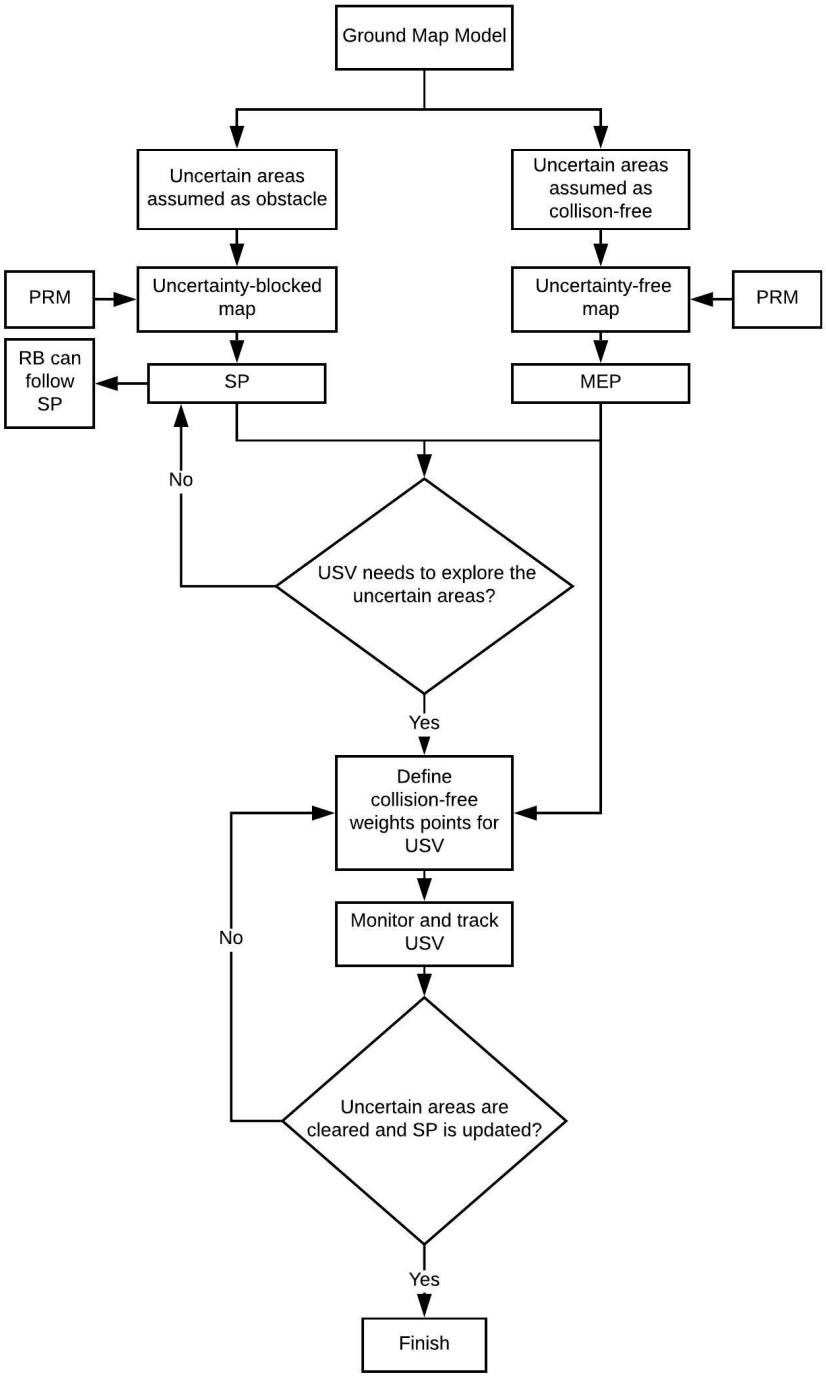


Figure 22.: Decision-feedback algorithm

CHAPTER 3

EXPERIMENTS AND EVALUATION

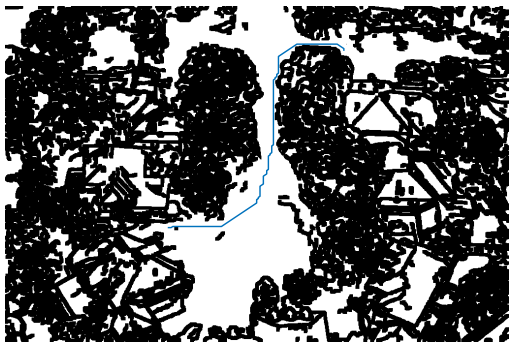
To evaluate the proposed system, a set of simulation experiments with different environmental setups are performed. Firstly, the PRM path planning algorithm is compared with benchmark path planning algorithms which are A* and GA to validate the performance. The simulation experiments were implemented using MATLAB with Robotics System Toolbox. For the simulation experiments, the full system of the map building, decision-feedback, path planning and trajectory tracking algorithms is tested using user-defined occupancy grid maps and real flood environment images.

3.1 Path Planning Algorithms Testing

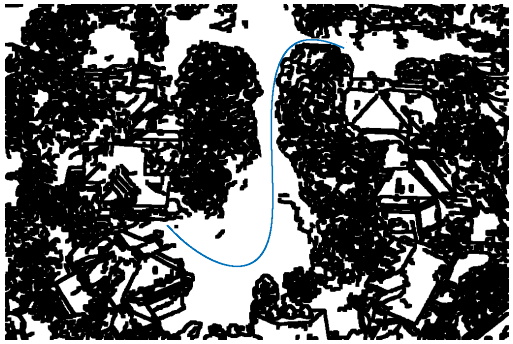
In this section, we analyzed the effectiveness of PRM, A* and GA path planning algorithms. In our experiment, we used an aerial image which is taken from Houston in Hurricane Harvey (2017) [25]. Then, the image is segmented by using proposed benchmark image processing techniques to model a ground map.

After building the ground map, path planning algorithms are applied to find near-optimal paths for rescue boat between source and destination points. The parameters used for each algorithm in our experiments are shown in Table II. The starting point is defined as (690, 90) and the destination point is defined as (330, 450) as x and y coordinate points respectively for all the path planning algorithms. The planning paths are indicated by the blue curve. We can observe the planning paths for each algorithm from Figure 23.

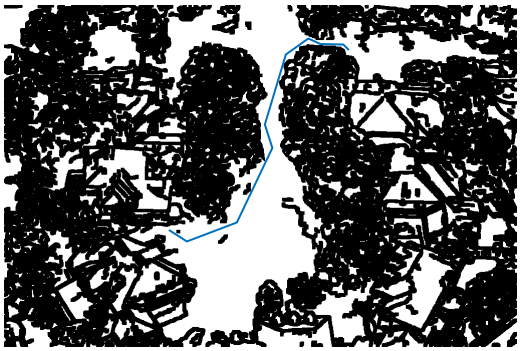
According to the results in Table III, it can be seen that the generated path in A* is shorter than the generated path in GA even though computation time for GA is better than A. PRM algorithm gives the shortest path and best computation



(a)



(b)



(c)

Figure 23.: (a) A* (b) GA (c) PRM

Algorithm	Source(x,y)	Destination(x,y)	Population size	Number of Generations	Number of nodes
A*	(690, 90)	(330, 450)	N/A	N/A	N/A
GA	(690, 90)	(330, 450)	80	60	N/A
PRM	(690, 90)	(330, 450)	N/A	N/A	400

Table II.: Path planning algorithm parameters

time in these three algorithms. Therefore, we have applied the PRM algorithm in our simulation tests as a part of system algorithms to validate the performance of the system algorithms most efficently.

Algorithm	Path Length(m)	Computation time(sec)
A*	666	158
GA	907	92
PRM	653	35

Table III.: Path planning algorithm comparison results

3.2 User-defined Occupancy Grid Map Testing

There are some different size of user-defined map structures are generated to test system algorithms. Firstly, Map Model #1 with the dimension of 20*20 and the resolution of 20 which is shown in Figure 24 is generated. According to the decision-feedback algorithm, the generated aerial map is divided into two different map structures to compare L_{SP} and L_{MEP} which is shown in Figure 25. According to the preliminary path planning results which are shown in TableIV, the L_{SP} is found as 645.4 m, in contrast, L_{MEP} is found as smaller than L_{SP} with 240.3 m and PE_{SP} is found as 0.63. Therefore, USV needs to explore uncertain areas to enhance the

SP for the RB while visiting the WP which is shown in Figure 26. The simulations are tested in the user-defined ground truth map by using the differential drive and particle drive USV and the results are showing that USV successfully searched uncertain areas and the aerial map is updated based on the USV sensor results which are shown in Figure 27. According to the results in Table II, the SP is updated and the PE_{SP} is found as 0.26 after USV is completed its task. Therefore, RB can reach the destination point in a more efficient manner by following the updated SP which is shown in Figure 27.

