Informative and Fast Exploration Planning Using UAV for Reconnaissance Operations

Kaleb Ben Naveed*[1], Brady Moon[2] and Sebastian Scherer[2]

***Abstract*— In the missions related to search and rescue operations, reconnaissance Unmanned Aerial Vehicles (UAV) are used to effectively search the given environment map and return information about the detected objects with limited flight time. This involves solving the NP-hard problem of maintaining balance between the tasks of fast exploration and data acquisition. Most of the existing work focuses on optimizing only one of these factors. In this paper, we propose Prioritized-FUEL, which is built on top of the FUEL (Fast UAV Exploration) algorithm, a frontier-based exploration technique. The proposed hierarchical structure maintains balance between fast coverage and data acquisition through the introduction of two high-level planner options: Exploration planning and Informative planning. In order to facilitate decision making for informative planner, we modify Frontier Information Structure (FIS) in the original FUEL paper to incorporate information about objects of interest. Moreover, we introduce Frontier Priority Que (FPQ) to store information about all the frontiers, which have a higher probability of the presence of the objects of interest near them. The results from the experiments in the light UAV simulation environment show that the proposed method resulted in almost 2 times faster data acquisition as compared to the original FUEL algorithm.**

***Index Terms*— Search and Coverage, Informative Path Planning, Fast Exploration**

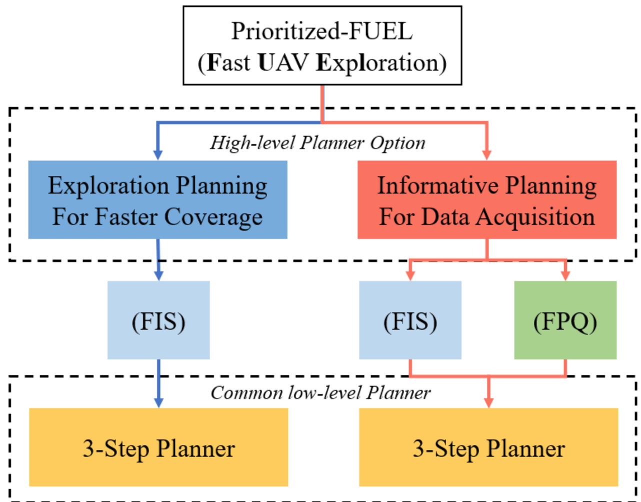

Fig. 1. The Prioritized-FUEL (Fast UAV Exploration) proposes a hierarchical structure with two high-levels options: Exploration planning for faster coverage and Informative planning for data acquisition. High-level option of Informative planning uses **Frontier Priority Que (FPQ)** in addition to **Frontier Information Structure (FIS)** for decision making. At lower level, 3-step planner is used for trajectory generation. Both high-level options use same low-level 3-step planner. The only difference is the input to 3-step planner.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) are widely used for the purpose of data acquisition tasks. Some of the examples of these tasks include victim search in rescue operations [1], fault inspection for infrastructure [2], and water body exploration for ecosystem management [3]. The main dilemma for UAVs in these operations is maintaining a balance between the tasks of fast exploration and information gathering. This is specifically important in a scenario of search and rescue, where the UAV does not only has to explore the region faster but also has to periodically pause exploration to focus on potential interesting areas where a victim could be present.

Some of the work [4][5] proposed for the task of optimal and rapid exploration have shown great results in the real world settings but do not consider information gain. On the other hand, some of the work [6][7] proposed on informative path planning or uncertainty reduction, which have shown promising results by maximizing information gain during exploration tasks, lack fast coverage guarantees. Thus most of the existing work either focuses on rapid exploration or information gain, which limits their ability to provide optimal solutions in reconnaissance operations.

In order to solve this problem, we propose, Prioritized-FUEL, which is inspired from FUEL (Fast UAV Exploration) algorithm. The proposed method add the ability in existent FUEL framework to balance exploration and exploitation for information gain by using hierarchical structure, which contains two high-level planning options: Exploration planning for faster coverage and informative planning for data acquisition. The High-level option of exploration planning is responsible for the task of fast coverage of the search space, while the high-level option of informative planning is responsible for the task of information gathering of the detected objects. The selection of the high-level option is made based on the contents of Frontier Information Structure (FIS) and the proposed Frontier Priority Que (FPQ). FIS contains essential information about the search space and frontiers. We modify the original structure of FIS introduced in the FUEL algorithm [8] according to our decision making requirements. Details about modification can be found in Section III and Section IV.

