IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 20, 2021, accepted January 20, 2022, date of publication January 31, 2022, date of current version February 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3147720

Utilizing a Rapidly Exploring Random Tree for Hazardous Gas Exploration in a Large Unknown Area

YAQUB A. PRABOWO[1,2], BAMBANG R. TRILAKSONO[1,2], (Member, IEEE), EGI M. I. HIDAYAT[1,2], AND BRIAN YULIARTO[1]

[1]Bandung Institute of Technology, Bandung 40132, Indonesia
[2]University Center of Excellence—Artificial Intelligence on Vision, NLP and Big Data Analytics (U-CoE AI-VLB), Institut Teknologi Bandung, Bandung 40132, Indonesia

Corresponding author: Bambang R. Trilaksono (briyanto1@gmail.com)

This work was supported in part by the Indonesian Ministry of Research and Technology/National Agency for Research and Innovation and Indonesian Ministry of Education and Culture under World Class University (WCU) Program Managed by the Institut Teknologi Bandung.

ABSTRACT The use of robotics olfaction for gas source localization or mapping has become a concern given the issues of terrorism or industrial accidents that may cause damage to the environment. A typical scenario is to send a robot to a place where a dangerous gas leak has just occurred. The robot's task is to map gas concentrations in the region of interest as effectively as possible. This paper addresses how the robot performs gas exploration in a large and unknown environment. One of the issues that needs to be addressed is the fact that the computation time of the path planning, frontier detection, goal decision making and gas distribution mapping is slower if all cells in the occupancy grid map are involved in a large environment. Consequently, the Rapidly-exploring Random Tree (RRT) algorithm is chosen as the main algorithm. The RRT graph guides the robot's navigation, utilizes the vertices as goal candidates, gas mean and variance value, and searches for a new frontier. A new strategy is proposed to address the frontier exploration and gas exploitation trade-off. Finally, a Robot Operating System (ROS), Gazebo, and a 3D gas simulator are used to compare the proposed strategy performance with the others in a large outdoor environment.

INDEX TERMS Robot olfaction, robot exploration, rapidly-exploring random trees.

I. INTRODUCTION

In this modern era, robotics systems are very popular for use in search and rescue missions, which are very dangerous if they are done directly by humans [1]. A robot is expected to explore the affected area to obtain as much information as possible to support the evacuation process. A typical search and rescue mission is when a hazardous gas contaminant is dispersed, whether in an industrial area [2] or because of a natural disaster [3]. From a comprehensive review in [4], it can be concluded that some important things that must be considered in the use of robots in related missions are battery efficiency, exploration of coverage areas, and exploitation in a region of interest. The complexity of computing is also a problem, especially for missions in an extensive area. The use of multiple robots can overcome this scalability problem [5], [6]; however, using multiple robots is expensive.

The associate editor coordinating the review of this manuscript and approving it for publication was Wai-Keung Fung.

In terms of gas distribution mapping, [7] developed a method to drive a robot using an artificial potential field method, while [8] used a particle filter algorithm to perform gas source localization. However, the problem was not complex, as the environment was assumed to be known and free from obstacles. Recent research by [9] used optimal policies to perform gas distribution mapping in a cluttered environment, but the robot previously knew the occupancy map. Another researcher [10] developed an integration between Simultaneous Localization And Mapping (SLAM) [11] and gas distribution mapping in an unknown area, but the robot was controlled remotely. By aggregating all the problems above, we address the development of a fully autonomous robot to explore and exploit hazardous gas contamination in a cluttered and unknown large area.

Exploration in a wide area requires a scalable algorithm. Using all cells in a vast grid map for the path and goal candidates will increase the processing load. This means that all of the cells should be involved in the computation.

Therefore, graph-based representation is chosen because it is lighter and computationally tractable. However, graph representation is less optimal in generating the trajectories than grid representation, but at least it suits motion planning problems, as mentioned in [12]. The graph is generated over the obstacle-free area, which is utilized as the robot's path. This approach then takes advantage of the vertices as the goal candidates of the robot. Inspired by [13], a Rapidly-exploring Random Tree (RRT) is chosen as the algorithm that can rapidly generate a tree graph. However, although there are fast non-grid-based path planning based on evolutionary algorithms exist such as [14], [15], we still choose RRT as it has multi-functions. RRT can be exploited not only for path planning, but also for frontier search, gas distribution representation and goal candidates which will be explained in detail in the next Section. Moreover, in an exploration mission, we need an explorative path as the RRT algorithm generated.