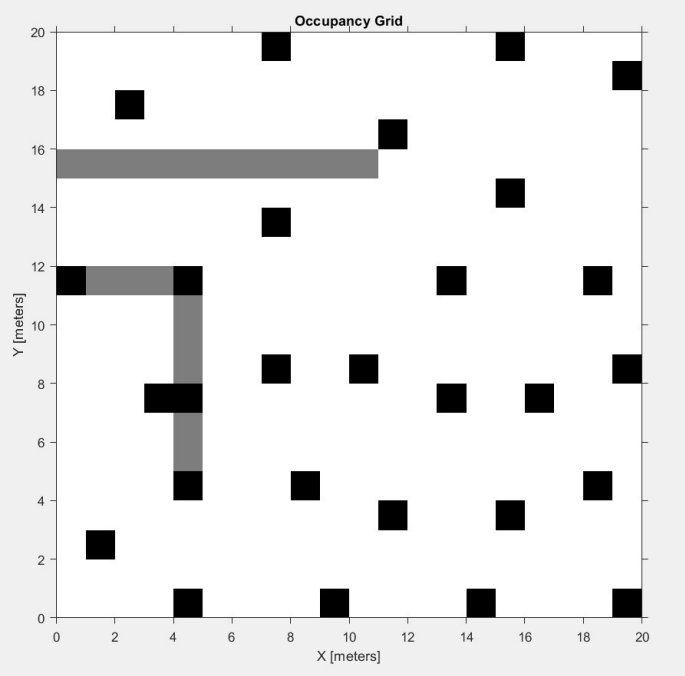
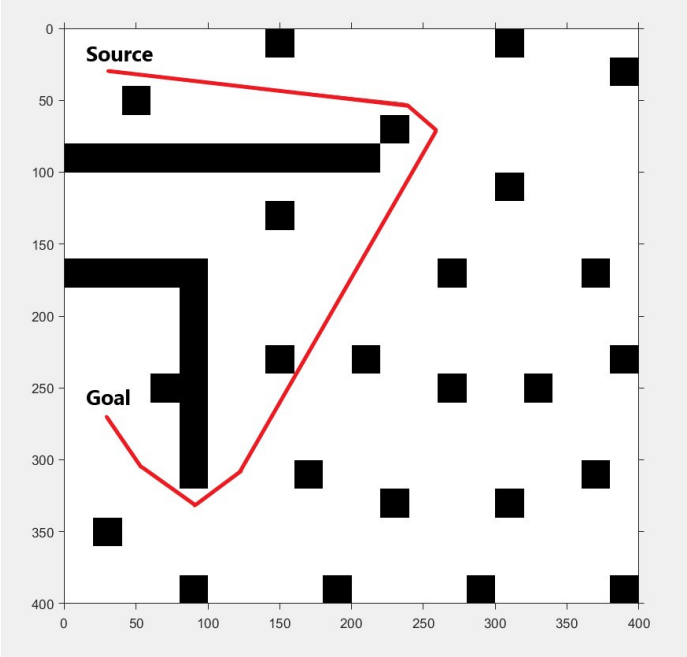


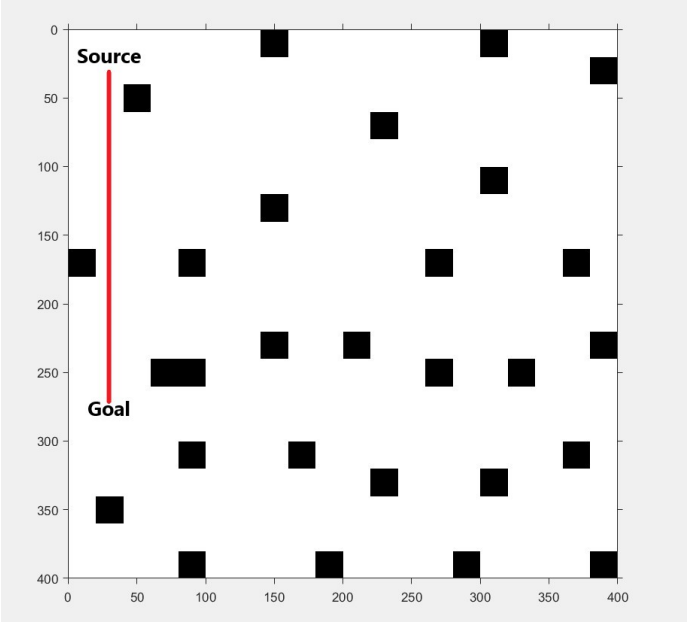
Figure 24.: Map Model #1

Test Case	L_{SP}	L_{MEP}	PE_{SP}
Map Model #1	645.4	240.3	0.63

Table IV.: Preliminary path planning results in Map Model #1



(a)



(b)

Figure 25.: a)SP with length 645.4 m b)MEP with length 240.3 m

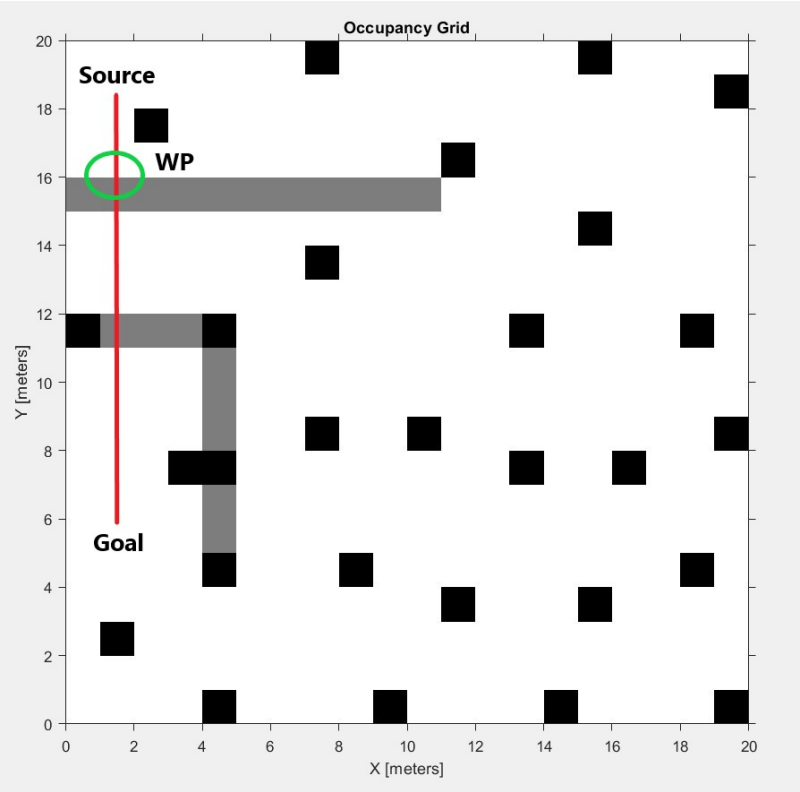


Figure 26.: Assigning a WP in Map Model #1

As it is mentioned above, the SP was found in the stage of the pre-path planning for Map Model #1. However, an SP might not exist in some cases because a destination point might be located into the uncertain areas or some uncertain areas might be blocking all the possible paths to reach the destination point. In Figure 28, Map Model #2 with the dimension of 20*20 and the resolution of 10 is generated to test the system algorithms.

According to the pre-path planning results which are shown in Table III, the SP is not found, in contrast, MEP is found with the length of 371.2 and the PE_{SP} is found as 1. Therefore, USV needs to explore uncertain areas to find a possible SP for the RB while visiting the WPs which is shown in Figure 29. The simulation is tested on the generated ground truth map which is shown in Figure 30. In Figure 30,

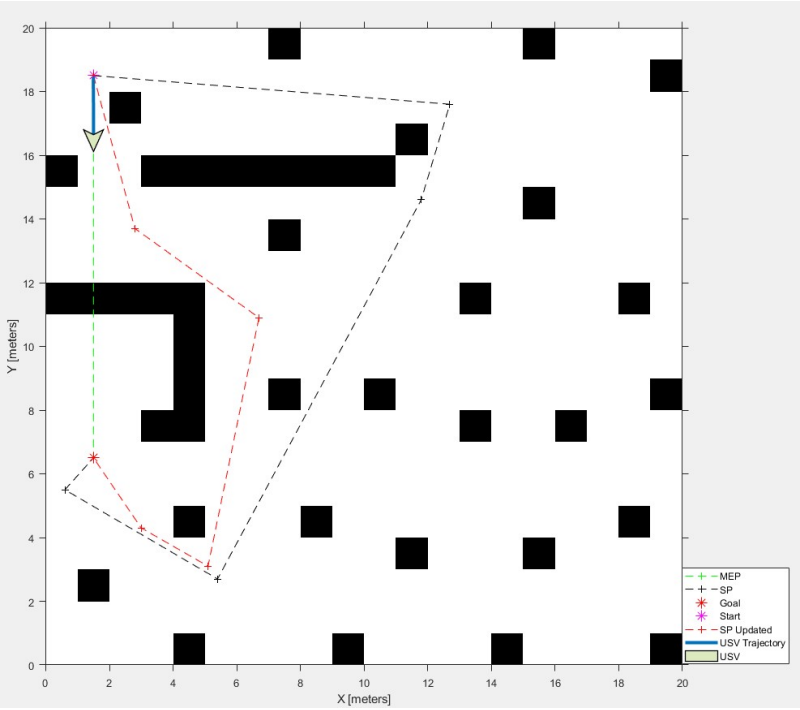


Figure 27.: USV simulation in Map Model #1

USV Simulation	Updated L_{SP}(m)	PE_{SP}	Computation Time(s)
Differential Drive	341.2	0.26	3
Particle Drive	341.2	0.26	0.3

Table V.: USV simulation comparison results in Map Model #1

the simulation results are showing that USV successfully searched uncertain areas and the aerial map is updated with the sensor readings. According to the results in Table IV, the SP is found and the PE_{SP} is found as 0 after USV is completed their mission .