The High-level option of Exploration planning only uses

[1]Student of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, China. kaleb-ben.naveed@connect.polyu.hk

[2]The Robotics Institute, Carnegie Mellon University

* Supported by CMU Robotics Institute Summer Scholars Program

FIS; however, the Informative planning option uses both FIS and FPQ for decision making and planning stage. The overall structure can be seen in Figure 1. The proposed FPQ stores information about the frontiers, which have an overlap with the bounding box of the detected objects. After the high-level planner option is selected, a 3-step planner is used for generating minimum-time trajectories. The three steps include creating optimal global paths by posing the problem as the Travelling Salesman Problem (TSP), refining local viewpoints for maximum coverage, and generating minimum-time B-spline trajectories which are safe, obstacle-aware, and dynamically feasible.

The main contributions of this paper can be summarised as follows:

- The hierarchical structure, which ensures balance between fast coverage and data acquisition through high-level options of Exploration planning and Informative planning;
- The Frontiers Priority Que (FPQ) that facilitates decision making for high-level planner options by storing information about the frontiers, which have an overlap with detected objects bounding box.

The remainder of the paper is organized as follows: Section II gives a detailed overview of the related work done in the scope of exploration planning and informative path planning (IPP). Section III overviews system and preliminaries. Section IV describes the proposed methodology in detail. Section V evaluates the proposed methodology, lists down preliminary results and comments on future work and, lastly, section VI concludes the paper.

II. RELATED WORK

The area of exploring and mapping unknown environments through mobile robots has received considerable attention in the recent past. Some of the state of the art approaches for mapping and exploration include frontier based approaches, information-theoretic approaches and adaptive sampling based approaches.

Frontier based approaches [9] for exploration are geometric in nature and explore the region by travelling to the boundaries between unknown and known regions. These boundaries are called frontiers. According to the original approach, during the exploration task, the closest frontier is chosen greedily as the next frontier to visit. There are a number of improvements made on this method. Instead of greedily selecting the closest frontier, [4] selects the next frontier that minimizes the velocity change to ensure maximum exploration speed. [10] introduced a method, which generated shorter exploration trajectories, by amalgamating a frontier based approach with local vector field strategy. In order to solve the problem of optimal coverage and fast exploration, [8] proposed the FUEL (Fast UAV Exploration) algorithm. This method proposed the Frontier Information Structure (FIS), which contains important information about the search space and is updated incrementally as exploration continues. By using FIS, they proposed a hierarchical 3-step planner for trajectory generation. The three steps are finding global coverage paths, refining the local set of viewpoints, and generating minimum-time B-spline trajectories. This approach resulted in much faster exploration of the search area but did not prioritize information gain for the objects of interest, thus limiting its use for search and rescue operations.

Another approach, Adaptive sampling, requires random sampling of the search space to create viewpoints for exploration planning. [11] proposed an algorithm called Adaptive Search Space Coverage Path Planner (ASSCP) to generate a set of viewpoints by performing adaptive sampling that directs research towards areas with low accuracy and low coverage. [12] presented a new RRT*-inspired algorithm, which continuously expanded the single tree of candidate trajectories and refined intermediate paths. This method ensured global coverage and path utility function maximization. Some work involving a team of robots has also been explored. [13] proposed a method, using an adaptive sampling based approach, exploiting a team of Autonomous Underwater vehicles (AUVs) to explore the region. In this method, overall search space was partitioned in the regions close to each given AUV using voronoi diagrams and each robot runs adaptive sampling within its partition using map entropy of the environment. The environment used in this method has communication constraints and requires vehicles to initiate data sharing after some time. [14] further improved the partitioning procedure of search space by proposing voronoi partitioning which considers newly discovered obstacles and also updates regions continuously to improve load balancing between robots. The sampling-based approaches have shown state of the art results but they are computationally expensive which limits their usage for real world applications.

