Fast Radioactive Hotspot Localization Using a UAV

Abdullah Al Redwan Newaz, Sungmoon Jeong, and Nak Young Chong

***Abstract*— We address the problem of hotspot seeking in an unknown radiation field using an Unmanned Aerial Vehicle (UAV) with limited resources. For on-the-spot investigation of accidental radiation releases, without *a priori* knowledge on the whereabouts of the source of radiation substances leakages, it is very difficult to navigate and return a UAV for fast hotspot localization. We propose a novel Hexagonal Tree (HexTree) based sampling algorithm to find such an optimal tour path based on the appropriate measurement locations. We make a realistic assumption on the environment, theoretically analyze the optimality of proposed algorithm, and numerically compare the performance with the existing method. The proposed algorithm gives faster convergence to the hotspot, an optimal exploration termination condition, and more informative locations while returning to the initial position than conventional random sampling based exploration and path smoothing algorithms.**

I. INTRODUCTION

The radiation hotspot is the zone where the level of radiation is significantly greater than in other regions. The formation of hotspot depends on the geometric dimension and spatial distribution of radiation elements. The localization of hotspot is of great importance to understand the range of radiation effects over contaminated areas. Efficient emergency relief and disaster recovery coordination can be facilitated by immediate hotspot localization using unmanned aerial vehicles (UAVs) that must be accomplished within limited time frame. Since the UAV equipped with dedicated sensors needs to explore over large areas, an efficient UAV navigation algorithm is crucial for fast hotspot localization.

Hotspot seeking is similar to the extremum seeking problem which has been widely studied in robotics [1], [2], [3]. The goal is to plan an optimal path in which the UAV can find an unknown hotspot location, while minimizing the exploration cost. Although these approaches are often implemented via gradient-based methods, in practice, it is unlikely to get a significant gradient difference at every exploration step especially in large areas. Likewise, when there is no distinct hotspot, the exploration cannot be terminated. To cope with such problems caused by similar radiation levels, the randomly exploring information gathering (RIG) algorithm is a worthy candidate [4], which is computationally efficient and ensures an asymptotically optimal solution to achieve information gathering in continuous space with motion constraints. However, when there is a specific hotspot, the performance of the RIG should be further improved, leading to more efficient methods of exploration. Specially, the exploration performance can be evaluated using the area coverage metric, which is the amount of sample points required to converge to the hotspot. Furthermore, after localizing the hotspot, a UAV return path should be generated that connects the hotspot and the UAV initial position. This loop closure is necessary to generalize the informative path and the measurement uncertainty assessment of the same locations, and to stop exploring due to emergency reasons, etc.

Thus, the hotspot localization problem is the combination of UAV path planning, area coverage, and loop closing. We propose a hexagonal sampling based approach that constrains the UAV location to a finite set of hexagonal grids, and a dynamic programming based optimal loop closing approach controlling the trade-off between the travel cost and the information gain. We combine these three problems to find an effective yet efficient hotspot localization solution. We theoretically investigate the upper bound of the sampling path and numerically compare the efficiency of the proposed algorithm with the existing one. The main contributions of this paper are as follows:

1) *a novel hexagon tiling based workspace decomposition,*
2) *efficient HexTree based sampling strategy,*
3) *optimal return path generation strategy, and*
4) *theoretical and numerical analysis.*

II. RELATED WORK

In contrast to conventional path planners finding the path between the current position and the goal position, the goal (hotpsot) position is unknown in our case. Since the UAV detects the hotspot only with the intensity measurement, our problem is closely related to the active sampling, selecting observation locations that minimize the prediction uncertainty or maximizing the information gain [5], [6].

Earlier studies were concerned with the localization of sources that do not affect one another [7]. However, the hotspot is no longer coincident with the source position, if the cumulative effect of sources exists. Several strategies attempted to find the radiation hotspot generated by multiple sources [8], [9], which can be generalized into the model-free and model-based approaches. The model-free approaches are extremum seeking methods, where the gradient ascending or the maximum likelihood path is followed. They tend to converge to local maxima [10]. In the context of model-based approach, source seeking can be performed using either the mutual information (MI) [11], [12] or MI gradient [13]. While such algorithms have been shown to be useful for a range of applications, they typically rely on restrictive

The authors are with the School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa 923-1292. {redwan.newaz, jeongsm, nakyoung}@jaist.ac.jp

assumptions on the field and do not explicitly optimize the exploration path.