As we have seen the previous two models, USV is needed to search uncertain areas to enhance the aerial map. However, USV does not have to explore uncertain areas in some cases if PE_{SP} does not satisfy the threshold value which is 0.25. In

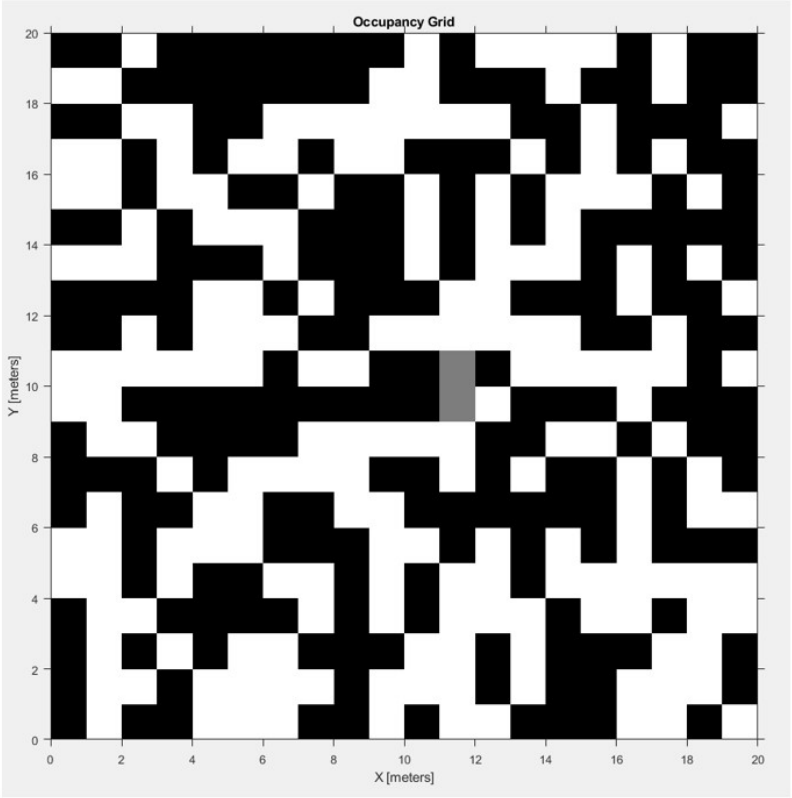


Figure 28.: Map Model #2

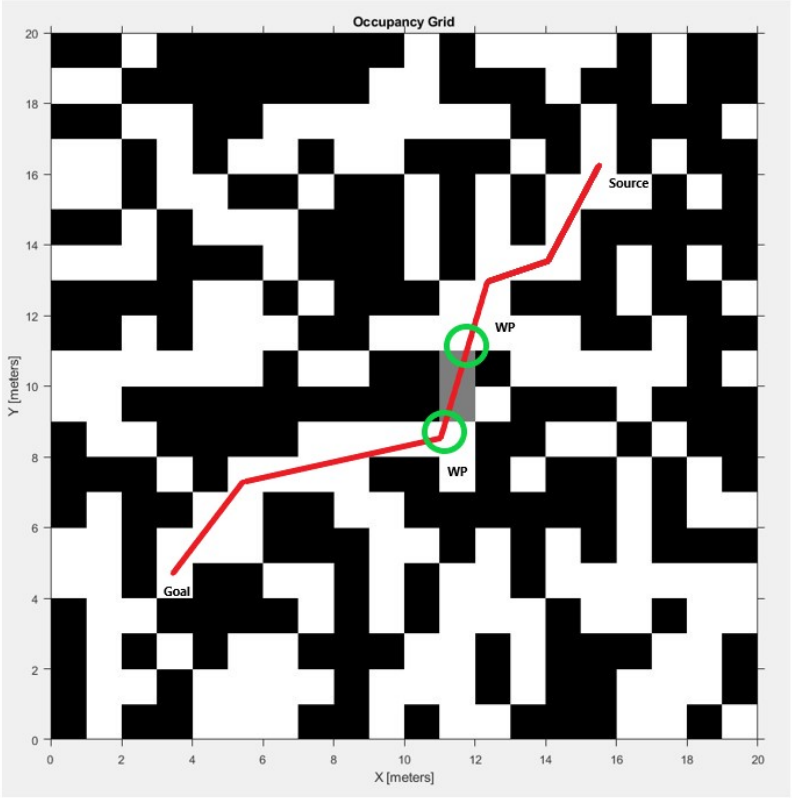


Figure 29.: Assigning WPs in Map Model #2

Map Model	L_{SP}	L_{MEP}	PE_{SP}
Map Model #2	Not found	271.2	1

Table VI.: Preliminary path planning results in Map Model #2

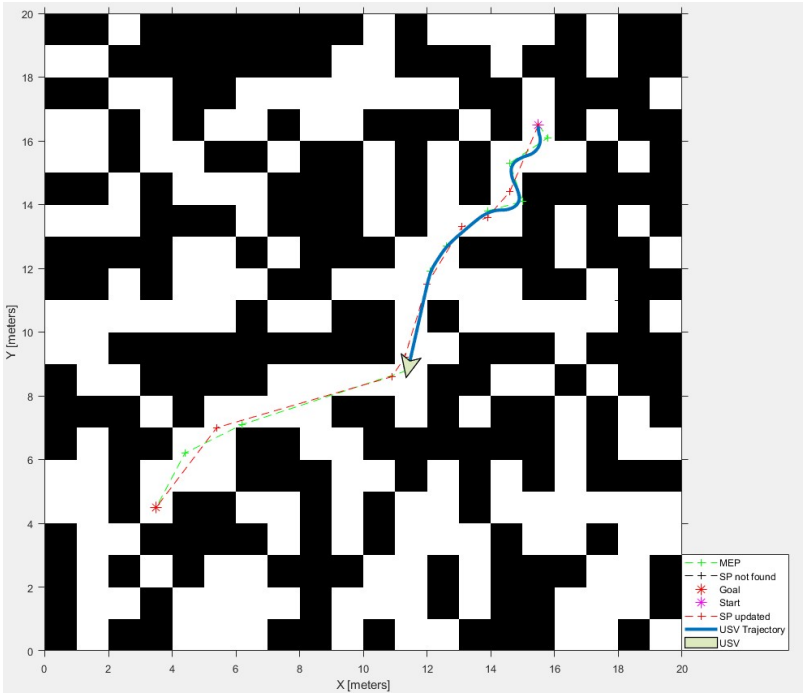


Figure 30.: USV simulation in Map Model #2

Figure 18, Map Model #3 with the dimension of 20*20 and the resolution of 10 is generated to test the system algorithms. According to the preliminary path planning results which are shown in Table VI, the L_{SP} is found as 247.5 m, in contrast, L_{MEP} is found as of 242.3 m and PE_{SP} is found as 0.036 which is less than 0.25. Therefore, USV does not have to explore uncertain areas to find a possible SP for the RB as it shows in Figure 32.

As we have seen in three map models, the simulation results of USV are showing that SP is found when it was not found previously or enhanced after reducing