An alternative approach to sampling-based and frontier-based approaches include Information-Theoretic Planning. These methods normally optimize an information theoretic measure for exploration. [15] used a map entropy measure to select the next frontier to visit in a Frontier-based approach. [6] proposed method for information-theoretic planning approach, which chooses a trajectory from a set of global and local trajectories. Then they use gradient-based optimization to refine the chosen trajectory to maximize the Cauchy-Schwarz quadratic mutual information (CSQMI) objective. [16] proposed a method for target search problem which combines informative planning and obstacle awareness.They used layered optimization approach using Bayesian Optimization (BO) that balances the exploration-exploitation trade off between information gain, altitude dependent sensor performance, Field of View (FOV) and target re-observation. Another approach proposed by [7] solved the problem of exploration and informative planning by posing the problem as a correlated orienteering problem and travelling salesman problem. The proposed method provided anytime solutions in adaptive scenarios and also used a multiresolution sensor to gather target information.

In this paper, our approach is based on the work of [8] and add the ability in the structure to prioritize information gain for the detected objects of interest while maintaining a fast exploration rate. We propose Prioritized-FUEL, which

TABLE I
FRONTIER INFORMATION STRUCTURE OF THE CLUSTER

Data	Description
$Cell_i$	Frontier cells that belong to the cluster
$Cell_{avg,i}$	Average position of the cluster
BC_i	Bounding box of the cluster
BI_j	**Bounding box of the object of interest**
$Prob_{detect,j}$	**Probability of the Detected Object**
VP_i	Viewpoints around the cluster
$Cost_i$	Connection costs to other clusters

either selects a high-level exploration planner or informative planner based on information contained in Frontier Information Structure (FIS) and Frontier Priority Que (FPQ). Afterwards, both high level planners use common 3-step low-level planner to generate minimum-time trajectories.

III. PRELIMINARIES AND SYSTEM OVERVIEW

A. Frontier Information Structure

In frontier-based exploration [9], frontiers are defined as known-free voxel cells adjacent to the unknown cells. Clusters are defined as known-free voxel cells combined together. The method proposed by [8] introduced Frontier Information Structure (FIS) which provides richer and more organized information about the search space.

Whenever a new frontier Fr_i is detected, all the relevant information about that particular cluster is stored in the FIS using the cylindrical coordinate system. Table 1 summaries the data contained in the FIS. In our method, for the task of informative planning, we also add information about the object of interest to the FIS Structure. This includes the bounding box BI_j and the Probability of detection $Prob_{detect,j}$ of the j^{th} detected object. These two entries are used for decision making and will be explained later in the next section. When the map is updated, the information about the updated region is fetched and the bounding box $BB_{updated}$ is drawn around it. Afterwards it is checked if there is any overlap between the updated region bounding box $BB_{updated}$ and cluster bounding box BC_i. Similar to FUEL paper [8], for searching and clustering of new frontiers, we use region growing algorithm and then use Principal Component Analysis (PCA) to split each large cluster recursively in order to ensure robust decision making as large clusters do not help in characterizing different unknown regions.

B. Viewpoint Generation and Inter Frontier Cost Update

In our work, we use the methods proposed by [8] and [9] for the generation of viewpoints and for inter-frontier cost update. When a cluster Fr_i is created, the rich number of viewpoints $VP_i = \{x_{i,1}, x_{i,2}, ..., x_{i,n_i}\}$ are generated so that the viewpoint with the maximum coverage can be selected through optimization. For each viewpoint, the information about the sampled point P_i, in cylindrical coordinate system, and its yaw angle ψ is stored i.e. $x_{i,j} = (P_{i,j}, \psi_{i,j})$. In addition to generating viewpoints, we also compute costs between frontier clusters. The connection cost is calculated as time lower bound $t_{lb}(x_{k_1,j_1}, x_{k_2,j_2})$ between two viewpoints of clusters. The formula for calculating time lower