There is a gas measurement problem in search and rescue missions in an area contaminated with hazardous gases. Aside from the gas sensor noise, which is quite large, the time variant of gas propagation also makes it difficult to measure because it is influenced by wind flow, gas diffusion, or object movement. Therefore, the value of the gas variance is significant in determining the certainty of a gas concentration value in a particular place. Some researchers have developed several gas map extrapolation methods that produce variance values, including kernel gas distribution mapping plus Variance (Kernel DM + V) [16], [17], a Gaussian Markov random field [18], or a Gaussian process [19]. From these methods, Kernel DM + V is the method that has the lowest computational complexity. For instance, [7] used a Gaussian Markov random field in a large cluttered environment but needed simulation with a 10000 second duration because of the big time complexity of the model. In this case, the robot needs to get the update of the gas distribution model in real-time. To get the update faster, we do a modification by approximating the grid-based Kernel DM + V into a graph-based Kernel DM + V although the graph representation will be less accurate. In addition to being faster, graph-based Kernel DM + V is also more suitable for application in nonconvex environments as it can utilize any type of graph that suits in nonconvex areas such as RRT graph.

In an unknown environment, it is necessary to perform online obstacle mapping. The occupancy map formed by 2D LIDAR contains several frontiers, which are the border areas between the free map and the unknown map. As developed by [20], the frontiers can be searched rapidly using the RRT algorithm. We utilize it even further by using the graph to construct a robot path as well. Every vertex is considered a goal candidate. The proposed graph-based Kernel DM + V can also utilize the graph. Each vertex has attributes of the gas concentration mean and variance.

In this paper, an Unmanned Aerial Vehicle (UAV)-type robot is used for evaluation of the method. The UAV is installed with a gas sensor and 2D LIDAR. It is sent to an unknown area by the operator. Mapping the whole area of a large area takes a long time. The UAV only has a short time operation. Therefore, by using our proposed strategy, it is efficient if the robot only explores and exploits the gas in the hazardous area. Our proposed strategy has a switching mechanism between ''frontier exploration'' and ''gas exploitation'' which will make a robot only covers the hazardous area.

Initially, the robot builds the occupancy grid map from the beginning and then generates the graph to find the frontier. The first state active is ''frontier exploration''. This indicates a set of frontiers as candidates for the robot's goal. The optimal frontier point is the point that has the maximum information gain according to the occupancy map and the distance between the frontier and the robot.

If there is at least a vertex with a high gas concentration mean and variance detected/estimated by Kernel DM+V, the robot will enter the ''gas exploitation'' state. In this state, the robot goes to the vertex that has the following two conditions: highest variance and concentration mean value above the predetermined threshold. Thus, the robot will visit an area with the most uncertain and high expected gas contamination. More gas samples measured in one area will decrease the variance of gas measurement in that area. Practically, in the recovery and mitigation process after hazardous gas has leaked, an area containing a high gas concentration is more dangerous than an area containing a low gas concentration. Moreover, gas contamination sources with high probability might be located in areas with high gas concentrations.

As long as a vertex with a high gas concentration mean and variance exists, the state remains at ''gas exploitation''. In this state, the robot visits the vertices around the area contaminated with high gas concentration. It will stay in this state and collect measurements in the aforementioned area. With an increasing number of measurements, the variance of the gas concentration distribution in that area decreases, and then the robot will switch back to the ''frontier exploration'' state. It will explore different areas of the map and try to find other high gas concentration measurements.