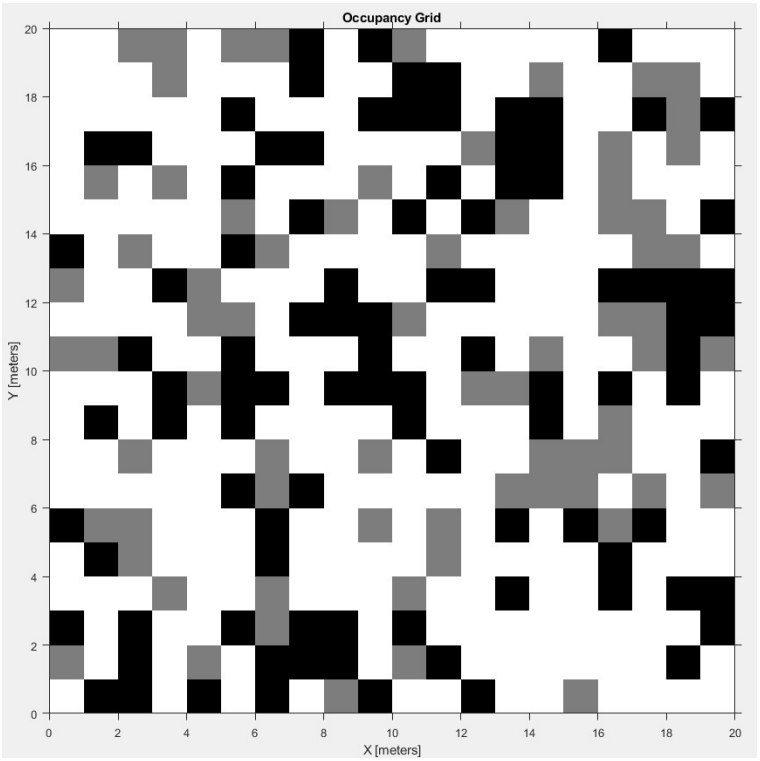


Figure 31.: Map Model #3

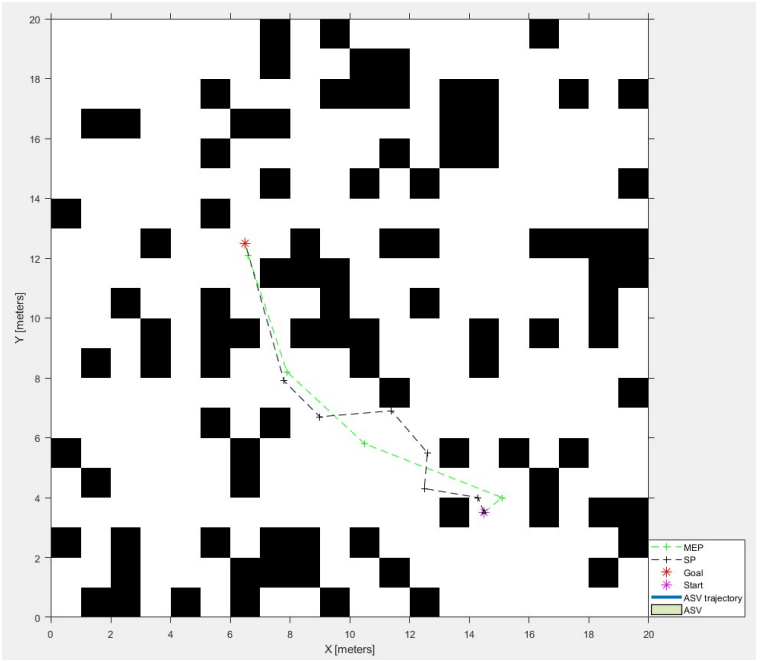


Figure 32.: USV simulation in Map Model #3

USV Simulation	Updated SP Length(m)	PE_{SP}	Computation Time(s)
Differential Drive	271.2	0	8.6
Particle Drive	271.2	0	1.4

Table VII.: USV simulation comparison results in Map Model #2

Map Model	L_{SP}	L_{MEP}	PE_{SP}
Map Model #3	247.5	242.3	0.036

Table VIII.: Preliminary path planning results in Map Model #3

the uncertainty. However, USV might not find or update the SP in some cases. In Figure 33, Map Model #4 with the dimension of 40*40 and resolution of 10 is generated. According to the preliminary path planning results which are shown in Table IX, the L_{SP} is found as 285.6, in contrast, L_{MEP} is found with the length of 155.8 and PE_{SP} is found as 0.48%. Therefore, USV needs to explore uncertain areas to find a possible SP for the RB while visiting the WP which is shown in Figure 34. According to the results in Table X, the SP is not updated after ASV and USV are completed their mission but USV is completed its task in a short time.

Map Model	L_{SP}	L_{MEP}	PE_{SP}
Map Model #4	285.6	155.8	0.48

Table IX.: Preliminary path planning results in Map Model #4

3.3 Flooded Urban Environment Image Testing

In flooded urban environment image testing, some aerial images which are taken by UAV from the flood areas in Houston [25] after Hurricane Harvey (2017) is analyzed

Figure 33.: Map Model #4

USV Simulation	Updated SP Length(m)	PE_{SP}	Computation Time(s)
Differential Drive	0	0	4.6
Particle Drive	0	0	0.75

Table X.: USV simulation comparison results in Map Model #4

to apply the system algorithms. Figure 36(a) displays the aerial image with the dimension of 1920×798 is tested in our system. Firstly, image processing techniques are applied to build a ground map. In the process of ground map generation, trees and bridges are assumed as uncertain areas, roads are assumed as collision-free paths and the remaining areas in the map are considered as certain obstacles which are shown in Figure 36(b). Then, we modeled a ground truth map to test our system in the simulation environment. Google Maps is used as a reference to build a ground

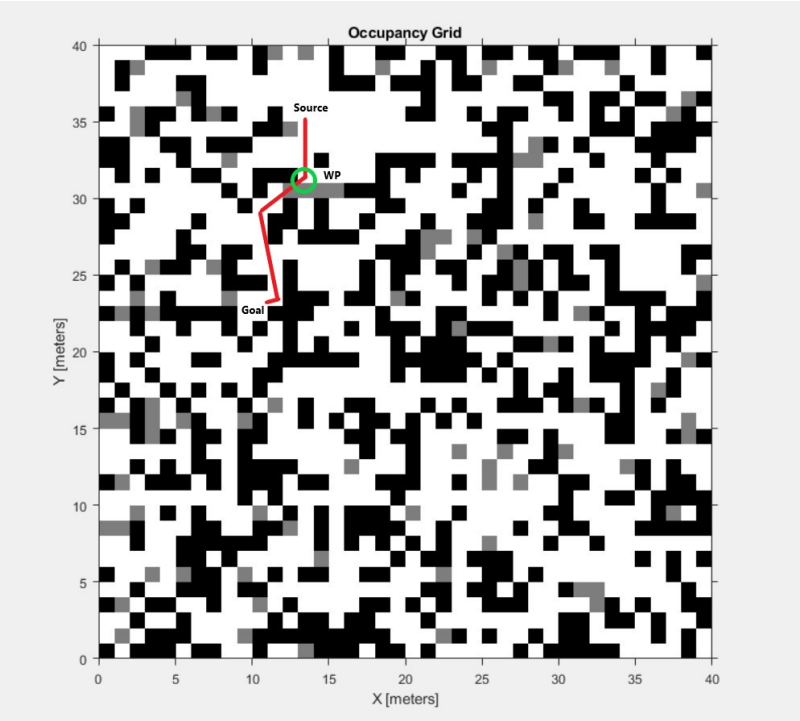


Figure 34.: Assigning the WP in Map Model #4

truth map based on the aerial view of the test scenario. In the ground truth map, the shape of tree bodies, the houses and the fences are represented as circles, polygons and lines respectively which is shown in Figure 36(c). As we compared the differential and particle drive USV in user-defined occupancy grid maps, the particle drive USV is showed lower computation time than the differential drive USV. Therefore, we used the particle drive USV in flooded area images testing to test system algorithms in a more efficient manner.

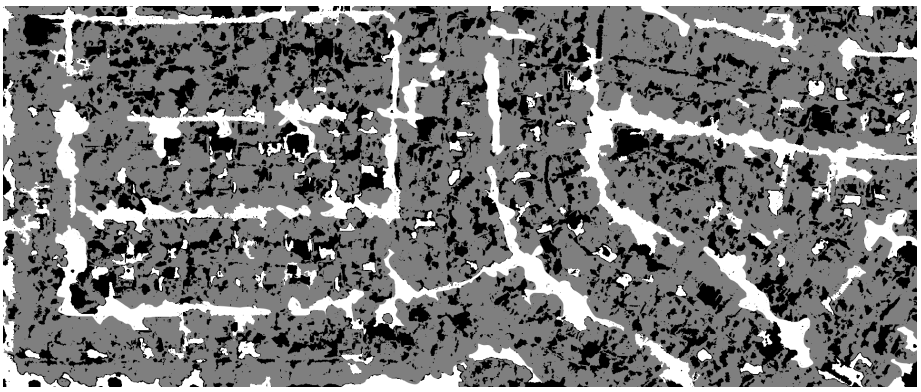
In the first simulation test, we defined a source and a single target point which is shown in Figure 37. According to the decision-feedback and path planning algorithms, an SP is not found, in contrast, MEP is found as 233 meters and accordingly PE_{SP} is calculated as 1. Therefore, USV needs to explore uncertain areas to find a possible SP for the rescue boat to reach the target in a more efficient manner.

Figure 35.: USV Simulation in Map Model #4

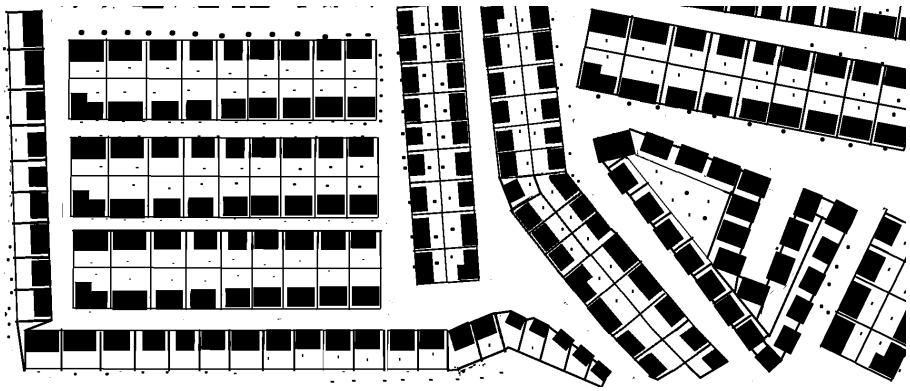
The simulation results are showing that USV successfully explored the uncertain areas while visiting the WPs and the SP is found as 1184 meters which are shown in Table XI. Figure 41 displays updated ground map where the uncertain areas are cleared by USV's laser range sensor. The changes in SP and MEP for a single target during the simulation process is shown in Figure 39.