Algorithm 1 Prioritized FUEL

Initialize Frontier Information Structure FIS and Frontier Priority Que FPQ.
while Not whole region explored **do**
  Search for new frontiers fr_i
  Generate Viewpoints and inter frontier cost using time-lower bounds eq.1
  if $BC_i \cap BI_j$ **then**
    if $Prob_{detect,j} > \epsilon)$ **then**
      Append fr_i to FPQ
      while $FPQ \neq \emptyset$ **do**
        $InformativePlanner()$ // Information gathering
      end while
    end if
  else
    $ExplorationPlanning()$ // faster coverage
  end if
end while

bound is given in equation 1, where $P(p_{k_1,j_1}, p_{k_2,j_2})$ denotes collision free path between p_{k_1,j_1} and p_{k_2,j_2}, and v_{max} and ψ_{max} are velocity and yaw angle limits respectively. The collision free path is searched through A^* algorithm.

$$t_{lb}(x_{k_1,j_1}, x_{k_2,j_2}) = \max\Big[\frac{length(P(p_{k_1,j_1}, p_{k_2,j_2}))}{v_{max}}, \frac{\min(|\psi_{k_1,j_1} - \psi_{k_2,j_2}|, 2\pi - |\psi_{k_1,j_1} - \psi_{k_2,j_2}|)}{\psi_{max}}\Big] \quad (1)$$

IV. METHODOLOGY

In the proposed Prioritized-FUEL, we develop a hierarchical structure with two high-level options: Exploration planning and Informative planning. The overall structure is shown in Figure 1. The exploration planner is responsible for generating paths which result in faster coverage of the search area while informative path planner generates information-theoretic paths for gathering information about the detected objects. Both exploration and informative planners plan paths in three steps, which is similar to the original FUEL paper [8]. The detailed working of the Prioritized-FUEL is described in Algorithm 1 and explained below.

A. Detected Object Information and Frontier Priority Que

When new frontiers are searched, the information about all the clusters is maintained in the FIS. For the purpose of informative path planning, we also keep the information about detected objects, if any, in the FIS structure. According to the Table 1, we keep track of the bounding box BI_j and probability of the confidence of detection $Prob_{detect,j}$ of the detected object.

If and when the object is detected, the bounding box BI_j is drawn around it. Then precise checks are made for all clusters and a list of all those clusters is returned whose bounding box BC_i intersects with the detected object bounding box. Afterwards, the probability of confidence of the detected object is checked and if it is greater than ϵ, then that cluster is added into the Frontier Priority Que (FPQ). Then the latter check is made to avoid the presence of false positives in detection.

B. Exploration vs Information theoretic Planning

Based on the status of the FPQ, different planners are activated accordingly. The only difference between informative planner and exploration planner is that in the former some of the clusters are given higher priority based on the possibility of the presence of an object of interest, while exploration planner does not prioritize any frontier but creates a minimum-time trajectory between any set of candidate clusters. The two different cases are mentioned here for further elaboration.

*1) **FPQ contains at least one cluster:*** If FPQ contains at least one cluster, the high-level option of exploration planning will pause and an informative planning option will be activated. In this case, clusters present in the FPQ are given higher priority and global paths are created to cover those clusters first. During this time, the UAV focuses on gathering data about the detected objects rather than exploring new areas. This might decrease the overall search time but ensures faster detection of the objects of interest, which can save lives in search and rescue operations. Here for data gathering, instead of circling around the object as in [7], we purely rely on local viewpoint refinement to select viewpoints which give maximum information of the object while not wasting energy circling around the object. The description of the working of the 3-step planner is given in the next subsection.

*2) **FPQ contains no cluster:*** If FPQ contains no cluster, then an exploration planner is chosen. Under the exploration planner, minimum-time trajectories are generated through all active clusters. This happens incrementally as new clusters are made along the search path. The purpose of the exploration planner is to ensure fast coverage of the search area. The detailed steps involved in generating minimum-time trajectories through a 3-step motion planner are mentioned in the next section.

C. 3-Step Planner

The 3-step planner is adopted from the original FUEL algorithm structure [8] and some modifications are made to incorporate our structure into it . The low-level planning procedure for both the high-level exploration planning option and informative path planning option is the same. The only difference is that during informative path planning, only selected or prioritized clusters are considered, while exploration planner considers all active clusters. The overall low-level 3-step planner includes Global Path Planning, Local Viewpoint refinement, and minimum-time B-spline trajectory generation. The overall structure can be shown in Figure 2.