In the grid based approaches, the area is decomposed into a finite number of rectangular cells [14]. A topographic mapping based exploration strategy was proposed to localize multiple sources in our earlier work [15]. Similar approaches can be found in [16]. Probabilistic random sampling based exploration can also be applied when the radiation fields are considered as vector fields [17]. Even though these aforementioned methods perform heuristically well, they overlook the area coverage issues and the exploration was not guaranteed to terminate optimally.

The computational complexity of the planner such as submodular function based near optimal path planning [18], maximum entropy sampling [19], maximum mutual information [20] does not suit on-line implementation. Recent path planning algorithms have focused on the generation of approximate paths with limited computation. Our work extends these ideas to the domain of hotspot seeking in a radiation field. Hollinger *et. al.* proposed three variants of the random sampling based information gathering algorithm subject to a budget constraint [4]. Out of their three algorithms, namely, RIG-roadmap, RIG-graph and RIG-tree, they conclude that the RIG-tree is the best in terms of effectiveness. In this work, we take the hotspot seeking problem as the information gathering problem considering a threshold value to terminate the exploration. Focusing on the area coverage issues, we generate a regularly spaced hexagonal grid of sampling points. After localizing the hotspot, we propose a loop-closing path, allowing the UAV to return back to its initial position while visiting the informative locations. Fig. 1 shows all the necessary steps for the proposed system.

III. Problem formulation

A hotspot can be generated by multiple sources with unknown strength in a 2D area. Let $\mathbf{x_H} = \langle x_h, z_h \rangle$ denote the hotspot, where the postion of it is given by x_h and the expected measurement is a positive real number, $z_h \in \mathbb{R}_+$. A radiation field generated by multiple sources can be characterized using Gaussian Mixture Model [15]. Therefore, $\mathbf{x_H}$ can be remotely traced by the radiation sensors mounted on the UAV. For simplicity and without loss the generality, we assume that the measurement of the hotspot, z, from a remote location, x, can be given by

$$z\left(x\right) = z_h \exp\left(-\frac{||x - x_h||}{\sigma}\right), \tag{1}$$

where σ is the spread of the measurement and z_h is the strength of hotspot. If we assume that measurements are spatially distributed throughout the environment, then the hotspot seeking in a radiation field is somewhat similar to the informative path planning, where the planner queries the path subject to the maximal information gain, I. Let, at time t, the entropy of previously gathered measurements be $H(z_{1:t})$ and the entropy of measurements given the location x_t be $H(z_{1:t}|x_t)$. The information gain is then computed by the absolute difference of these entropies as follows

$$I := abs\left(H(z_{1:t}) - H(z_{1:t}|x_t)\right). \tag{2}$$

Using this metric, one of the efficient way for path planning is the random sampling based approach [4]. In [4], information is gathered in two folds. Firstly, a number of sample locations are generated in a random manner and then measurements are gathered after traveling to each of them. Let $\mathbf{n}$ be the set of samples at time t such that $\mathbf{n}_t = \{1, 2, ..., n\}$. A tree τ is then constructed by iteratively sampling in neighbor locations. Thus, the goal of an informative path planner at time t is to find a maximal informative location around the neighbors in such a way that

$$x_t^* = \underset{i \in \forall \mathbf{n} \in \tau}{\arg\max} \ \ abs\left(H(z_{1:t}) - H(z_{1:t}|x_t^i)\right). \tag{3}$$

If the sampling budget time T is given, the path $\mathcal{P}$ is then generated by assimilating all the local best locations given by

$$\mathcal{P} = \bigcup_t^T \{x_t^*\}. \tag{4}$$

However, the hotspot seeking algorithm is considered as a goal oriented problem whereas in [4] the informative path planning is basically a goal free problem. An informative path planner can then find the hotspot by simply adding a termination condition, z_h, such that

$$\mathcal{P}^* = \bigcup_t^T \{x_t^*\} \ s.t. \, z\left(x_t\right) < z_h. \tag{5}$$