In the second simulation test, we defined a source and multiple target points which are shown in Figure 40. According to the decision-feedback and path planning algorithms, the SPs are not found for each target, in contrast, MEP is found for each target as 617 meters, 1436 meters and 1948 meters respectively and then accordingly PE_{SP} for each target is calculated as 1. Therefore, USV needs to explore the uncertain areas while visiting the WPs to find a possible SPs for the rescue boat to reach the targets more efficiently. The simulation results are showing that USV successfully searched uncertain areas and the SPs are found as 560 meters, 1298 meters and

(a)



(b)



(c)

Figure 36.: (a) A flood area aerial image (b) Ground map model (c) Ground truth map model

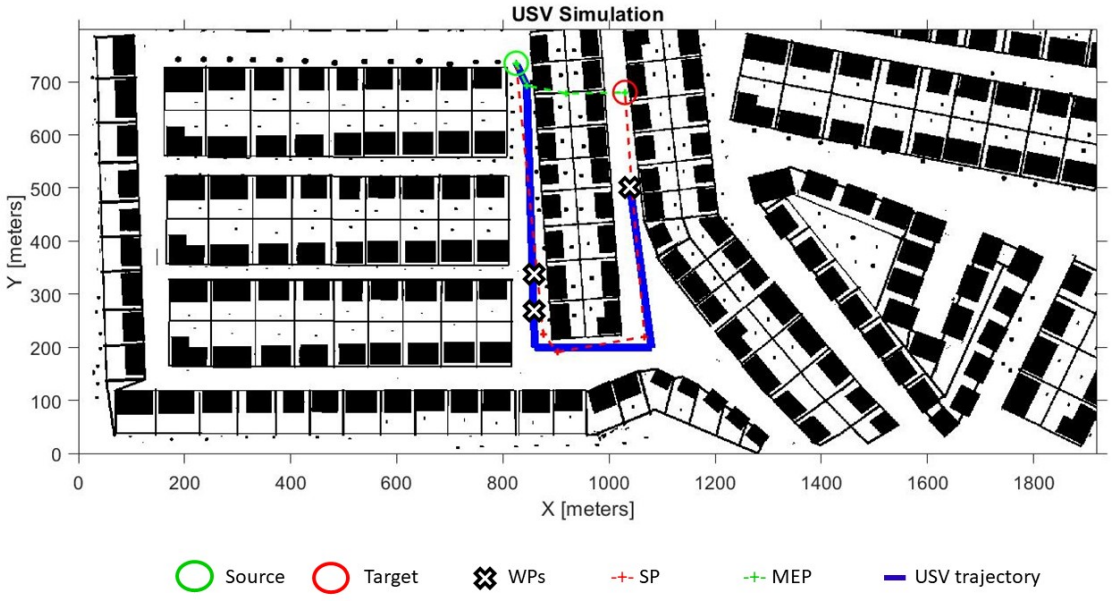


Figure 37.: USV simulation with one target.

1764 meters respectively which are shown in Table XII. Figure 38 displays updated ground map where the uncertain areas are cleared by USV's laser range sensor. The changes in SP and MEP for multiple targets during the simulation process is shown in Figure 42.

In the third simulation test, we tested another flooded area image which is shown in Figure 43. In this test case, we defined a source and a single target point which is shown in Figure 44. According to the decision-feedback and path planning algorithms, the SP is not found, in contrast, MEP is found as 1445 meters and accordingly PE_{SP} is calculated as 1. Therefore, USV needs to explore uncertain areas to find a possible SP for the rescue boat to reach the target in a more efficient manner. The simulation results are showing that USV successfully explored the uncertain areas while visiting the WPs and the SP is found as 1315 meters which are shown in Table XIII. Figure 45 displays updated ground map where the uncertain

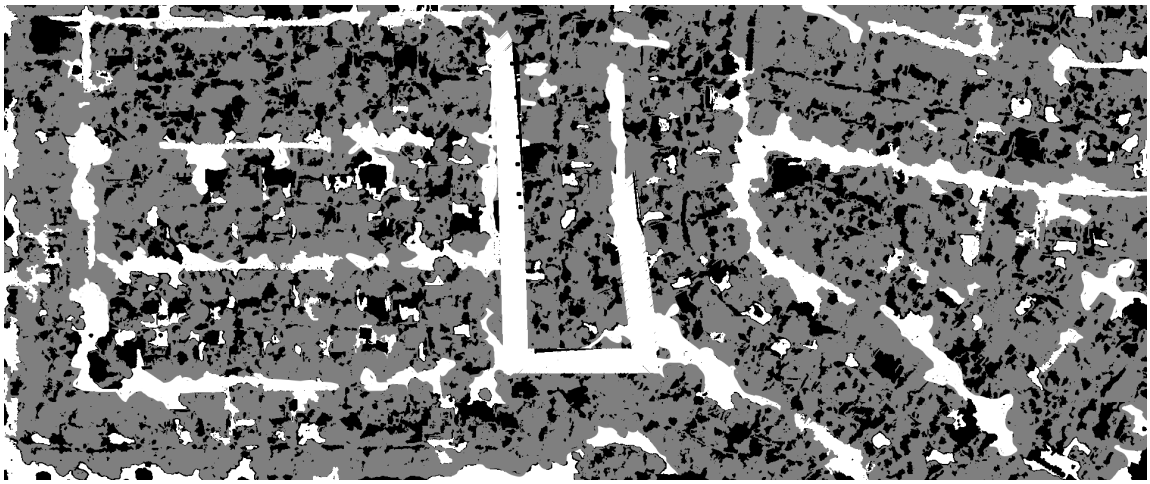
Process	L_{SP}(m)	L_{MEP}(m)	PE_{SP}
Initial	0	233	1
WP 1	0	674	1
WP 2	0	654	1
WP 3	1184	1184	0

Table XI.: SP and MEP findings in the single target test

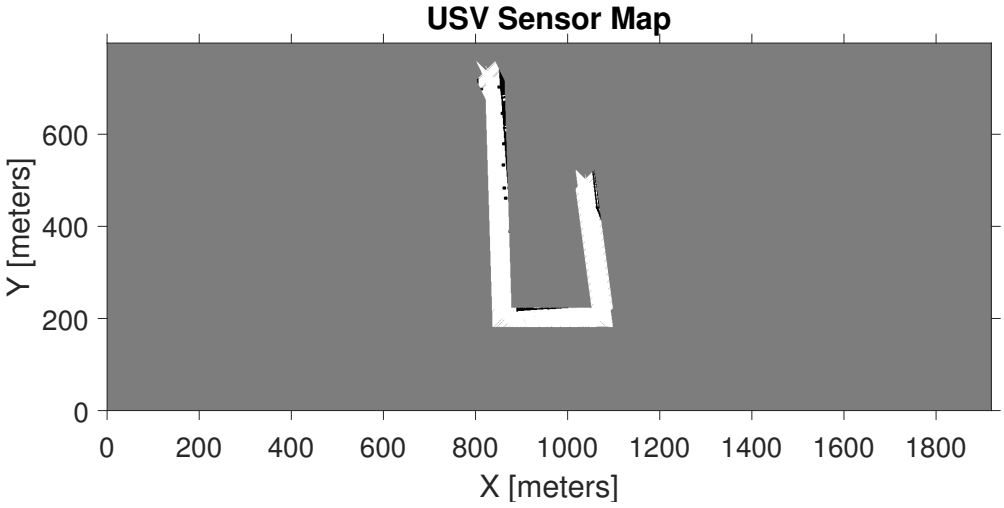
areas are cleared by USV's laser range sensor. The changes in SP and MEP for a single target during the simulation process is shown in Figure 46.

Target	Process	L_{SP}(m)	L_{MEP}(m)	PE_{SP}
I	Initial	0	617	1
	WP 1	0	565	1
	WP 2	560	560	0
Target	Process	L_{SP}(m)	L_{MEP}(m)	PE_{SP}
2	Initial	0	1436	1
	WP 1	0	1415	1
	WP 2	1298	1298	0
Target	Process	L_{SP}(m)	L_{MEP}(m)	PE_{SP}
3	Initial	0	1948	1
	WP 1	0	1847	1
	WP 2	0	1812	1
	WP 3	1764	1764	0

Table XII.: SP and MEP findings in the multiple target test



(a)



(b)

Figure 38.: (a) Updated ground map for one target test and (b) USV range sensor (r = 20m) data representation in the occupancy grid map.