*1) **Global Path Planning:*** Global planner creates a global path through the planner by posing the problem as the Asymmetric Travelling Salesman Problem, which creates an open-loop tour starting from the current viewpoint of

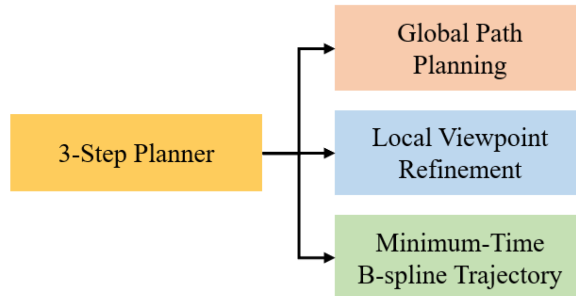

Fig. 2. The 3-step planner include Global Path Planning, Local Viewpoint Refinement and Minimum-time B-spline Trajectory Generation

cluster C_i and passing to all clusters. The cost from the current viewpoint x_0 to the x_k cluster can be evaluated using equation 2. Here time-lower bound $t_{lb}(x_0, x_{k,1})$ is used which was calculated and stored in FIS when frontiers were detected.

$$TSP_{cost}(x_0, x_k) = t_{lb}(x_0, x_{k,1}) + w_c \cdot c_c(x_{k,1}), \quad where, k \in \{1, 2, 3, ...N_{cluster}\} \tag{2}$$

In equation 2, $c_c(x_{k,1})$ is used as motion consistency cost which eliminates inconsistency by penalizing large changes in flight conditions. This inconsistency might rise due to several paths having similar time-lower bound.

*2) **Local Viewpoint Refinement:*** Here in this second step, the global path is improved based on the different viewpoints which were computed earlier. In original classical frontier-based approach, while calculating the global path, only a single viewpoint from each cluster is considered, which might not provide optimal collective coverage. For local viewpoint refinement, we create a graph of nodes from the current viewpoint x_0 to all viewpoints VP_i. In our method, we consider all active clusters and use the notation N_{at} to refer to them. After connecting nodes between clusters through directed edge, Dijkstra algorithm is used to search for the optimal local tour by minimizing the cost $LT_{cost}(x_0, x_{N_{at}})$ shown in equation 3. This approach is similar to some of the other proposed methods [8], [5], and [17].

$$LT_{cost}(x_0, x_{N_{at}}) = t_{lb}(x_0, x_{1,j_1}) + w_c \cdot c_c(x_{1,j_1}), + t_{lb}(x_{N_{at},j_{N_{at}}}, x_{N_{at}+1,1}) + \sum_{k=1}^{N_{at}-1} t_{lb}(x_{k,j_k}, x_{k+1,j_{k+1}}) \tag{3}$$

*3) **B-spline Trajectory generation:*** For generating minimum-time B-spline trajectory, we use the method developed by [8] and [18]. The trajectory planner generates smooth, safe, and dynamically feasible B-spline trajectories, and also optimizes all the parameters for the B-spline, which result in minimum-time trajectories.

The quad-rotor used during experiments is considered to be flat, so thus flat outputs include $x \in (p, \psi)$ where

Fig. 3. Here the UAV is shown ready for exploring the area. The light blue point cloud represents occupied region and colourful boundaries around occupied region represents frontier clusters. The red rectangle represents search area specified. The bounding box BI_j represents the bouding box around the object of interest. The grey shadows are the obstacles present in the environment.

$p \in (x, y, z)$. Thus the output can be shown as $X_{c,b} = \{x_{c,0}, x_{c,1}, ..., x_{c,N_b}\}$, where $x_{c,i} = (p_{c,i}, \psi_{c,i})$ are the N_b+1 are control points in p_d degree uniform B-spline. The knot number represents a number of control points used with a curve degree. The knot span here is referred to as Δt_b. The overall optimization problem can be written as equation 4 and we suggest the reader to refer to work of [8] and [18] for more details.

$$\underset{X_{c,b}, \Delta t_b}{\arg\min} \, fs + w_t T + \lambda_c f_c + \lambda_d (f_v + f_a) + \lambda_{bs} f_{bs} \quad (4)$$

Here in this equation, fs is the elastic band smoothness cost, R_s is the penalty matrix and f_c, f_v, f_a are penalties to ensure safe and dynamic feasibility and T is the total trajectory time. The detailed equations can be found in [8] and [18].