Several simulations with various scenarios are conducted to evaluate the proposed strategy toward exploring and exploiting hazardous gas in an unknown wide area. The simulations are performed in a Robot Operating System (ROS) and Gazebo platform exploiting the use of a 3D gas dispersion simulator [21]. To the best of our knowledge, there is no past research about gas exploration in an unknown environment with some obstacles in which the robot is fully autonomous. However, the evaluation of the proposed method is also compared with some methods based on the scalar objective function and Artificial Potential Field (APF) from [7], although, in the original paper, APF was implemented in a free obstacle environment. In particular, APF strategy uses three objectives directing the robot towards areas. The objectives are: (1) high estimated mean, (2) high estimated variance, while maximizing the coverage area (3). The first and second objectives are achieved by visiting some areas with high estimated mean and variance. The third objective is implemented by a

repulsive potential generated by placing charges at all prior gas measurements.

The mission can also be conducted using the manual teleoperated method as in reference [10]. However, the manual teleoperated method has a disadvantage in that it requires very good communication between the robot and the station so that the command can be given in real-time, although this incurs a considerable cost in reality. Moreover, it requires a human operator to control the robot. He or she must be far away from the location because of the danger of gas contamination, while the method proposed in the present paper does not require a human operator. In other words, the robot can operate autonomously.

The main problem addressed in this paper is how to autonomously drive a mobile robot in a large, non-convex and unknown environment to explore the gas contamination in a relatively short time. To deal with this problem, the robot needs to do path planning that is adapted to such a complex environment. As no prior information about the map is available, the robot should partially extend the map into new territory using a frontier exploration algorithm. An online decision-making mechanism towards the gas exploration should be implemented so that the gas distribution model can be computed in real-time. Moreover, as the robot has a relatively short time operation, it should exploit the gas only in the hazardous area once the robot finds a gas contamination.

Those problems will be solved considering some assumptions as follows. The experiments are conducted in a computer simulation using the Robot Operating System platform. The expected area of the environment is 500m x 700m without a dynamic obstacle. There is only one stationary gas source. The UAV has a precise localization using IMU and RTK-GPS which is combined with an Extended Kalman Filter. A 2D LIDAR with low uncertainty is installed in the simulation so that the occupancy grid map can be accurately obtained. The occupancy grid map is limited in two dimensions so that the path and goal candidates are generated on a 2D plane. In other words, the UAV flies at a static altitude. Jetson TX2 with 8GB RAM and Quad-Core ARM Cortex A57 CPU is used as the processing board in the simulation.

The contributions of this paper are an extension of Kernel DM+V and an autonomous strategy toward gas exploration-exploitation in a large-scale area. The extension of Kernel DM+V to graph representation is developed because it is more scalable and suitable to implement in a nonconvex area. The goal-oriented decision-making strategy in hazardous gas exploration-exploitation is developed by exploiting the RRT algorithm as the goal candidates, frontier search and path planning support.

The paper is organized as follows. Section II elaborates the methods proposed to solve the specifics of the related problems. Section III discusses the results of conducted simulations. Section IV concludes the paper and discusses several future works.

II. METHODOLOGY

In this section, some methods are elaborated according to the problem addressed by this paper. The RRT graph is grown on a partially expanded occupancy grid map. The graph-based Kernel DM+V is used to acquire the mean and the variance value of the gas concentration by using the generated graph. Some frontiers are also detected by the graph, while the optimal goal location is determined by a finite state machine that considers the gas concentration mean and variance according to the graph-based Kernel DM+V. The operator may choose the robot's priority for either exploration over exploitation or vice versa. The robot performs exploration to gather more gas information in an unexplored area. It does exploitation to investigate the most interesting region with a high gas concentration.

A. GENERATING THE RRT GRAPH

In this paper, the very basic RRT graph is used since it has the fastest computation. In our case, RRT is not only used for path planning but also used for frontier detection. The frontier should be detected as fast as possible. That is why we choose the fastest RRT. However, the basic RRT path is less optimal than RRT*, informed RRT, etc. Therefore, the guidance mechanism is modified to not follow the original graph. This mechanism is usually called path pruning. The robot can go straight to the farthest vertex, that is, Line of Sight (LoS) with the robot. This technique will be elaborated in subsection II-B.