The problem of hotspot seeking becomes complicated when the information metrics for each sample location does not vary significantly at each sampling step. In that situation, we define the area coverage ratio as the performance index of the planner. We assume that the target area can be fully characterized by sampling the finite number of spatially distributed locations. Given a target area, $\mathcal{A} \leftarrow \forall\left\{x\right\}$, let the sampled locations be denoted by the set, $\mathcal{D} \leftarrow \cup\{\exists x \in \mathcal{A}\}$. The UAV will visit only unexplored area and when $\mathcal{A} \backslash \mathcal{D} \leftarrow \{\}$, the area would be fully covered. Let $\mathcal{D}_{t+1}^i$ be the prediction over $\mathcal{D}_{t+1}^i$ given the next sampling location x_t^i, and $c\left(x_t^i\right) \triangleq \frac{|\mathcal{D}_{t+1}^i|}{|\mathcal{A}|}$ be the area coverage ratio at exploration time, t. Since expensive explorations are required to gather the measurement attributes, z_t, we want to minimize the area coverage ratio at each sampling step t. Thus, our desired hotspot directed sampling path can be expressed as follows

$$\mathcal{P}^* = \min_{c(x_t)} \max_{I(x_t)} \sum\nolimits_t \left(c\left(x_t\right) + \eta I\left(x_t\right)\right) s.t. \, z\left(x_t\right) < z_h, \tag{6}$$

where η is a normalizing constant. Although a sampling path is the necessary condition to localize an unknown hotspot, it is observed that all the sampled locations are not informative. Thus, when the UAV has to return back to its initial location, it can visit only to the most informative locations to make an efficient tour. In this way, we can then find the loop closure path by visiting only the most informative locations. Let W be the total information gain for the loop closure path, which

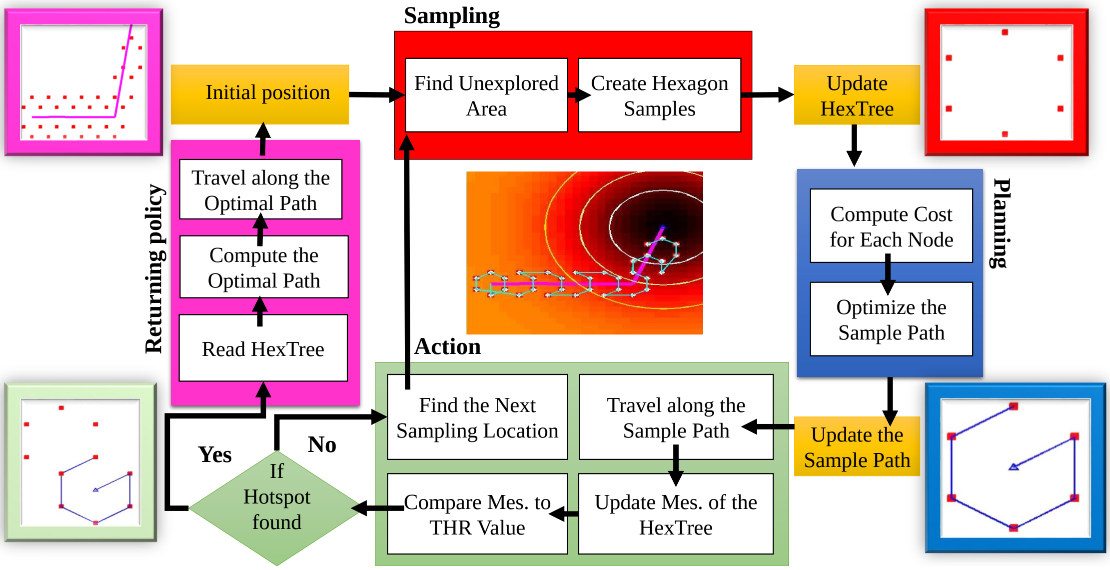

Fig. 1: **System Overview:** The radiation field is characterized by sampling finite locations which are distributed in hexagons manner. The hotspot localization is performed by sampling, planning and action phases. It is noteworthy that the robot not only finds the hotspot position but also returns to the initial position while visiting the most informative locations.

can be computed by visiting all the sampled locations. Since we already know all the sample locations $x \in \mathcal{D}$ and their corresponding information $I(x)$, the informativeness of each location can be evaluated by a weight, w, as follows

$$w(x) \triangleq \frac{I(x)}{dist(x, x_0)}, \tag{7}$$

where x_0 is the initial position of UAV and $dist$ is the function that computes the Eucledian distance. Therefore, the optimal returning path can be obtained through avoiding less informative locations given by

$$\mathcal{P}^*_{opt} = \max \sum_{x \in \mathcal{D}} I(x)\, x \;\; s.t. \sum_{x \in \mathcal{D}} w(x) x \; <= W\,. \tag{8}$$

In summary, the goal of this paper is to answer the following question- *given a termination threshold z_h of an unknown hotspot, $\mathbf{x_H}$; how to generate a tour path, $\langle \mathcal{P}^*, \mathcal{P}^*_{opt} \rangle$, to quickly localize the hotspot position x_t, where $z(x_t) > z_h$?*

IV. Algorithm Descriptions

We now discuss the sampling strategy that generates the UAV trajectories toward the hotspot, minimizing the area coverage ratio. The key idea is to use the samples over a set of hexagonal grid points and to build a tree of possible trajectories by extending candidate trajectories toward the sampled points. The hexagonal grid has two benefits: it covers more directions compared to the rectangular grid, and is the most efficient shape that can tile the 2D plane [21].

