Autonomous Navigation and Mapping of Water Channels in a Simulated Environment Using Micro-Aerial Vehicles

Syed Izzat Ullah*
Department of Computing Sciences
Texas A&M University-Corpus Christi
Texas, USA
sizzatullah@islander.tamucc.edu

Abubakr Muhammad
Department of Electrical Engineering
Lahore University of Management Sciences (LUMS)
Lahore, Pakistan
abubakr@lums.edu.pk

***Abstract*—Irrigation canal networks serve as the bedrock of agriculture sectors across the globe as they are the primary channel through which water runs from major sources to agricultural lands. However, the water-carrying capacity of these water channels significantly reduces over time because of erosion, structural deterioration, and silt accumulation. As a result, routine inspections are required to analyze and repair these water channels which necessitates automation because of the vast length of the channels. We present a framework that enables Micro-Aerial Vehicles(MAVs) not only to navigate in an unknown cluttered canal environment but also to provide a complete 3-Dimensional map for the inspection. The framework consists of three main components (mapping, path planning, and mission planner) that gradually explore the environment while solving for start to local goal queries. We use Octomap; an octree-based representation of the environment for mapping, and we extended the Informed Rapidly-exploring Random Tree (Informed-RRT*) for optimal path planning and replan paths with respect to the static nearby and dynamic obstacles perceived during the execution of the mission. A simulated 2,378 meters length of canal environment is implemented and demonstrated by using the Airsim simulation in the Unreal engine, running on Robot Operation System (ROS) and Linux OS. Results obtained show that the framework enables the MAV to navigate over a simulated canal environment and allows the MAV to map the 3D structure of the canal.**

Index Terms*—UAV, navigation, Mapping, Path Planning, ROS, Unreal engine, Airsim, RRT

I. Introduction

Water has become an inevitable global sustainability risk. Agriculture, as the greatest water consumer, faces challenges associated with water quality and availability. Many river basins, including the Indus, Nile, Ganges, Yangtze, and Mississippi, rely significantly on huge canal networks. The Indus River Basin, one of the mightiest and longest rivers, comprises over a hundred thousand big and small channels, with the main channel stretching for 57,000 km [1] and small channels spanning 1.6 million Km [2]. Due to natural environmental changes and human factors, such as silt accumulation, flooding, pollution, and water theft, the extensive infrastructure is constantly deteriorating, resulting in the efficiency reduction of the irrigation system. Hence, for operations and maintenance, a frequent inspection of the structure is required. Currently, the inspection and repair of these water channels are done manually by human labor which results in considerable uncertainty and inefficiencies [3]. Hence, an automated system with sensing and navigation capabilities is required that can traverse the length of the canal at large distances with minimal human interventions.

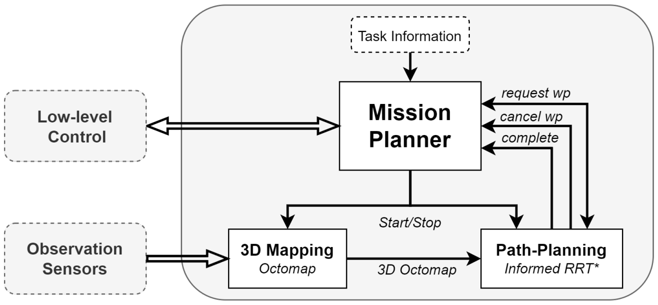


Fig. 1. Main modules of the proposed framework

Several researchers have attempted to solve this problem using robotics platforms. A small-scale aerial vehicle with perception and navigation capabilities that can navigate beneath tree canopies, avoid overhanging tree branches and bridges, and profile canal channels quickly and efficiently could be an option. Gadre et al. [4] report the use of an unmanned surface vehicle to map a natural and riverine environment, however, their system is unable to map above the canal banks and below the canopy. Bio-inspired snake robots with autonomous navigation [5] and waterproof design [6] can be configured to map the level of siltations and canal structural deterioration. Rathinam et al. [7] has used fixed-wing UAVs at higher altitudes to map rivers or coastal lines which will not work beneath canal canopy. This platform may be more useful in typical riverine environments, but our goal is to develop a