The robot generates the graph from the beginning on the occupancy grid map, initially opened by the robot's 2D laser range finder. Let $G = \{V, E\}$ be a graph with a set of vertices (V) and edges (E) that are generated over the obstacle-free map. Each vertex will be used for goal candidates. Each vertex also has mean and variance gas concentration attributes which will be explained in Section II.D. Each edge has length attributes used to calculate the approximation distance between two vertices. This graph is used for robot path planning and frontier detection which will be explained in Section II.B and II.C respectively.

A 2D LIDAR with a point cloud form is converted into a grid map by this technique. If a cell contains a laser point, then it will be considered as an occupied cell. Some cells that are passed by the laser line, then it will be considered as free cells. Otherwise, they are unknown cells. The occupancy grid map is denoted as a matrix $occ(x)$. A grid cell x is free if $occ(x)$ is equal to 0. A grid cell x is unknown if $occ(x)$ is equal to -1. Otherwise, the value of $occ(x)$ is between 1 and 100 which x is considered as an occupied cell.

Two RRT graphs, global and local RRT graphs with different functions, are generated. Global RRT is used for robot path planning (Section II.B). At the same time, both global and local RRT are used for frontier detection (Section II.C). These are generated based on the very basic RRT algorithm from [22]. The graph is generated from the initial robot position and continually expanded, taking a random point

inside the area where the robot should explore. In short, a new vertex and edge are appended to the existing graph as long as they are free from the obstacle.

Due to the LIDAR noise and inaccurate grid interpolation, it is necessary to check whether a graph element coincides with an obstacle or not. There is a very rare condition where a cell is considered as a free cell but the truth is occupied by an obstacle. Therefore, a module is created to check whether there is an edge that coincides with the obstacle. If this occurs, then the vertex or edge and its children have to be removed, as it is dangerous for the robot if it is chosen as a goal. Notably, if there is a graph element that coincides with the obstacle, the robot should hold the position because removing some children takes time. After updating the graph, the robot can continue to navigate again to its current goal.

Algorithm 1 Algorithm for Generating Global RRT Graph

Input: $\eta_{\max}$, $\eta_{\min}$, occ, l, w
Output: G, new frontier
// initialize with empty graph
$V_{local} \longleftarrow \{x_{robot}\}$;
$E \longleftarrow \emptyset$;
$G_{local} \longleftarrow (V_{local}, E_{local})$;
while true do
  $x_{rand} \longleftarrow$ UniformRandomSample(l, w);
  $x_{\text{nearest}} \longleftarrow$ Nearest$(G = (V, E), x_{\text{rand}}, \eta_{\max})$;
  $x_{new} \longleftarrow$ Steer$(x_{\text{nearest}}, x_{\text{rand}})$;
  if $(\|x_{\text{new}} - x_{\text{nearest}}\| > \eta_{\min})$ **and** ObsFree$(x_{\text{new}}, x_{\text{nearest}})$ **then**
    if occ$(x_{\text{new}}) = 0$ **then**
      // cell is free
      $V \longleftarrow V \cup \{x_{\text{new}}\}$; *// add a new vertex*
      $E \longleftarrow V \cup \{(x_{\text{new}}, x_{\text{nearest}})\}$; *// add a new edge*
      $G \longleftarrow (V, E)$; *// assign updated graph element*
    else if occ$(x_{\text{new}}) = -1$ **then**
      // cell is unknown
      PublishNewFrontier(x_{new});
    end if
  end if
  for $(x_i, x_j) \in E$ **do**
    if not ObsFree$((x_i, x_j))$ **then**
      HoldRobotPosition();
      // remove children belong to vertex (x_i, x_j)
      $E \longleftarrow E \backslash$Children$((x_i, x_j))$;
      $V \longleftarrow V \backslash$Children$(x_i)$;
      $G \longleftarrow (V, E)$;
    end if
  end for
  PublishGlobalRRTGraph(G);
  ContinueRobotNavigation();
end while

The pseudocodes in Algorithms 1 and 2 explain how the global and local RRT graphs are generated. The global RRT graph is continually expanded as long as the robot opens a new free occupancy grid map. The global RRT graph is used to plan the robot's path, but it is also used to search for a new frontier. Two constants η_{min} and η_{max} are the lowest and highest limits of the edge length, which are helpful to control the number of graph elements. Using too many graph elements will slow the computation, and some short edges do not significantly increase the coverage performance.