A. *Local field Sampling*

We propose a HexTree based sampling strategy inspired from the RIG-tree structure [4], where vertices in the graph represent a tuple of location, cost, and information. Unlike the RIG-tree, a HexTree has a fixed number (six) of children nodes, q_{hex}, centering at a (virtual) parent node v_t. As the name implies, the children nodes are hexagonally distributed, and each of them is separated from its neighbors by the same interval a, and the adjacent parents share common children nodes depending on their locations. The root of the tree represents the initial position. During the sampling period, the robot visits only the children nodes and then the planning decision is stored in the parent node. Thus, each parent node contains the information of the local best child node.

Algorithm 1 HexTree formulation

Require: $\mathcal{A}$, x_0
$Tree$
$Tree.AddRoot(x_0)$
$v_t \leftarrow x_0$
while not converged to hotspot **do**
    $q_{hex} \leftarrow HexagonalSampling(\mathcal{A}, v_t)$
    $q_{sort} \leftarrow GetSamplePath(q_{hex})$
    $q_{best} \leftarrow FindMaxInfoEdgeOnTree(Tree, q_{sort})$
    $v_{t+1} \leftarrow GetNewParent(\mathcal{A}, Tree, q_{hex}, q_{best})$
    for all $i \in \mathbf{n}_t$ **do**
        if $z^i_t > z_h$ **then**
            return $OptimalReturnPath\,(tree)$
        end if
    end for
end while

B. *Sampling path generation*

The tree structure is now under a favorable condition to reduce the UAV exploration using radiation field properties.

We can create an incremental version of Hamiltonian path to visit each child location, $x_t^i \in q_{hex}$, at most one time and avoid redundant visits to the same location. Instead of n number of random points, the HexTree constraints sample points to six at sampling time t, and that some of them are shared. In order to generate a Hamiltonian path to the current UAV location, first, we assign a global index set J with the index function ind. ind generates a unique index for each node mapping the node from the position domain $x_t^i \in \mathbb{R}$ into the index domain $ind(x_t^i) \in \mathbb{N}$. We store each visited node into $J = \cup \left\{ind(x_t^i)\right\}$. Secondly, we compute the traveling cost between the UAV and each new node position given by

$$q_t^i.cost = dist(x_t^i, x_0)_{ind(x_t^i) \notin J} \tag{9}$$

Once we compute the cost of each node, we sort all the candidate nodes using the Hamiltonian path algorithm [22]. Thus, we find the optimal sequence to sample the local hexagonal grid. Note that, even though we sample the area using a regular hexagonal grid, the sample path does not need to follow such a pattern. At each step, the UAV exploration is ended with a candidate child location and the subsequent sample path is generated with the evolved location.

Algorithm 2 Sampling path generation

```
Require: q_hex
  for ∀x_t^i ∈ q_hex do
      if ind(x_t^i) ∉ J then
          q_t^i.cost ← dist(x_t^i, x_0)
      else
          q_t^i.cost ← ∞
      end if
  end for
  q_sort ← HamiltonianPath(q_hex)
  return q_sort
```

C. The next best parent

Given the sample path defined by Alg. 2, the robot travels to each of the node location at most one time similar to Fig. 2 and gathers the measurement attributes. Based on the gathered information, the next best parent location is computed using the following steps- firstly, we assign the virtual edges for each sample location deterministically, for instance, the virtual edges $\{BF, EF, EA, ...\}$ in Fig. 2 represent the nodes $\{B, F, E, ..\}$. Secondly, we compute the information gain for each node using Eqn. 6. Thirdly, we find the neighbor edges around the edge of maximum information gain. For instance, the edges HB and EF are the neighbor edges of BF in Fig. 2. Finally, we find the intersection point by extending those edges, i.e., G in Fig. 2. We call it the next best parent location. Note that G is a virtual point which the robot does not visit. It is only computed to generate the next sample locations in a hexagonal manner.