*Syed Izzat Ullah was with the Department of Electrical Engineering, Lahore University of Management Sciences, Lahore, Pakistan. He is now pursuing a Ph.D. in the Department of Computing Sciences at Texas A&M University-Corpus Christi, TX, USA system that can operate in difficult conditions such as flowing water, overhanging trees, or dense canal canopies. Yang et al. [8] has used a small flying vehicle (MAV) to describe mapping riverine environments in short distances, using a passive monocular vision sensor for mapping and an ultrasonic sensor for altitude estimation. However, the goal of our work is the same, but we deliberately plan out significantly longer missions where it is critical to profile not only the magnitude of the canal but also the vegetation along its bank, as well as avoid obstacles that could occur in the path of the vehicle. Similar to our work, Scherer et al. [9] has demonstrated autonomous exploration and mapping of riverine environments through aerial vehicles, using a front-facing camera for river detection, and a 2D rotating LiDAR coupled with a stereo camera for 3D reconstruction of the river. The system has been validated on a 2 km-long river. However, because of the weight and computation limitations, using separate sensors for mapping and exploration may not be a good choice for an aerial vehicle. Abbas et al. [10] has proposed and evaluated an aerial autonomous canal traversal system based on deep learning techniques.

This paper addresses the challenge of autonomous navigation and mapping for a Micro-aerial vehicle (MAV) that operates in a significantly large, unknown, and cluttered canal-like environment. We achieved it in a simulation framework, with a front-facing stereo camera mounted on the front of the vehicle, as the only perception sensor for both mapping and obstacle avoidance. We used octomap [11]; an octree-based representation for mapping, and an Informed-RRT* [12] as path planner. We also implemented a mission planner, that is responsible for the overall flow of the MAV autonomous navigation. In this work, we assume that the localization of the vehicle and the local goal points at every 25 meters over the whole canal is known *apriori*.

II. Background

A. Mission Planner

The mission planner is a high-level controller, responsible for the overall flow of the proposed MAV autonomous navigation framework over a canal. When the mission is started, the path planning module and mission planner establishes a bidirectional communication. Since the complete canal environment is divided into discrete local goal points, therefore, the mission planner, when required, requests path planning module for obstacle-free waypoints from the MAV current position to the subsequent local goal position, and adapts them for the low-level control of the MAV. Additionally, the path planning module can inform the mission planner to cancel the ongoing waypoints when the same path is re-planned with updated waypoints.

B. Mapping

The mapping module provides the free and occupied space with reference to an inertial frame of reference by incrementally building a representation of the environment using information from perception sensors, such as the stereo camera, which provides range information about nearby obstacles. To process this information, we use an Octomap [11], a hierarchical 3D octree-based representation of an environment or space $V \subseteq \mathbb{R}^3$, as shown in Figure 2. Each cube in the octomap is called a voxel v, and can be free, occupied, or unknown. Additionally, free voxels combine to form free space $V_{free} \subset V$, occupied voxels combine to form occupied space$V_{occ} \subset V$, and unknown voxels combine to form unknown space $V_{un} \subset V$. Cumulatively, the three subspaces combine to form the entire space $V = V_{free} \cup V_{occ} \cup V_{un}$.

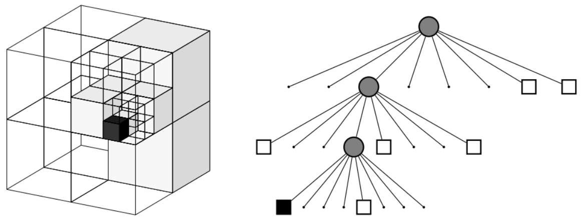


Fig. 2. Illustration of an octree storing free and occupied cells as shaded white and black, respectively. The left picture is a volumetric model the right one is its corresponding tree representation [11].

Octomap has three main characteristics. First, the probabilistic representation of the states, which updates map information and also protect it from noisy measurements, i-e., based on prior information of the position state, probabilistically determines the new value. Second, the capability of representing and incorporating unexplored areas, which may be useful in the exploration of unknown environments. Third, due to its volumetric nature, Octomap has the ability to efficiently extend and enlarge the map as demanded. Finally, in sampling-based methods such as the one we used in this work i-e., Informed-RRT*, Octomap results in an effective collision-checking method.