Algorithm 2 Algorithm for Generating Local RRT Graph

Input: $\eta_{\max}$, occ, l, w, x_{robot}, F, G, t_{out}
Output: new frontier
while true do
  // initialize with empty graph:
  $V_{local} \longleftarrow \{x_{robot}\}$;
  $E \longleftarrow \emptyset$;
  $G_{local} \longleftarrow (V_{local}, E_{local})$;
  newFrontierFound $\longleftarrow$ **false**;
  while true do
    $x_{rand} \longleftarrow$ UniformRandomSample(l, w);
    $x_{\text{nearest}} \longleftarrow$ Nearest$(G_{local} = (V_{local}, E_{local}), x_{\text{rand}}, \eta_{\max})$;
    $x_{new} \longleftarrow$ Steer$(x_{\text{nearest}}, x_{\text{rand}})$;
    if ObsFree$(x_{\text{new}}, x_{\text{nearest}})$ **then**
      if occ$(x_{\text{new}}) = 0$ **then**
        // cell is free
        $V_{local} \longleftarrow V_{local} \cup \{x_{\text{new}}\}$; *// add a new vertex*
        $E_{local} \longleftarrow V_{local} \cup \{(x_{\text{new}}, x_{\text{nearest}})\}$; *// add a new edge*
        $G_{local} \longleftarrow (V_{local}, E_{local})$; *// assign updated graph elements*
      else if occ$(x_{\text{new}}) = -1$ **then**
        // cell is unknown
        PublishNewFrontier(x_{new});
        RobotNavigatesTo(x_{new});
        Break; *// because a new frontier is found*
      end if
    end if
    if Timeout(t_{out}) **then**
      // no new frontier
      // Run Algorithm 3
      $x_{goal} \longleftarrow$SelectFrontier(F, G, t_{out}, x_{robot})
      RobotNavigatesTo(x_{goal});
      Break;
    end if
  end while
end while

The local RRT graph is only used for searching a new frontier near the robot. This graph is very important because when the robot opens a new free area, it should quickly find the frontier. Then, the robot can go straight to the frontier near the robot rather than far to one of the past frontiers. The l and w constants are the length and width of the rectangle where a random point (x_{rand}) is obtained, with the center of the rectangle being the robot position.

There are some conditions when the robot cannot find a frontier because the obstacle in the region of that rectangle boundary is fully mapped, or it may be difficult to search the frontier in a narrow area. To solve this issue, a timeout is used to stop the local frontier search. There is no new frontier if the timeout is reached. Therefore, the latest set of frontiers is used as the candidate where the robot goal is. There is no minimum edge length for the local RRT graph. It only needs a parameter called $\eta_{\max}$ because it should search the frontier as quickly and flexibly as possible.

B. NAVIGATING THE ROBOT

Fig. 1 illustrates how the robot starts generating the RRT graphs and then goes to a particular point, which is the frontier, since the robot has not yet sensed any gas concentration more than zero.

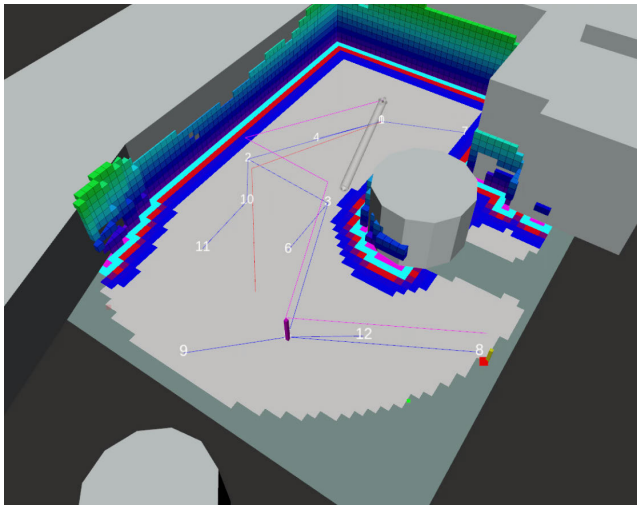

FIGURE 1. A robot initially starts and opens the grid map. The blue graph with white text numbers in each vertex and the red graph are the global and local RRT, respectively. The purple graph is the RRT subgraph connecting the current robot pose to the goal. The green dots are the frontier candidates. The yellow bar is the robot's final goal, while the purple bar is the robot's current goal since it is the nearest vertex to the final goal and is LoS.