D. The optimal return path

While the UAV iteratively seeks the hotspot location, all the sampled locations are not likely to be informative due to the fixed sample interval. We therefore generate an optimal return path using the dynamic programming based Knapsack algorithm [23]. Specifically, we constrain the travel distance while maximizing the information gain. Since the HexTree already tracked all the information through the sampling step, we convert it to the Knapsack variables. The return path abandons those locations where the measurement does not add much important information to the field. The local field was represented by the child node having the maximum information gain, which was also stored in the virtual parent node. Neglecting the distance between such a child node and its parent node, we can easily generate the return path based on the parent node locations only.

V. Optimality Analysis

Our approach is to tile the hexagon through the sample nodes using the area coverage strategy. Since the tiling process involves node sharing, the redundant visit can be avoided by generating the Hamiltonian path. However, when the robot has to avoid the shared nodes, there can be an increase in the path length. We will now give an upper bound on this path length.

Definition 1: Let's assume that 'a' is the step length which is required to travel to two consecutive children nodes in a parent grid and 'n' is the total number of parents in the tree. An online algorithm solving the navigation problem is said to be optimal if the total path length is bounded by a factor $\left(5 + \sqrt{3}\right) na$

We first state a number of modest assumptions that are required for this analysis. Next, we place a bound on the sampled path by finding the proofs of two essential theorems.

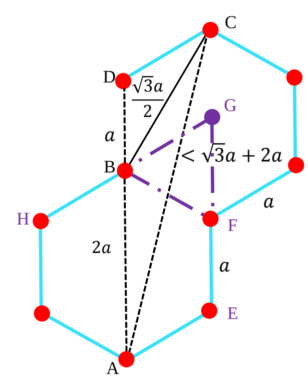

Fig. 2: **HexTree Sampling Pattern:** the red dots are the sampled locations, the cyan lines are the sampled path. The black lines are imaginary lines drawn for analysis and the purple lines are drawn to determine the next parent (virtual) location, denoted by purple dot.

Assumption 5.1: There exists at least one new child node in each parent node to extend the tree towards the goal position.

This assumption requires that some free space is available to create a sample parent. Since two consecutive sample parents share at least one edge, to make a valid sample patent at least one new vertex is required.

Assumption 5.2: Starting with an arbitrary location, the robot travels to each child node of a sample parent in a Hamiltonian path manner.

We want to optimize the total length of exploration time. Thus, this assumption leads the navigation problem to an optimal solution. Since the adjacent parents are sharing at least one edge- which means at least two children nodes, traveling to all the children at most one time is helpful to avoid the redundant visits.

Theorem 5.1: If 'n' number of hexagons are required to reach the goal position, there exists at least $5n$ edges which are required to generate a Hamiltonian path

Proof: Given a hexagonal grid, initially, the robot can travel to at most $6 - 1 = 5$ vertices to make a Hamiltonian path. If this pattern follows, then it is straightforward to conclude that $5n$ edges are required in total. However, since two adjacent hexagonal grids share at least one edge, the robot skips 2 vertices while travailing to all the children nodes. From Fig. 2, we can understand that the number of edges which is required to extend the Hamiltonian path for the adjacent grid is $6 - 2 = 4$. However, since the robot needs to travel at least distance a to reach the adjacent grid, which is equivalent to the length of an edge, we can then compute the total edges as follow

$$4n + n = 5n. \tag{10}$$

■

Theorem 5.2: If 'n' number of parents are required to extend the tree onto the goal position and 'a' is the distance between the consecutive neighbors, the sampled path for the HexTree can be extended to $a\sqrt{3}(n-1)$.

Proof: In the worst case, the robot travels to the farther child node of the neighbor parent and the neighbor parent shares only two common children. Fig. 2 shows the worst case scenario, where the robot travels to the farthest child node. Let's assume that the robot is located at node A and the goal node is denoted as B in Fig. 2. In order to compute the worst case path distance, AC, we draw two imaginary edges from A to B and B to C. Since a regular hexagon comprises of six equilateral triangles, the length of BC can be computed from the triangle BDC as follows

$$BC = 2\sqrt{(a^2 - a^2/4)} = a\sqrt{3} \tag{11}$$

And it is obvious from Fig. 2 that the length of AB is the diameter of the hexagon, computed by

$$AB = 2a. \tag{12}$$

Thus, using the triangle inequality theorem for the triangle ABC, we can state that $AB + BC > AC$. As a result, we can compute the upper bound of AC from Eqn. 12 and Eqn. 11 as follows