C. Path Planning

The path planning module provides an optimal and obstacle-free path from the MAV's current/start position to a goal position while incorporating the vehicle's kinematics and dynamics constraints. When defining a mission, it is one of the most important components of an autonomous navigation system. Path planning aims to plan a real-time global path to the goal, avoid collisions, and optimize a cost function while taking kinodynamic constraints into account [13]. Since the MAV is a three-dimensional vehicle that operates in $\mathbb{R}^3$ space (i.e., $\mathcal{W} = \mathbb{R}^3$), therefore, the configuration space $\mathcal{C}$ (or state space) is expressed as $\mathcal{Q} \in \mathcal{C} = \mathrm{SE}(3) = \mathbb{R}^3$ x $\mathrm{SO}(3)$, considering both position and orientation of the MAV. Generally, the vehicle is surrounded by obstacles that have to be avoided. Therefore, the state space is further subdivided into free space $\mathcal{Q}_{\text{free}} \subseteq \mathcal{Q}$ and obstacles space $\mathcal{Q}_{\text{obs}} \subseteq \mathcal{Q}$ such that $\mathcal{Q} = \mathcal{Q}_{\text{free}} \cup \mathcal{Q}_{\text{obs}}$. Hence, an optimal path is defined as a continuous path $p : [0, 1] \rightarrow \mathcal{Q}_{\text{free}}$ that minimizes a given cost function while connecting $p(0) = \mathcal{Q}_{\text{start}}$ to $p(1) = \mathcal{Q}_{\text{goal}}$ through the free space [12].

Our work is inspired by Informed-RRT* [12], which is a variant of the RRT*. As shown in Figure 3, Informed-RRT* works as RRT* until the first solution is obtained, then it only tests from the sub-state set that can improve the current solution. Because of its focused search, informed-RRT* is less dependent on the dimension and domain of the planning problem and has a faster ability to find improved topologically unique paths.

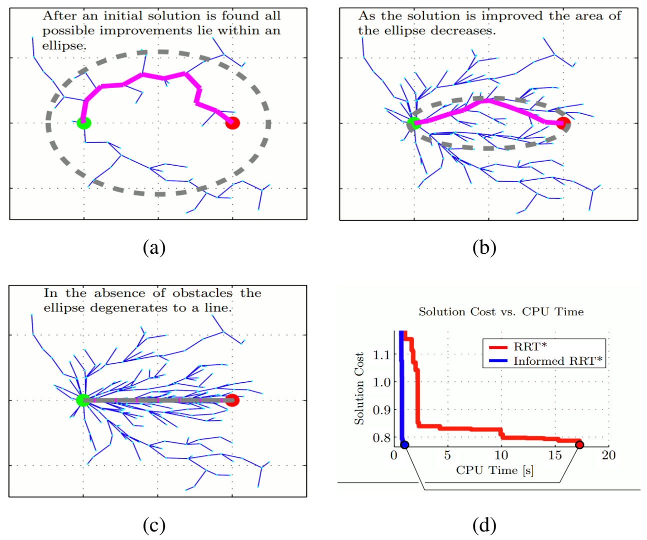


Fig. 3. Example of the Informed-RRT* [12]. Informed RRT* works the same as RRT* until the first solution is determined, then as shown in 3(a) it focuses its search to only a set of states within the ellipsoid that contains the first solution. 3(c) shows an optimal path provided by Informed-RRT*. 3(d) shows the comparison of RRT* and Informed-RRT* in terms of their solution cost vs computation time

III. Methodology

A. Simulation environment

In this work, the simulation framework consists of Microsoft Airsim [14] in the Unreal Engine, coupled with the Robot Operating System (ROS). Unreal Engine has the ability to create 3D realistic environments with features such as water, vegetation, rain, fog, and modeling wind. Microsoft Airsim is a plugin developed for Unreal Engine and Unity, mainly for aerial vehicles and cars. The aerial vehicle in the Airsim is an AR drone that is equipped with common robotic sensors such as LiDAR, stereo and monocular cameras, GPS, and Inertial Measurement Unit (IMU). Apart from this, the drone model also provides properties and parameters required for computing rigid body dynamics such as mass, inertia, coefficients for friction and restitution, coefficients for linear and angular drag, etc. Lastly, ROS plays an important role as a middleware communication module that allows components to get connected and facilitates modular design.

Inspired from a real-world canal environment, where an aerial vehicle can expect bridges, tilted trees, brushes, branches, and trunks as obstacles, in this work, we have modeled and included them in various locations in the environment as shown in fig. 4. Using Autodesk, the 3D canal mesh is created with the dimension of 3.5m height and 6m to 9m width, and then extended to a closed-loop shape using the spline tool in the Unreal engine, as shown in Figure 5. The environment also includes sharp turns (approx. 57^o) to demonstrate either the separation of canals from the main channel or an actual turn.