How the robot navigates by utilizing the global RRT graph is shown in Fig. 2. First, the robot finds the vertex nearest to the current robot position and the vertex nearest to the current goal. Then, a subgraph connecting those two vertices is formed. As shown in Fig. 2 left, if the robot applies a straight movement directly to the goal, it will obstruct the wall. The robot should find the vertex nearest to the goal, which is LoS with the robot.

Ideally, the robot can go to that vertex, but if there is any disturbance, such as wind or controller anomalies, then the robot may not track that straight line. For example, this is shown in Fig. 2 right by the green arrow. If this occurs, the robot must change the current goal nearest to the ultimate goal and LoS. Therefore, the robot should check periodically whether the current goal is still LoS or not. In this paper, a PID

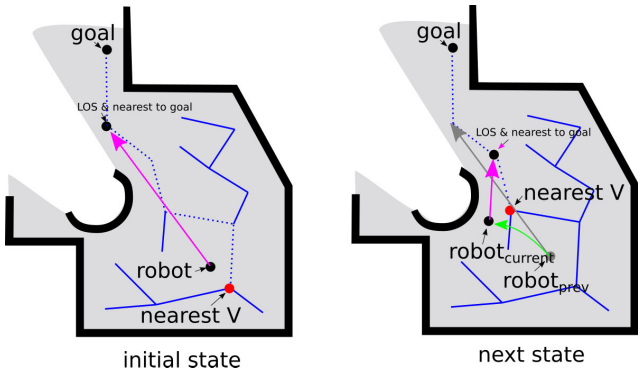

FIGURE 2. A figure shows how a robot navigates by utilizing the global RRT graph. The blue lines are the RRT graph vertices, with the dashed lines being the subgraph as the guidance for the robot toward its goal. The magenta arrow indicates an LoS line starting from the robot position to the nearest LoS vertex to the goal. The green path is the robot path, and the gray arrow is the past LoS line.

Algorithm 3 Algorithm for Selecting Frontier

Input: F, G, t_f, x_{robot}
Output: x_{goal}

$x_{\text{goal}} \longleftarrow \text{Nearest}(F, x_{\text{robot}})$
while true do
  $F \longleftarrow$ getTheLatestFrontiers();
  NavigateTheRobot(x_{goal});
  if $\|x_{\text{goal}} - x_{\text{robot}}\| > \epsilon_d$ **then**
    Continue; *// because the robot has not arrived*
  else
    nearFrontierFound$\longleftarrow$**false**;
    while not nearFrontierFound **and not** Timeout(t_f) **do**
      $F \longleftarrow$ getTheLatestFrontiers();
      for $f_i \in F$ **do**
        $cost \longleftarrow \text{Dist}(G, x_{\text{robot}}, f_i)$;
        if $cost <$ laserRange $+ \epsilon_l$ **then**
          nearFrontierFound$\longleftarrow$**true**;
          Break; *// because a frontier is found*
        end if
      end for
    end while
  end if
  // get the best frontier point
  $\text{idx}_{\text{goal}} \longleftarrow \text{argmax}_i(IG(f_i) - w_d.\text{Dist}(G, x_{\text{robot}}, f_i))$;
  $x_{\text{goal}} \longleftarrow f_{\text{idx}_{\text{goal}}}$
end while

controller is used to control the linear velocity (v_{robot}) of the robot toward the current goal (x_{goal}).

C. FRONTIER EXPLORATION

As explained before, the frontiers are detected by two graphs, which in this paper are named the global and the local RRT graphs. This method was initially invented by [20], where the frontiers were collected in a single array and then filtered according to their information gain and position

on the occupancy grid map (free, unknown or occupied). We perform a slight modification to ignore a new frontier where the location is nearby with one of the existing filtered frontiers. Having some adjacent frontiers only makes the frontier selection take a longer time.