$$\therefore AC < (a\sqrt{3} + 2a). \tag{13}$$

However, as we know from the definition that the minimum travel distance between two consecutive children nodes is a and in the worst case scenario two consecutive parents share at least one edge whose length is also a, thus, the maximum length of AC can be exceeded to

$$\Delta AC = a\sqrt{3} + 2a - a - a = a\sqrt{3}. \tag{14}$$

As we can see from Fig. 2, we need at least two hexagons to compute the length of AC, therefore, the worst case path could be extended by the n number of hexagons as follows

$$\sum\nolimits_{n-1} \Delta AC = (n-1)\Delta AC = a\sqrt{3}(n-1). \tag{15}$$

■

Combining the theorems and the above equations, we can compute the final bound of the sampled path as follows

$$\sum\nolimits_{n-1} \Delta AC + 5na < a\sqrt{3}(n-1) + 5na \approx (5 + \sqrt{3})na. \tag{16}$$

TABLE I: Overview of parameters

Parameter	Value	Description
$\mathcal{A}$	$30m \times 30m$	Target area size
$\mathcal{H}_{str}$	3000	Strength of hotspot
$\mathcal{H}_{pos}$	[25 20]	Position of hotspot
σ	[150 150]	Spreading matrix
a	$1m$	Sample step length
ξ	2900	Termination threshold

TABLE II: Algorithm performance

	RIG Tree	HexTree
# iterations	55	7
# samples	55	42
path len.(m)	24.849	7.865
time (s)	104.364	43.256
std. dist. (m)	10.416	0
Max path len. (km)	∞	2.33

VI. Simulation Results

We present results of numerical experiments with the HexTree planner compared to the RIG-tree planner. Our return path is also compared to the path smoothing algorithm. Table I presents an overview of parameters used for the experiments. Each algorithm is implemented in MATLAB on a PC with a $3.40GHz$ Intel(R) Core(TM) i7 processor and $8.0GB$ RAM.

A. Compared Strategies

The RIG-tree planner is in principle capable of finding the hotspot, which is asymptotically optimal. However, it takes a very long time to converge to the hotspot. For a single UAV exploring over a large area, the HexTree planner is even faster than the RIG-tree planner. To compare the performance of both planners, we measure the total length of sampled path and the time to converge with the same initial position, step size, and termination condition.

Fig. 3 shows the difference between the HexTree and RIGtree planners. The HexTree sampled path was restricted by the hexagons, and only 7 iterations were required to converge to the hotspot, while 55 iterations were required for the RIG-tree. Furthermore, since the HexTree sampled

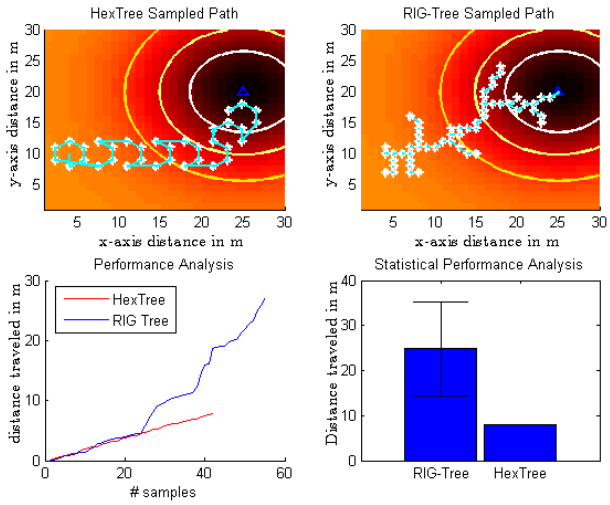

Fig. 3: **Performance comparison:** White dots are the sample locations, cyan lines are the sampled path, the blue triangle is the hotspot position, the baseline colored map is the radiation distribution. A UAV needs to sequentially explore each location to gather the information. The RIG tree does not have sequential path generation characteristic, which is why the HexTree always outperforms it.

path was sequentially optimized, we can see from Table II that the HexTree traversed only $7.865m$ distance to find the hotspot, whereas the RIG-tree traversed $24.849m$. Since the RIGtree randomly sampled the area, it required more sample points (55) and longer time $(104.364s)$. On the other hand, the HexTree clearly outperformed the RIG-tree in terms of the sample point (42) and the convergence time $(43.256s)$. The total distance required to reach the hotspot with the RIG-tree varied significantly across 10 experiments, while the HexTree remained constant. In case of the absence of hotspot, the RIG-tree could not find any condition for which sampling will be terminated, resulting in a sampled path of infinite length. On the other hand, the HexTree needed only 346 iterations to completely cover the $900m^2$ area, resulting in $2.33km$ sampled path to reach a termination point