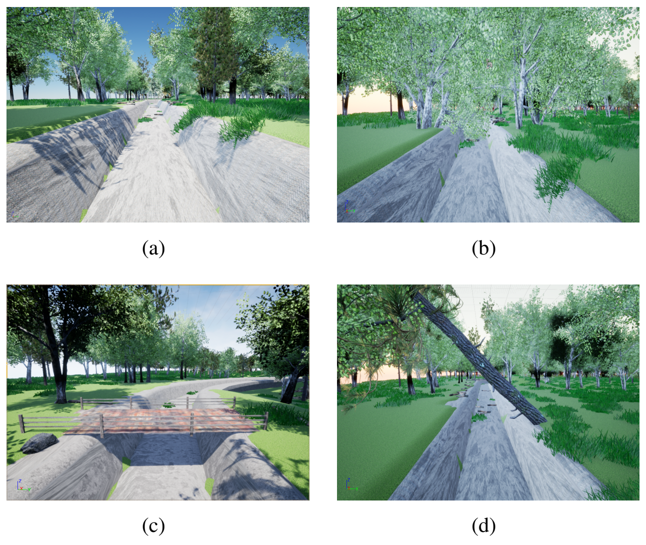


Fig. 4. Illustration of the features incorporated in the simulation environment. (a). The magnitude of the vegetation on the canal bank, can block waterways and may hinder the navigation of the MAV. (b). Hanging trees that can make navigation challenging for the MAV. (c). around five bridges have been built at different locations of the canal. (d). Fallen trees

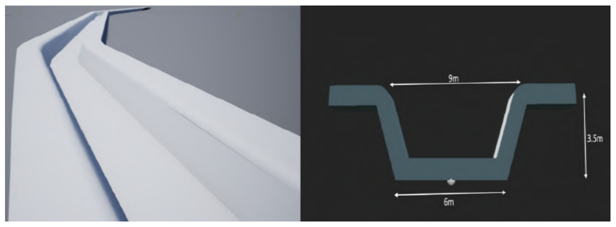


Fig. 5. The dimension and shape of the canal structure before adding any feature

As shown in figure 6, the complete canal structure consists of 2,378 meters, which contains brushes, bridges, tree branches, and eight various types of trees as the canopy of the canal. These various features will serve as soft and hard obstacles for the autonomous navigation of the vehicle.

B. System Architecture

Figure 7 shows the system architecture of the proposed framework. In this section, we have mainly discussed the details of the two main components of our system; mapping and path planning. As the observation sensors, we have used a stereo camera and GPS/INS sensors for the perception of the environment. The stereo camera provides raw images of the left and right cameras. In the mapping module, the raw images are calibrated and rectified to find matching points between them and avoid the correspondence problem. Using

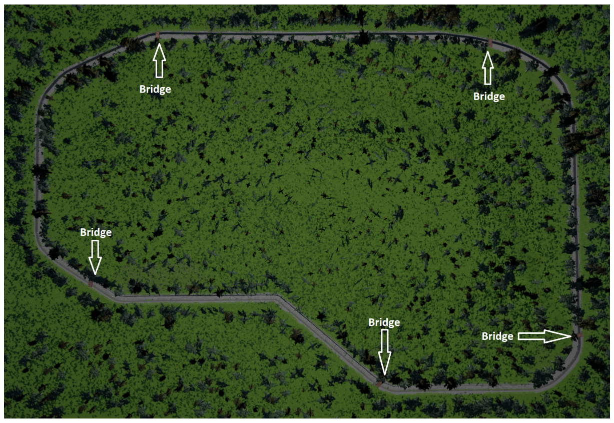


Fig. 6. Complete simulated environment in a closed-loop shape with a total length of 2,378 meters. various types of trees in different orientations are spread across the canal bank

these processed images, we find the disparity map and the point cloud in the MAV frame of reference. In the meanwhile, we estimate the state of the vehicle using GPS/INS sensors to transform the 3D point cloud data from the vehicle frame of reference to the world frame of reference in order to have a global perspective of the environment and obstacles. After the map is generated, the third module has to plan an obstacle-free path for autonomous navigation. In the following subsections, we will explain the implementation of these different modules.