V. EXPERIMENTS

A. Scenario and Experiment Setup

In this section, we test the proposed Prioritized-FUEL algorithm in light simulator. The purpose of this experiment is to test the validity of the idea and its preliminary performance. The screenshot from the simulator is shown in Figure 3. The red boundary specifies search area, which needs to be explored. The black coloured bounding boxes BI_j represent bounding boxes which are drawn around objects of interest when detected. The grey color shadows represent unknown obstacles. Initially, UAV does not have any information about the presence of objects of interest or obstacles.

In our work, we assume that we have a perfect perception system, so that we can focus on improving the planning part of the system. This assumption helps us with two important pieces of information. Firstly, the system knows the exact position of the object of interest so a perfect bounding box BI_j can be drawn around it, but UAV does not have any clue about the position of the object of interest at the start of the exploration process. It only comes to know about the presence of the object of interest when the bounding box

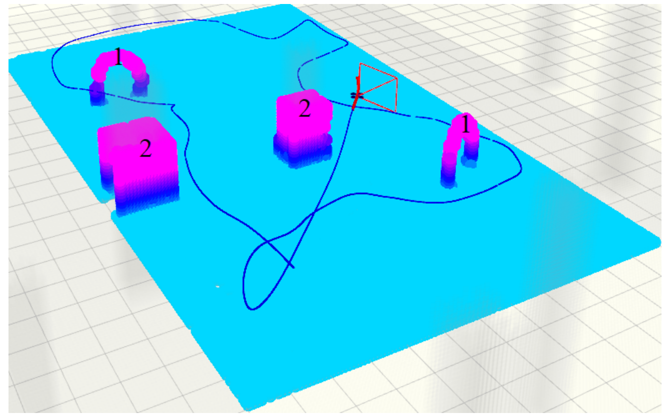

Fig. 4. The trajectory of the UAV is shown after fully exploring the region. The light blue point cloud represents the known region. objects with label "1" are objects of interest. Objects with label "2" are obstacles.

BI_j of the object of interest intersects with the bounding box BC_i of the frontier cluster. Secondly, we assume that $Prob_{detect,j}$ is always greater than ϵ, which is one of the requirements for informative planning as shown in algorithm 1. In experiments, with a focus on both perception and planning, bounding box BI_j of the object of interest and $Prob_{detect,j}$ would change as detection is carried out using some detection models [19], which might cause instability in the execution of planning routines. But in our system, this information is stable and reliable as we assume perfect perception system.

B. Results and Discussion

In order to evaluate the effectiveness of our proposed algorithm, Prioritized-FUEL, we compared it with the FUEL algorithm using the mentioned metrics over sample size of 10 experiments for each method:

- **Data Acquisition Time**: Time spent on exploring frontier clusters, whose bounding box BC_i have an overlap with the bounding box of the object of interest BI_j, once object of interest is detected;
- **Total Exploration Time**: Total time spent exploring the whole region.

The detailed results are shown in Table II. In our given simulation environment, Prioritized-FUEL was able to outperform the FUEL algorithm in the data acquisition part by focusing on exploring objects of interest first if detected, while keeping total exploration time almost the same.

As shown in Table II, the original FUEL algorithm spends on average 4.31 seconds on exploring the frontiers near objects of interest once the object is detected. While the proposed Prioritized-FUEL algorithm spends on average 2.38 seconds on exploring the frontiers near objects of interest.