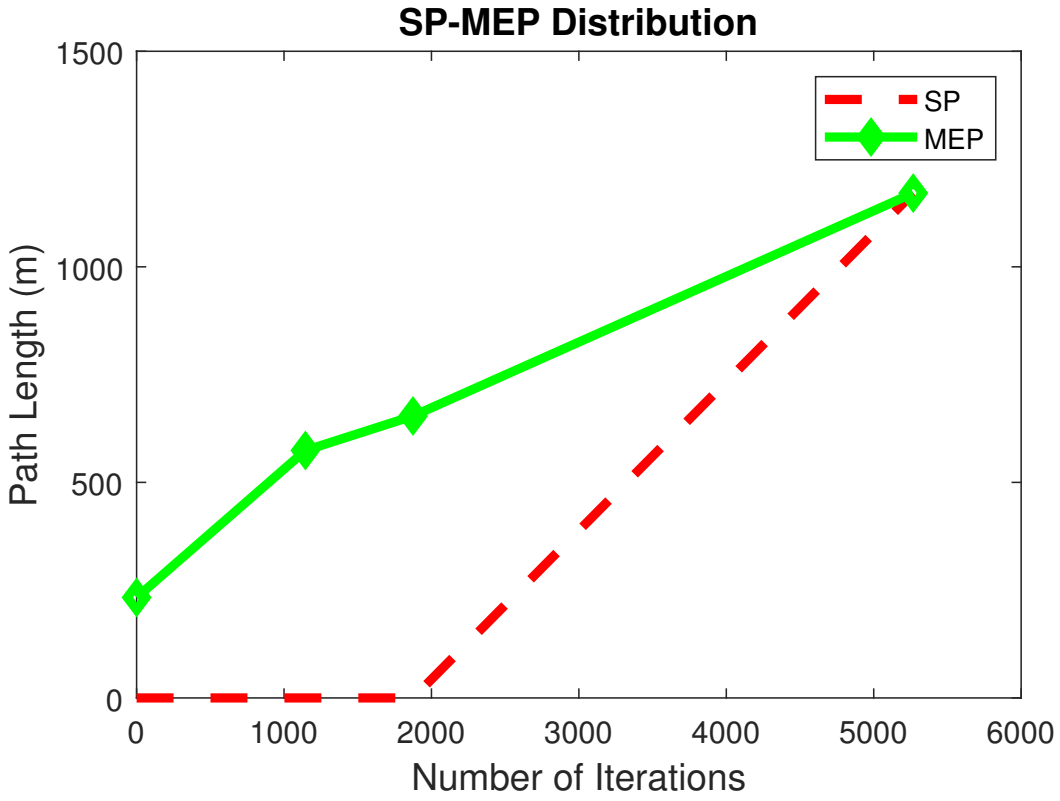


Figure 39.: Changes in the SP-MEP for a single target during the test simulation as new information is made available through clearing uncertainty.

Process	L_{SP}(m)	L_{MEP}(m)	PE_{SP}
Initial	0	1445	1
WP 1	0	1432	1
WP 2	0	1420	1
WP 3	0	1384	0
WP 4	0	1344	0
WP 5	1315	1315	0

Table XIII.: SP and MEP findings in the Flooded Area Test #2

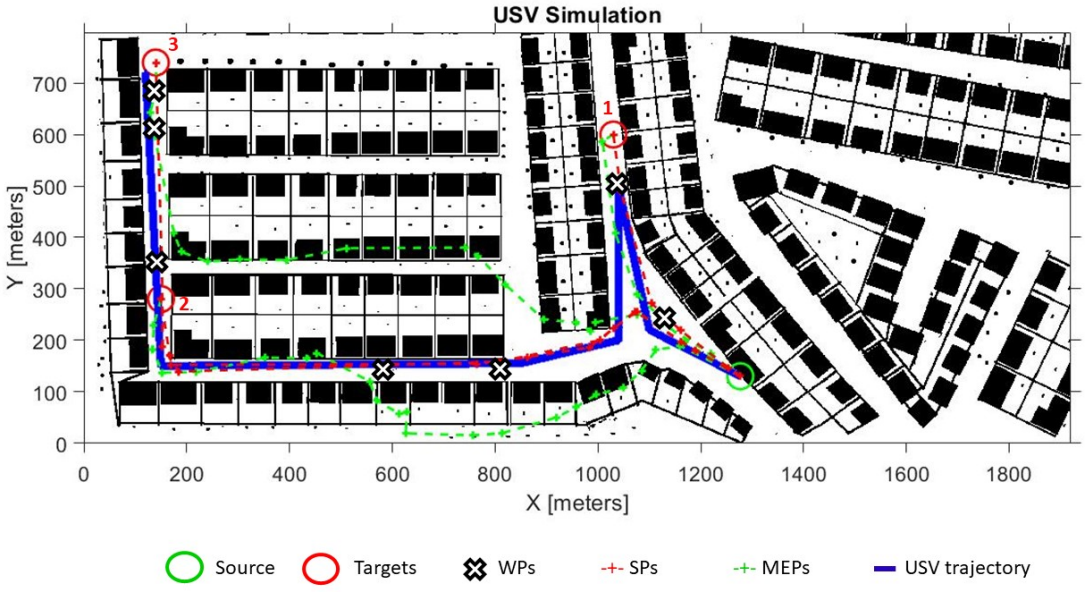
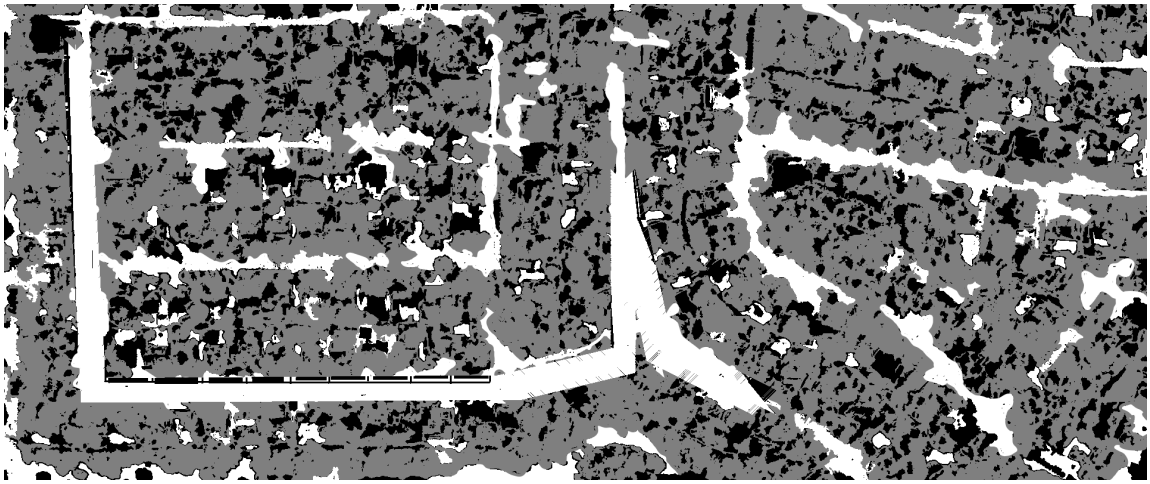
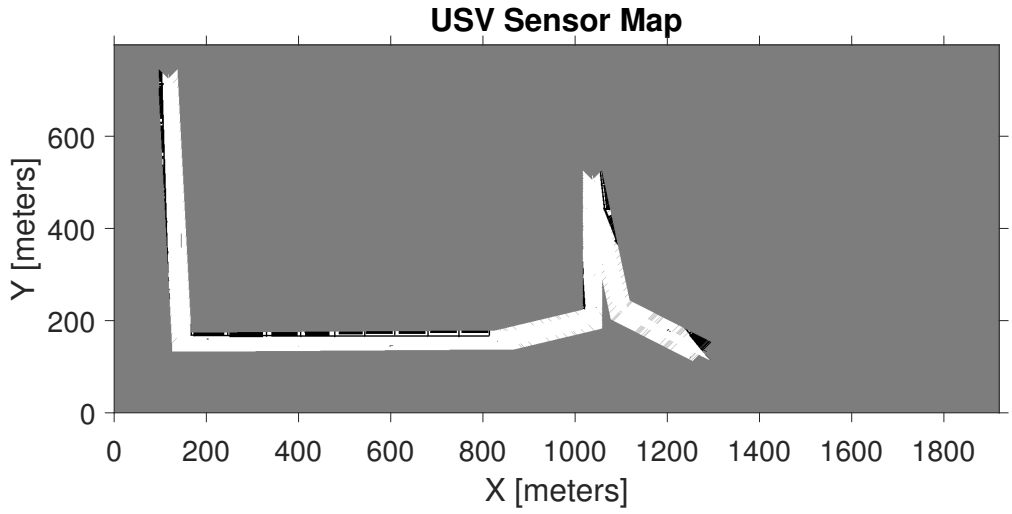


Figure 40.: USV simulation with multiple targets.



(a)



(b)

Figure 41.: (a) Updated ground map for multiple targets test and (b) USV range sensor (r = 20m) data representation in the occupancy grid map.