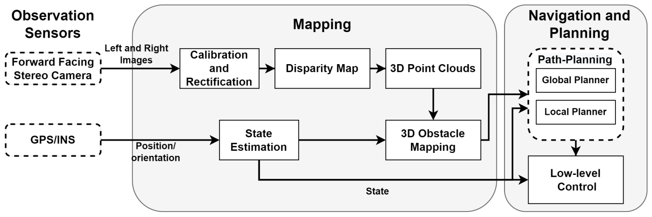


Fig. 7. System Architecture of the proposed framework

1) Disparity Image and Point Cloud Generation: The first step for disparity image and point cloud is to acquire the left and right images from the stereo camera. A disparity image in the context of the stereo camera is defined as the distance between the matching pixels in the left and right images. Several methods have been proposed in the literature to achieve the disparity image. In this work, we have used the block matching method, which is provided in the *Stereo-image-proc* node [15] of the ROS Image Pipeline. As shown in Figure 8, the *Stereo-image-proc* node subscribes to the left and right camera information and raw images, co-calibrates and processes the raw images, and produces the disparity image and point cloud.

Point cloud generated from *Stereo-image-proc* is in the stereo frame-of-reference. In order to have a global perspective of the environment, especially the obstacles, we need to transform the point cloud to the world frame of reference. As shown in Figure. 9, initially the point cloud data is locally transformed from stereo frame to the base frame (i-e vehicle frame, usually

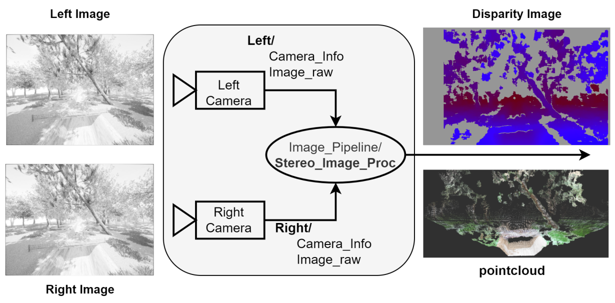


Fig. 8. Illustration and results of the *Stereo image proc*. It subscribes to the left and right image and then computes the disparity image and point cloud

at the center of the vehicle), which is always fixed as the vehicle is a rigid body and the stereo camera is tightly attached to it. The point cloud from the base frame is then transformed into a world frame. Since, the vehicle position changes at all times, therefore, knowledge of localization is required in order to transform. Hence, using the GPS/INS or the simulation positional information, transform the point cloud from the base frame to the world frame of reference.

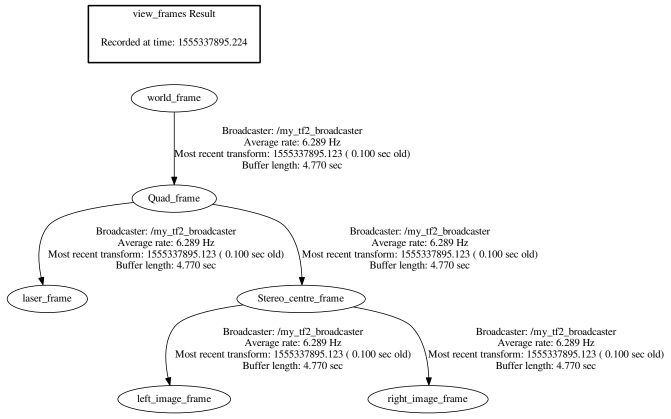

view_frames Result
Recorded at time: 1555337895.224
world_frame
Broadcaster: /my_tf2_broadcaster
Average rate: 6.289 Hz
Most recent transform: 1555337895.123 (0.100 sec old)
Buffer length: 4.770 sec
Quad_frame
Broadcaster: /my_tf2_broadcaster
Average rate: 6.289 Hz
Most recent transform: 1555337895.123 (0.100 sec old)
Buffer length: 4.770 sec
Broadcaster: /my_tf2_broadcaster
Average rate: 6.289 Hz
Most recent transform: 1555337895.123 (0.100 sec old)
Buffer length: 4.770 sec
laser_frame
Stereo_centre_frame
Broadcaster: /my_tf2_broadcaster
Average rate: 6.289 Hz
Most recent transform: 1555337895.123 (0.100 sec old)
Buffer length: 4.770 sec
Broadcaster: /my_tf2_broadcaster
Average rate: 6.289 Hz
Most recent transform: 1555337895.123 (0.100 sec old)
Buffer length: 4.770 sec
left_image_frame
right_image_frame

Fig. 9. Transformation tree from stereo camera to world coordinate system

2) 3D Mapping: As discussed in section II-B, we have used Octomap [11] in the ROS framework to map the canal environment. Octomap is based on octree representation of the environment. To ensure adaptability and cope with noisy sensor measurements, it uses probabilistic occupancy estimations. Our method does not require any prior map features to be known, instead, the map is generated dynamically over time as the vehicle traverses the environment. As shown in the rqt-graph of the ROS *Octomap server* in Figure 10, the octomap package subscribes to the $pointcloud$ and tf topics and publishes the $occupied\ cells$ which contains the obstacle map. As previously discussed, the $pointcloud$ is generated from the left and right images of the stereo camera. tf is a transformation channel, that publishes the transformation between the stereo frame and to the world frame, as shown in