TABLE II

COMPARISON OF FUEL AND PRIORITIZED-FUEL

Method	Data Acquisition Time (sec)				Total Exploration Time (sec)			
	Avg	Std	Min	Max	Avg	Std	Min	Max
FUEL [8]	4.31	1.38	1.79	9.00	55.50	2.99	49.21	59.43
Prioritized-FUEL	**2.28**	0.80	1.78	**4.21**	56.60	3.60	48.87	61.21

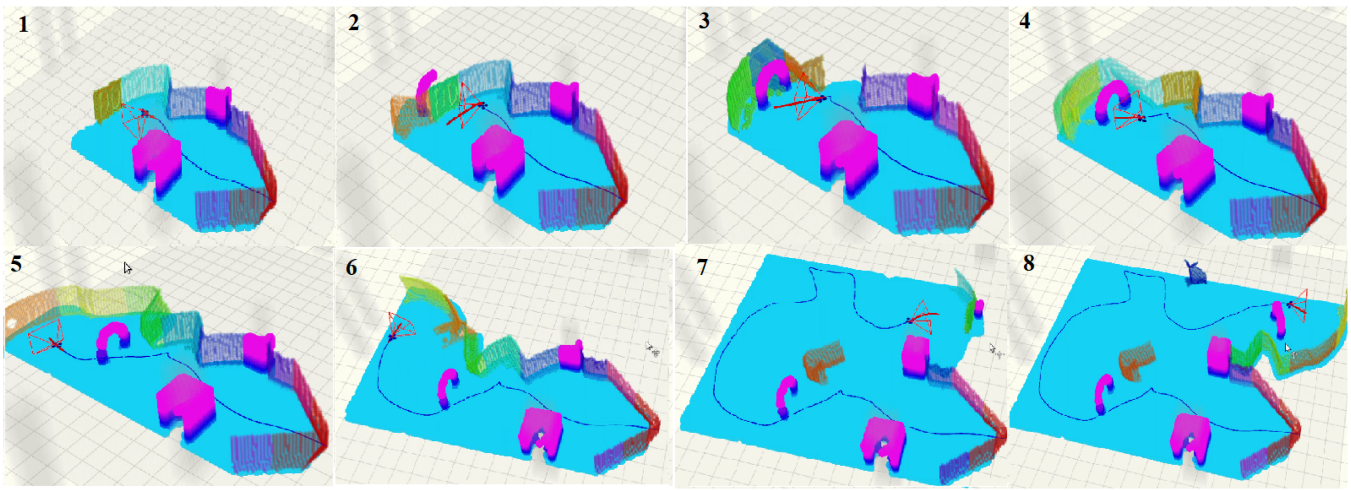

Fig. 5. The overall sequence of exploration is shown from one of the experiments from frames 1 to 8.

This is because in the original FUEL algorithm, UAV considers all frontiers as the same and generates minimum-time trajectories between them. This results in faster exploration of the environment. While in the Prioritized-FUEL algorithm, frontier clusters whose bounding box BC_i overlap with the object of interest BI_j are given higher priority and are added into Frontier Priority Que (FPQ). Thus UAV explores the high priority frontiers first by generating minimum-time trajectories between them, before exploring other normal frontiers.

Moreover it is shown in Table II that the total exploration time is not much affected. This is mainly because even while planning informative paths for data acquisition, minimum-time trajectories are generated and viewpoints are adjusted accordingly using Local Viewpoint refinement to maximize coverage. This shows that Prioritized-FUEL not only guarantees faster data acquisition but also faster exploration.

The sequence of the exploration from one of the experiments can be seen in Figure 5. The UAV can be seen to focus on objects of interest in frames 3 to 5 once it detects the first object of interest (cicle) in frame 2. This can also be seen in frame 7 when the second object of interest (circle) is detected by checking the overlap between the bounding box of the cluster and of the circle. The final map is shown in Figure 4.

C. Future Work

The Proposed structure Prioritized-FUEL (Fast UAV Exploration) ensures a balance between faster coverage and informative planning. There are several more components which can be added to the structure to make it more robust. This includes altitude-aware data acquisition, multi-UAV search and further testing in high fidelity simulations such as unreal engine and AirSim simulator.

Considering altitude while gathering information can help reduce uncertainty in sensor measurements. Obstacle-aware Adaptive Path Planning (OA-IPP) [12] incorporates altitude in it and shows that altitude-dependent sensor performance can be incorporated into cost or objective function. Moreover, Multi-UAV Search can help reduce overall search time by dividing search effort between multiple UAVs. One of the approaches famously used in the literature to divide the whole search region into partitions is voronoi partitions. One of the approaches proposed by [14] improved the partitioning procedure of search space by proposing voronoi partitioning which considers newly discovered obstacles and also updates regions continuously to improve load balancing between robots.