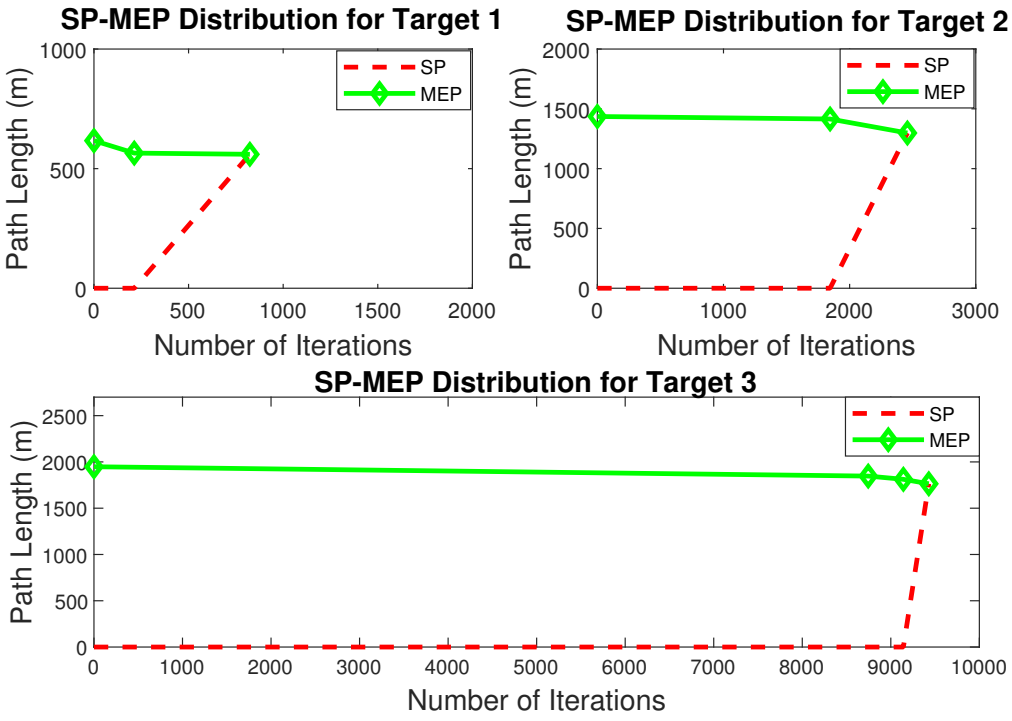
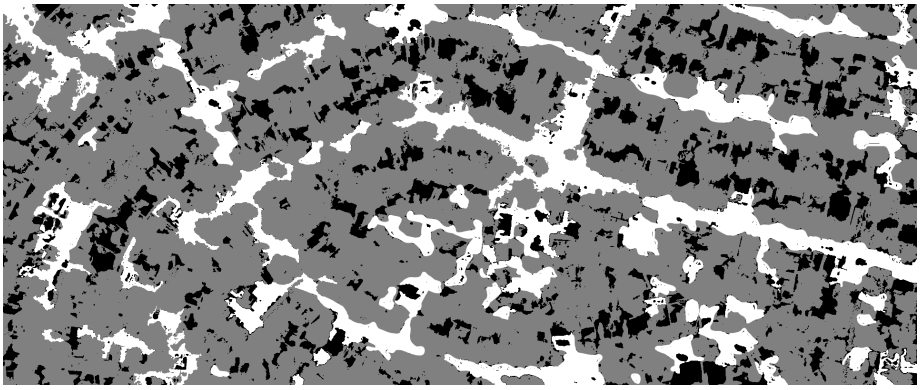


Figure 42.: Changes in the SP-MEP for multiple targets during the test simulation as new information is made available through clearing uncertainty.

(a)



(b)

(c)

Figure 43.: (a) A flooded area aerial image (b) Ground map model (c) Ground truth map model

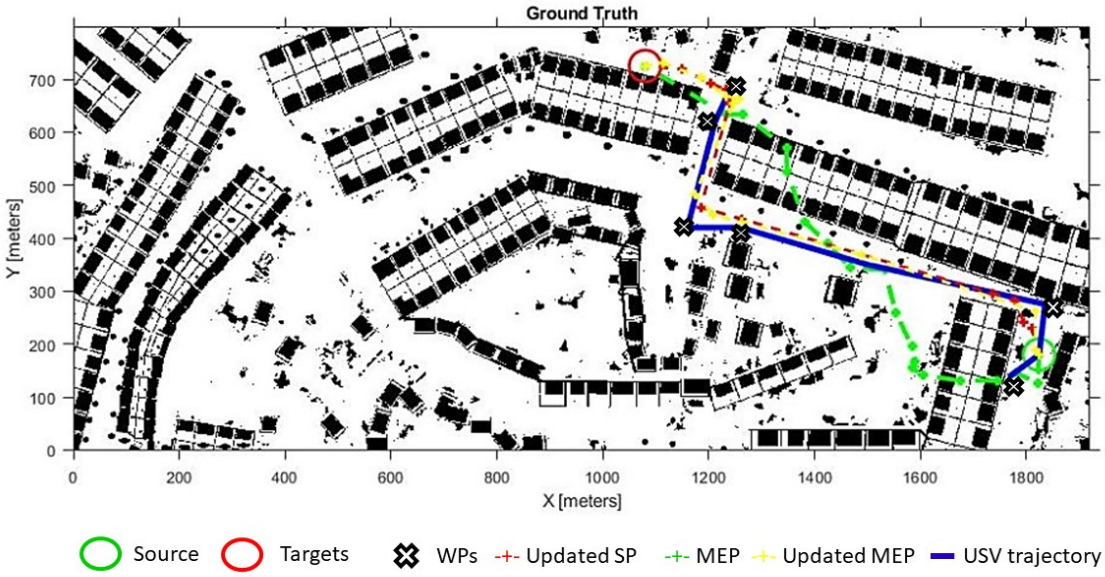
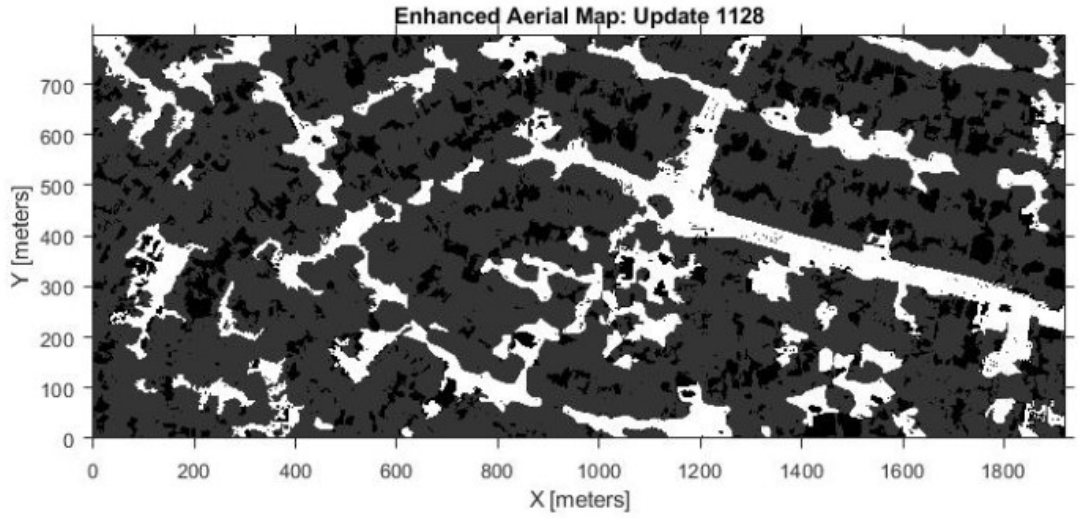
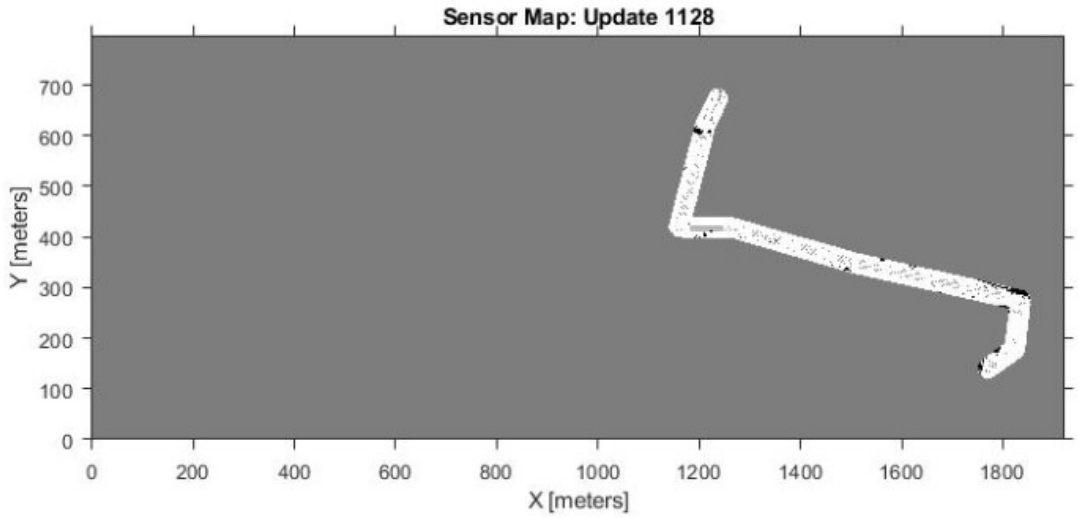


Figure 44.: USV simulation in Flooded Area Image Test #2.