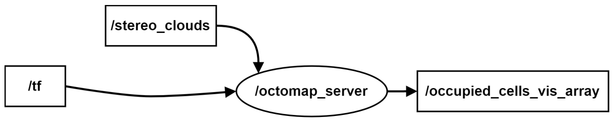


Fig. 10. rqt-graph of the Octomap server. It subscribes to the point cloud of the stereo camera and the real-time transformation tf, and publishes the occupied cells

Figure 9.

C. Path Planning

The path planner requires knowledge of the obstacles in the environment that has to be avoided. From the mapping module explained in section III-B2, we achieve a 3D map representation of the environment that includes information about the occupied (obstacles) and unoccupied (free space), hence, we can use it to plan obstacle-free paths to navigate the MAV over the canal. As mentioned in section II-C, our work is inspired by Informed-RRT* [12], which is an improved version of the famous RRT*.

We take the advantage of Flexible Collision Library(FCL) [16] for collision checking and proximity computation, and Open Motion Planning Library (OMPL) [17] for the implementation of Informed-RRT*.

IV. Experimental Results

To validate our approach, we have tested the proposed framework over the simulated canal environment in different scenarios. In the later subsections, we have explained the results of mapping and path planning one by one.

A. Mapping

Figure 11 shows the result of mapping the canal environment where only the occupied area is shown for clarity and the height is visualized with color coding. Figures 11(a) is the actual ground truth of the complete canal which has 2,378 meters of length. On the other hand, Figure 11(b) is its map generated using the stereo camera of the AR drone in the Unreal engine. The complete canal environment took 39.63 minutes for the vehicle to map and 184.9MB of memory is consumed. Furthermore, the resolution of the octree during mapping was 0.2 meters.

Figure 11(c) illustrates the simulated canal mainly with three important features that serve as an obstacle to the navigation of the MAV; (a) a tilted tree, (b) a bridge, and (c) vegetation on the canal bank. From Figure 11(d) we can clearly see that our approach has accurately sensed and mapped all three features. Furthermore, from the results, we believe that the proposed approach of using a Micro-aerial vehicle is able of mapping vegetation on the canal bank, silt accumulation, and major structural deterioration.

B. Path Planning

The role of the path planning algorithm is to provide an optimal and obstacle-free path, from the start position of the vehicle to the goal position, when requested by the mission

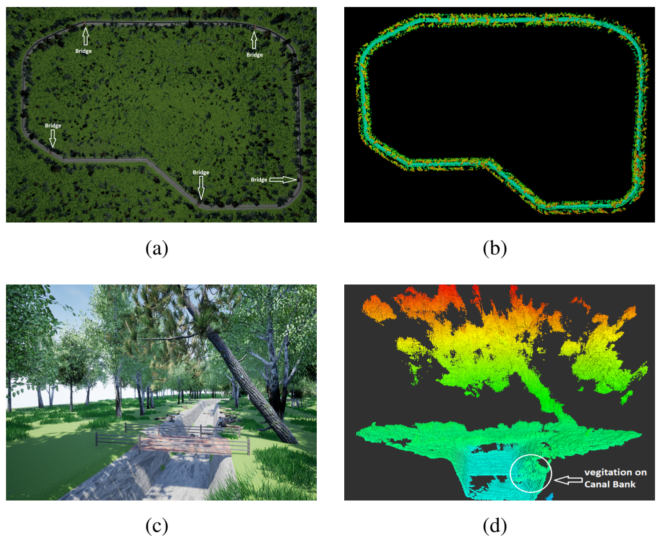


Fig. 11. (a) Actual ground truth of the simulated canal environment. (b) map of the complete canal environment using our framework. (c) A close look at the scene in the simulated canal where three important features are built at the same location; a bridge, a fallen tree trunk, and vegetation. (d) The efficient mapping of all the three features mentioned in d.

planner. Depending on the situation, the mission planner can request for replanning of the path. In order to traverse and map the complete canal, the mission planner will iteratively request feasible paths from the path planning algorithm unless the mission is completed. In our experiments, we performed experiments in different scenarios based on the clutteredness of the environment and evaluated whether our approach is successful or failing.