VI. CONCLUSION

In this paper, we proposed Prioritized-FUEL (Fast UAV Exploration) in order to ensure balance between fast exploration and data acquisition during reconnaissance operations. The hierarchical structure provides two high-level options: Exploration planning for faster coverage and Informative planning for data acquisition. In order to facilitate decision making for informative planner we modify Frontier Information Structure in the original FUEL paper to incorporate information about objects of interest. Moreover, we introduce Frontier Priority Que (FPQ) to store information about all the frontiers, which have a higher probability of the presence of the objects of interest near them. We test the proposed framework in a light UAV simulator and show that the prioritized-FUEL algorithm decreases the time to explore objects' interest by almost 40% as compared to the original FUEL algorithm while keeping total exploration time of the search space almost the same. In future, we want to test and compare the algorithms in the AirSim simulator with the scenario of open sea reconnaissance operations.

ACKNOWLEDGMENT

This work is supported by the Robotics Institute Summer Scholars Program (RISS), Carnegie Mellon University, and The Hong Kong Polytechnic University. Also special thanks to Rohit Garg for sharing his experience with frontier-based exploration.

REFERENCES

[1] S. Waharte and N. Trigoni, "Supporting search and rescue operations with uavs," in *2010 International Conference on Emerging Security Technologies*, 2010, pp. 142–147.

[2] M. Korki, N. D. Shankar, R. Naymeshbhai Shah, S. M. Waseem, and S. Hodges, "Automatic fault detection of power lines using unmanned aerial vehicle (uav)," in *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, 2019, pp. 1–6.

[3] P. Sujit, J. Sousa, and F. L. Pereira, "Uav and auvs coordination for ocean exploration," in *OCEANS 2009-EUROPE*, 2009, pp. 1–7.

[4] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2135–2142.

[5] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt, "Efficient autonomous exploration planning of large-scale 3-d environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1699–1706, 2019.

[6] B. Charrow, G. Kahn, S. Patil, S. Liu, K. Goldberg, P. Abbeel, N. Michael, and V. Kumar, "Information-theoretic planning with trajectory optimization for dense 3d mapping," 07 2015.

[7] S. Arora and S. Scherer, "Randomized algorithm for informative path planning with budget constraints," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4997–5004.

[8] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "FUEL: fast UAV exploration using incremental frontier structure and hierarchical planning," *CoRR*, vol. abs/2010.11561, 2020. [Online]. Available: https://arxiv.org/abs/2010.11561

[9] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, 1997, pp. 146–151.

[10] R. Shade and P. Newman, "Choosing where to go: Complete 3d exploration with stereo," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2806–2811.

[11] R. Almadhoun, T. Taha, D. Gan, J. Dias, Y. Zweiri, and L. Seneviratne, "Coverage path planning with adaptive viewpoint sampling to construct 3d models of complex structures for the purpose of inspection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7047–7054.

[12] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robotics and Automation Letters*, vol. 5, pp. 1500–1507, 2020.

[13] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme, "Multi-robot coordination through dynamic voronoi partitioning for informative adaptive sampling in communication-constrained environments," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2124–2130.

[14] A. Dutta, A. Bhattacharya, O. P. Kreidl, A. Ghosh, and P. Dasgupta, "Multi-robot informative path planning in unknown environments through continuous region partitioning," *International Journal of Advanced Robotic Systems*, vol. 17, no. 6, p. 1729881420970461, 2020. [Online]. Available: https://doi.org/10.1177/1729881420970461

[15] R. Rocha, J. Dias, and A. Carvalho, "Cooperative multi-robot systems a study of vision-based 3-d mapping using information theory," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 384–389.

[16] A. A. Meera, M. Popović, A. Millane, and R. Siegwart, "Obstacle-aware adaptive informative path planning for uav-based target search," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 718–724.

[17] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1462–1468.

[18] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," 2019.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.