(a)



(b)

Figure 45.: (a) Updated ground map (b) USV range sensor (r = 20 m) data representation in the occupancy grid map.

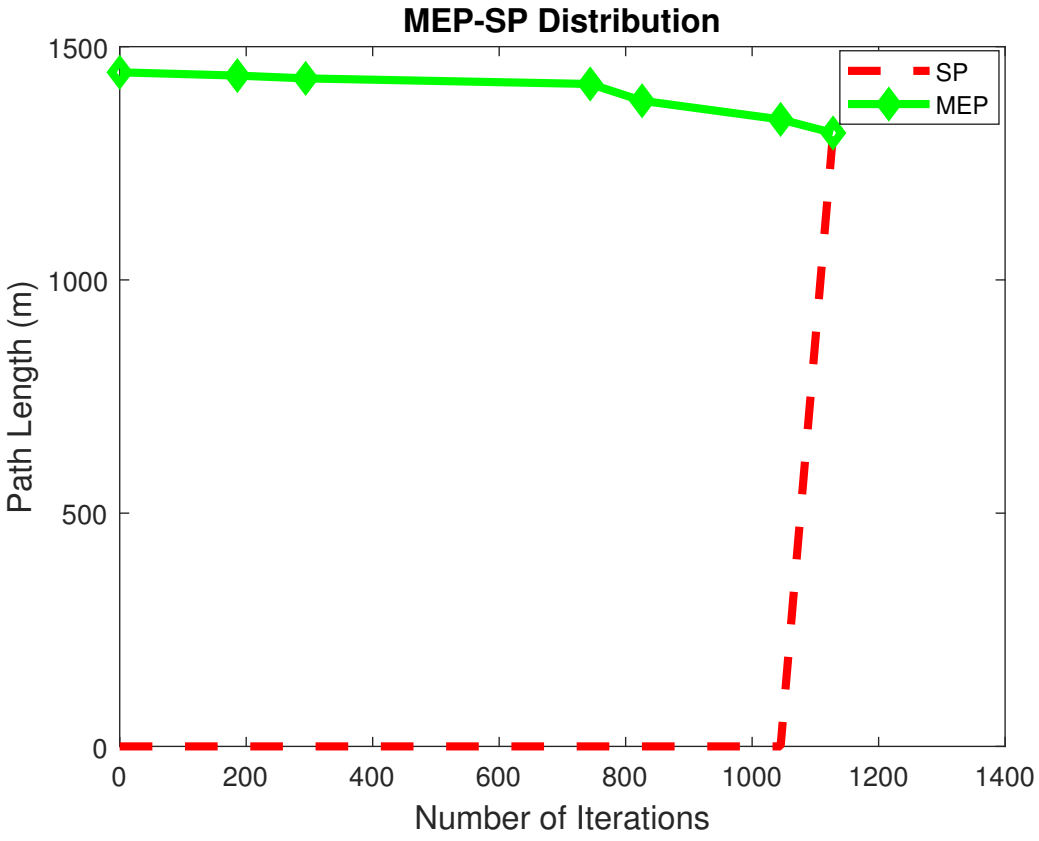


Figure 46.: Changes in the SP-MEP during the test simulation as new information is made available through clearing uncertainty.

CHAPTER 4

CONCLUSION

In this research, we proposed a flood rescue application where the main goal was exploring uncertain areas by USV to build a real-time ground map and plan near-optimal paths for a rescue boat in a flooded urban environment. Map building, path planning and decision-feedback algorithms are expected to greatly enhance the function of the rescue boat to handle flood rescue operations. The proposed research does not address all issues but should be a step toward enhanced flood rescue operations. Rather than focus on deeply hardware building, the research was the focus on deeply testing system algorithms. Simulation tests are used to demonstrate the effectiveness and applicability of the proposed approach for improving flood rescue operations.

In the purpose of using a rescue boat in the flooded urban environments, the rescue boat needs to reach victims in a more efficient manner. According to the path planning algorithm results, PRM path planning algorithm gives a shorter path and better computation time than A* and GA algorithms. Therefore, the PRM path planning algorithm satisfies the time requirement of the rescue operations.

Based on the work above, our next step will focus on the real world application to test system algorithms by using UAV/USV cooperative system.

REFERENCES

[1] E. Pinto, P. Santana, F. Marques, R. Mendonça, A. Lourenço, and J. Barata, "On the design of a robotic system composed of an unmanned surface vehicle and a piggybacked VTOL," in *Technological Innovation for Collective Awareness Systems*, pp. 193–200, Springer Berlin Heidelberg, 2014.

[2] R. Murphy, S. Stover, K. Pratt, and C. Griffin, "Cooperative damage inspection with unmanned surface vehicle and micro unmanned aerial vehicle at hurricane wilma," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, oct 2006.

[3] J. Dufek and R. Murphy, "Visual pose estimation of USV from UAV to assist drowning victims recovery," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, IEEE, oct 2016.

[4] A. Mancini, E. Frontoni, P. Zingaretti, and S. Longhi, "High-resolution mapping of river and estuary areas by using unmanned aerial and surface platforms," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, jun 2015.

[5] R. Mendonca, M. M. Marques, F. Marques, A. Lourenco, E. Pinto, P. Santana, F. Coito, V. Lobo, and J. Barata, "A cooperative multi-robot team for the surveillance of shipwreck survivors at sea," in *OCEANS 2016 MTS/IEEE Monterey*, IEEE, sep 2016.

[6] N. Miskovic, S. Bogdan, E. Nad, F. Mandic, M. Orsag, and T. Haus, "Unmanned marsupial sea-air system for object recovery," in *22nd Mediterranean Conference on Control and Automation*, IEEE, jun 2014.

[7] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, and Z. Ming, "A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems," *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 9585–9596, dec 2016.

[8] A. Lakas, B. Belkhouche, O. Benkraouda, A. Shuaib, and H. J. Alasmawi, "A framework for a cooperative UAV-UGV system for path discovery and planning," in *2018 International Conference on Innovations in Information Technology (IIT)*, IEEE, nov 2018.

[9] J. Zhang, J. Xiong, G. Zhang, F. Gu, and Y. He, "Flooding disaster oriented USV & UAV system development & demonstration," in *OCEANS 2016 - Shanghai*, IEEE, apr 2016.

[10] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "UAV assisted USV visual navigation for marine mass casualty incident response," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, sep 2017.

[11] D. Costea and M. Leordeanu, "Aerial image geolocalization from recognition and matching of roads and intersections,"

[12] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "Efficient road detection and tracking for unmanned aerial vehicle," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 297–309, feb 2015.

[13] C. Kanellakis and G. Nikolakopoulos, "Survey on computer vision for UAVs: Current developments and trends," *Journal of Intelligent & Robotic Systems*, vol. 87, pp. 141–168, jan 2017.

[14] K. U. Gunasekaran, E. Krell, A. Sheta, and S. A. King, "Map generation and

path planning for autonomous mobile robot in static environments using GA," in *2018 8th International Conference on Computer Science and Information Technology (CSIT)*, IEEE, jul 2018.

[15] L. Cheng, C. Liu, and B. Yan, "Improved hierarchical a-star algorithm for optimal parking path planning of the large parking lot," in *2014 IEEE International Conference on Information and Automation (ICIA)*, IEEE, jul 2014.

[16] M. Ganeshmurthy and G. Suresh, "Path planning algorithm for autonomous mobile robot in dynamic environment," in *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, IEEE, mar 2015.

[17] M. M. Kurdi, A. K. Dadykin, and I. Elzein, "Navigation of mobile robot with cooperation of quadcopter," in *2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)*, IEEE, feb 2017.

[18] M. S. H. Nizami, S. M. R. Al-Arif, A. I. Ferdous, M. M. S. Riyadh, and F. R. Faridi, "Efficient algorithm for automated rescue boats," in *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, IEEE, may 2012.

[19] "Differential drive robot," in *Path Following for a Differential Drive Robot - MATLAB & Simulink*.

[20] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Tech. Rep. CMU-RI-TR-92-01, Carnegie Mellon University, Pittsburgh, PA, January 1992.

[21] "Robotics.prm," in *https://www.mathworks.com/help/robotics/ug/pure-pursuit-controller.html,Pure Pursuit Controller - MATLAB & Simulink*.

[22] A. Upadhyay, K. R. Shrimali, and A. Shukla, "UAV-robot relationship for coordination of robots on a collision free path," *Procedia Computer Science*, vol. 133, pp. 424–431, 2018.

[23] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59–69, 2014.

[24] R. Kala, "Robot path planning," 2016.

[25] "Hurricane harvey flooding," in *Hurricane Harvey Flooding - Woolpert*.