1) Scenario 1: No obstacles: In this experiment, there was no obstacle between the start and the goal point. The expectation with the path planning algorithm was to provide a straight path as no obstacle lies between the start and the goal point. Hence, from Figure 12 we can see that the proposed framework in the mapped canal environment has returned a straight and efficient path.

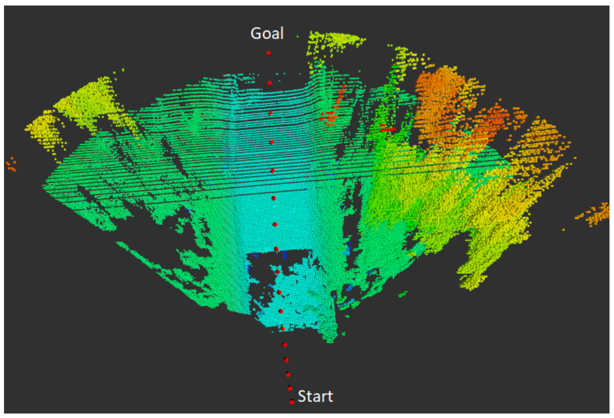
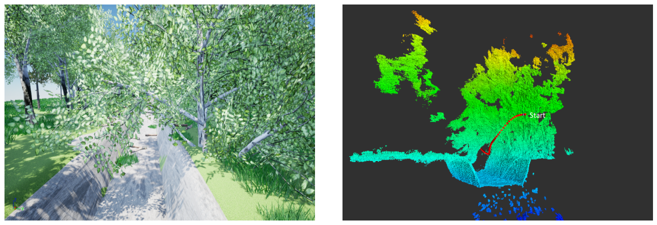


Fig. 12. Illustration and result of scenario 1 where the path planning algorithm has returned a straight path

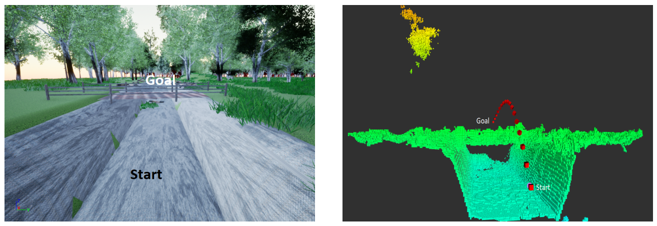
2) Scenario 2: Hanging branches of the Tree: This scenario represents a situation in which the vehicle faces hanging branches of the tree as shown in Figure 13. In this scenario, the vehicle can either navigate over the tree or find an obstacle-free path beneath tree branches, if one exists. As shown in Figure 13(b), the vehicle has successfully avoided the hanging branches and reached the goal point.



(a) (b)

Fig. 13. (a) Ground truth for scenario 1 where hanging tree branches makes the navigation of the MAV challenging. (b) the red line shows the path planned by our framework.

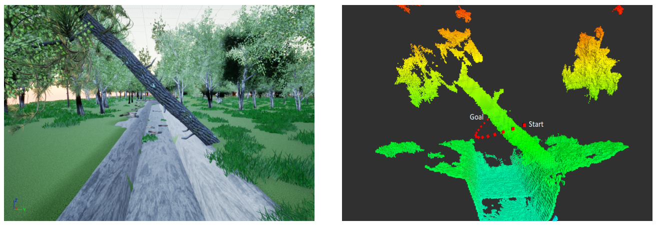
3) Scenario 3: Bridge Avoidance: Bridges are perhaps the only rigid obstacles in a canal environment that comes directly into the heading of the MAV. It is necessary to test the behavior of our approach in navigating over bridges. As shown in Figure 14(a), a goal point is set beyond the bridge. In this case, the path planner can provide a path above or under the bridge. In our experiments, we have seen both types of paths returned by the algorithm.



(a) (b)

Fig. 14. (a) The ground truth of the simulation scene where the MAV has to navigate over a bridge (b) In this setup, the start and goal are set in such a way that the bridge comes directly into the MAV's heading, hence, the MAV has to avoid.

4) Scenario 4: Tree Trunk Avoidance: As shown in Figure 15, another possible obstacle that can happen in the MAV navigation during the canal mapping could be a tree trunk. In this case, the goal point is set in such a way that the trunk lies in the middle of the MAV and goal position and the vehicle has to change its path and avoid it. From Figure 15(b) we can see that the proposed framework has successfully mapped and avoided the tree trunk.