B. Path Smoothing Vs. Optimal Return Path

The sampled path may contain locations that lead to redundant visits in view of information gathering. Optimizing the return path is therefore an essential post-processing step similar to the RRT [17] path smoothing to avoid zig-zag paths. Our focus is to find the most informative locations considering the travel cost. As the RIG-tree does not have the return path, we compare our return path with the RRT path smoothing. We have performed 3 simulations with different initial UAV positions and hotspot locations as shown in Fig. 4. Since the path smoothing finds the shortest path to the initial position, the return path for the RIG-tree looks more like a straight line and cannot find any informative locations from the sampled path. Meanwhile, the proposed path finds several locations considered as the most informative locations satisfying the travel distance constraint.

VII. Acknowledgment

This work was supported by the Industrial Convergence Core Technology Development Program (No. 10063172) funded by MOTIE, Korea.

VIII. Conclusion

We have proposed an online HexTree path planner for UAV navigation to efficiently localize the radiation hotspot. The sampled path generated by the HexTree planner was optimal and ensured termination of exploration. After localizing the hotspot, a loop closing trajectory was generated from sampled trajectories, which contained the most informative locations considering the travel cost. Through our theoretical analysis, we have proven the optimality of the HexTree path and found an upper bound of path length. Since the UAV explored a large area for spatial sampling, the HexTree planner clearly outperformed the RIG-tree planner in terms of the distance traveled and the convergence time. Future work will be devoted to extending the HexTree planner to localize multiple hotspots, and demonstrating the efficiency of the proposed approach through real world experiments.

References

[1] S.-i. Azuma, M. S. Sakar, and G. J. Pappas, “Stochastic source seeking by mobile robots,” *Automatic Control, IEEE Transactions on, 2308-2321*, 2012.

[2] N. Ghods, P. Frihauf, and M. Krstic, “Multi-agent deployment in the plane using stochastic extremum seeking,” in *Decision and Control, IEEE conf. on, 5505-5510*, 2010.

[3] N. A. Atanasov, J. Le Ny, and G. J. Pappas, “Distributed algorithms for stochastic source seeking with mobile robot networks,” *ASME Journal of Dynamic Systems, Measurement, and Control*, 2015.

[4] G. A. Hollinger and G. S. Sukhatme, “Sampling-based robotic information gathering algorithms,” *The Int’l Journal of Robotics Research*, 2014.

[5] K. H. Low, J. M. Dolan, and P. Khosla, “Adaptive multi-robot wide-area exploration and mapping,” in *Int’l conf. on Autonomous agents and multiagent systems, 23-30*, 2008.

[6] K. H. Low, J. M. Dolan, and P. K. Khosla, “Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing.” in *Intl Conf. Automated Planning and Scheduling*, 2009.

[7] J. Towler, B. Krawiec, and K. Kochersberger, “Radiation mapping in post-disaster environments using an autonomous helicopter,” *Remote Sensing, 1995-2015*, 2012.

[8] M. R. Morelande and A. Skvortsov, “Radiation field estimation using a gaussian mixture,” in *Information Fusion, Int’l conf. on, 2247-2254*, 2009.

[9] J.-C. Chin, D. K. Yau, and N. S. Rao, “Efficient and robust localization of multiple radiation sources in complex environments,” in *Distributed Computing Systems, IEEE Int’l conf. on, 780-789*, 2011.

[10] E.-W. Bai, K. Yosief, S. Dasgupta, and R. Mudumbai, “The maximum likelihood estimate for radiation source localization: Initializing an iterative search,” in *Decision and Control, IEEE conf. on, 277-282*, 2014.

[11] B. Charrow, V. Kumar, and N. Michael, “Approximate representations for multi-robot control policies that maximize mutual information,” in *Robotics: Science and Systems*, 2013.

[12] G. Hitz, A. Gotovos, F. Pomerleau, M.-E. Garneau, C. Pradalier, A. Krause, and R. Y. Siegwart, “Fully autonomous focused exploration for robotic environmental monitoring,” in *Robotics and Automation, IEEE Int’l conf. on, 2658-2664*, 2014.

[13] P. Dames, M. Schwager, V. Kumar, and D. Rus, “A decentralized control policy for adaptive information gathering in hazardous environments,” in *Decision and Control, IEEE conf. on, 2807-2813*, 2012.