(a) (b)

Fig. 15. The actual simulated scene of the canal where the MAV has to avoid collision with tree trunks.

V. Conclusion

In this paper, we proposed a framework in a simulated environment for the autonomous navigation and mapping of water channels using a Micro-Aerial Vehicle (MAV). The complete framework consists of three main modules (mapping, path planning, and mission planner) that gradually explore the complete canal while solving for start to local goal queries. We use the advantage of Octomap for mapping the environment and then implemented Informed-RRT* on the octomap to plan obstacle-free paths. We have not assumed any feature of the map to be known *apriori* for the implementation of the path planning, instead, our path planning method provides a path within the perception of the stereo camera mapping range which in our case is 25 meters. Navigating and mapping the whole canal is the responsibility of the mission planner which assumes intermediate waypoints on the canal as local goal points and iteratively requests the path planning algorithm for solving the queries for each local goal point. We used Microsoft Airsim in the Unreal engine for our simulation and built a realistic (near to real) canal densely cluttered with obstacles such as hanging tree branches, trunks, and bridges. Our methods show promising results in both mapping and path planning. Future work includes the implementation of the frontier algorithm for local goals assignment, the implementation of SLAM to localize the MAV instead of using GPS/INS, and testing the framework on hardware.

A working demo of the framework can also be watched in the video provided at this link https://youtu.be/Jb0h1OKiM3s

VI. Acknowledgement

This work was supported by the Higher Education Commission (HEC) of Pakistan at the NCRA Agricultural Robotics Lab, based at LUMS. Technical support was also provided by the Center for Water Informatics & Technology (WIT), LUMS.

References

[1] A. Qureshi, P. McCornick, M. Qadir, and Z. Aslam, "Managing salinity and waterlogging in the indus basin of pakistan," *Agricultural Water Management*, vol. 95, pp. 1–10, 02 2008.

[2] J. W. Bhatti, M. A.; Kijne, "Irrigation management potential of paddy/rice production in punjab of pakistan." in *Proceedings of the International Workshop on Soil and Water Engineering for Paddy Field Management*, 1992, pp. 355–366.

[3] H. Anwar, A. Muhammad, and K. Berns, "A framework for aerial inspection of siltation in waterways," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 6273–6278.

[4] A. S. Gadre, S. Du, and D. J. Stilwell, "A topological map based approach to long range operation of an unmanned surface vehicle," in *2012 American Control Conference (ACC)*, 2012, pp. 5401–5407.

[5] S. N. Khan, T. Mahmood, S. I. Ullah, K. Ali, and A. Ullah, "Motion planning for a snake robot using double deep q-learning," in *2021 International Conference on Artificial Intelligence (ICAI)*, 2021, pp. 264–270.

[6] P. Liljebäck, Stavdahl, K. Y. Pettersen, and J. T. Gravdahl, "Mamba - a waterproof snake robot with tactile sensing," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 294–301.

[7] S. Rathinam, P. Almeida, Z. Kim, S. Jackson, A. Tinka, W. Grossman, and R. Sengupta, "Autonomous searching and tracking of a river using an uav," in *2007 American Control Conference*, 2007, pp. 359–364.

[8] J. Yang, D. Rao, S.-J. Chung, and S. Hutchinson, *Monocular Vision based Navigation in GPS-Denied Riverine Environments*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2011-1403

[9] S. Scherer, J. Rehder, S. Achar, H. Cover, A. D. Chambers, S. T. Nuske, and S. Singh, "River mapping from a flying robot: state estimation, river detection, and obstacle mapping," *Autonomous Robots*, vol. 32, no. 5, pp. 189 – 214, May 2012.

[10] S. M. Abbas, H. Ali, and A. Muhammad, "Autonomous canal following by a micro-aerial vehicle using deep cnn," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 243–250, 2019, 6th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319324486

[11] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 2878–2883.

[12] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 2997–3004.

[13] J. Yang, D. Rao, S.-J. Chung, and S. Hutchinson, "Monocular vision based navigation in gps-denied riverine environments," *AIAA Infotech at Aerospace Conference and Exhibit 2011*, 03 2011.

[14] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," *CoRR*, vol. abs/1705.05065, 2017. [Online]. Available: http://arxiv.org/abs/1705.05065

[15] J. L. Patrick Mihelich, Kurt Konolige. Stereo_image_proc. [Online]. Available: http://wiki.ros.org/stereo_image_proc

[16] Flexible collision library. [Online]. Available: https://flexible-collision-library.github.io/

[17] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.