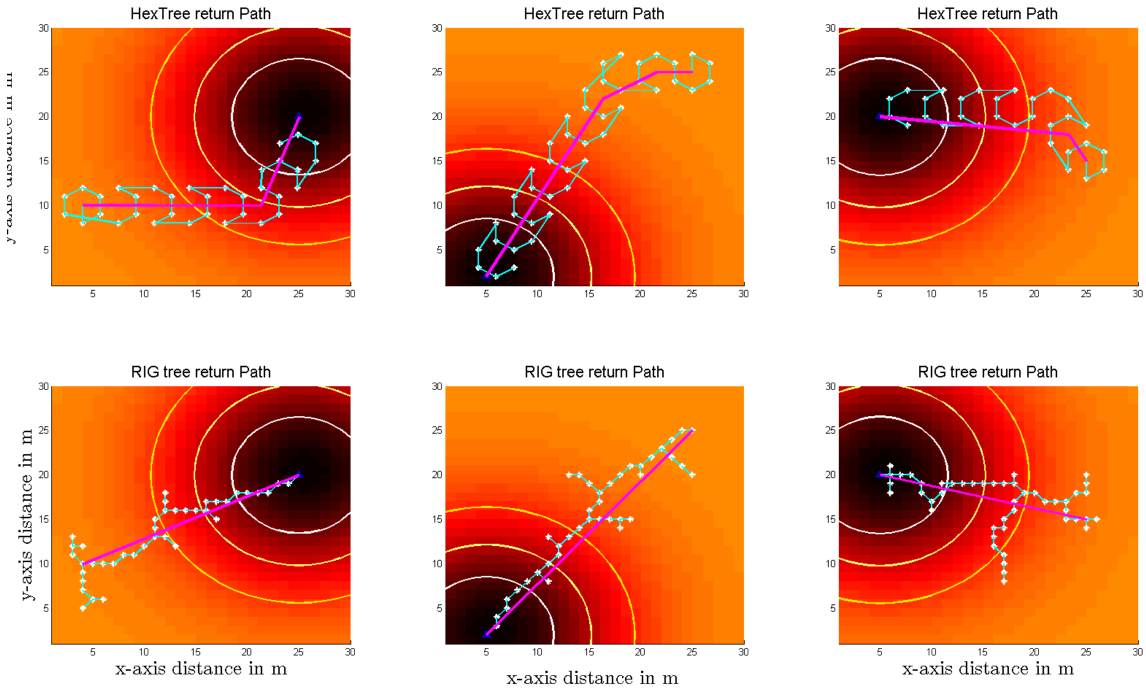

Fig. 4: **Return path generation:** HexTree return path is generated while visiting the most informative locations. On the other hand, the path smoothing algorithm finds the shortest path to the initial location subject to distance constraints.

[14] R. Cortez and H. Tanner, "Radiation mapping using multiple robots," in *ANS Joint Topical Meeting on Emergency Preparedness and Response and Robotic and Remote Systems, 157-159*, 2008.

[15] A. A. R. Newaz, S. Jeong, H. Lee, H. Ryu, N. Y. Chong, and M. T. Mason, "Fast radiation mapping and multiple source localization using topographic contour map and incremental density estimation," in *IEEE International Conference on Robotics and Automation, 1515-1521*, 2016.

[16] J. Han and Y. Chen, "Multiple uav formations for cooperative source seeking and contour mapping of a radiative signal field," *Jour. Intelligent & Robotic Systems, 323-332*, 2014.

[17] I. Ko, B. Kim, and F. C. Park, "Randomized path planning on vector fields," *The Int'l Journal of Robotics Research, 1664-1682*, 2014.

[18] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *AAAI Conf. Artificial Intelligence, 1650-1654*, 2007.

[19] N. Cao, K. H. Low, and J. M. Dolan, "Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms," in *Int'l Conf. on Autonomous agents and multi-agent systems,7-14*, 2013.

[20] Y.-H. Kim and D. Shell, "Distributed robotic sampling of non-homogeneous spatio-temporal fields via recursive geometric sub-division," in *Robotics and Automation, Int'l conf. on, 557-562*, 2014.

[21] T. C. Hales, "The honeycomb conjecture," *Discrete & Computational Geometry, 1-22*, 2001.

[22] J. Clark and D. A. Holton, *A first look at graph theory*. World Scientific, 1991.

[23] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.