Define $F = \{f_1, f_2, .., f_n\}$ as a set of filtered frontiers. Initially, assign the nearest F as the goal. The robot will continue to navigate as long as the goal is still far away. When the robot is near the goal, which is less than a small distance (ϵ_d), the robot gives priority to search for a new frontier nearby. The distance between the robot and a frontier is intuitively less than the laser range finder plus a small constant ϵ_l considered as the new frontier in the newly explored area. Without this, the robot may return to another frontier where the location is far away from the robot, which is ineffective and gives the robot a zigzag trajectory.

Dist(.) is the distance based on the global RRT graph (G), which estimates geodesic distance, not Euclidean distance, as the area is nonconvex. A timeout handle should be used because there is a possibility that the robot cannot find a new near frontier. After a new near frontier is obtained or a timeout is reached, it changes the goal by using all of the filtered frontiers as the candidates considering the information gain in a frontier (IG(f_i)) and a geodesic distance from the robot pose to f_i weighted by a constant w_d. The information gain is obtained by calculating the area of the unknown map around the frontier.

D. GRAPH-BASED KERNEL DM+V

In this section, the proposed method, named graph-based Kernel DM+V, is elaborated. A graph representation is used instead of a grid representation because it is more scalable and applied in a concave area. An illustration of how graph-based Kernel DM+V works is shown in Fig. 3.

Define x_t as the sample gas measurement point with T samples and x_i as an element in a set of graph vertices (V_D) located inside a circle with diameter D and center coordinates in x_t. A set V_D is formally defined in Eq. (1).

$$V_D = \{x_i | x_i \in V, d(x_i, x_t) \leq D/2\} \tag{1}$$

By using the Gaussian weighting function $\mathcal{N}$, a set of integrated importance weights (Ω) and integrated weighted gas measurements (R) are formulated in Eq. (2). The gas concentration reading and the kernel width are defined by r_t and σ. In this case, the function $d(x_t, x_i)$ is not Euclidean but can be made geodesic by adding the costs of edges connecting x_t and x_i. Because x_t is not connected to the graph, it is approximated by the nearest vertex in V_D ($x_{nearest}$), as shown in Fig. 3.

$$\begin{aligned} \Omega_i &= \Sigma_{t=1}^{T} \mathcal{N}(d(x_t, x_i), \sigma) \\ R_i &= \Sigma_{t=1}^{T} \mathcal{N}(d(x_t, x_i), \sigma).r_t \end{aligned} \tag{2}$$

Confidence values α_i should be computed before computing the mean gas concentration r_i, which are formalized in Eq. (3) and (4), respectively. The variable r_0 is the average of

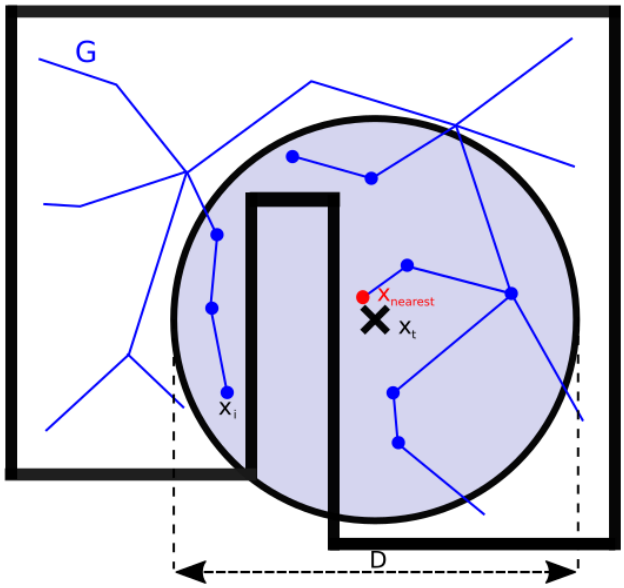

FIGURE 3. Illustration of how the graph-based Kernel DM+V is determined. The blue lines are the graph with the blue dots being elements of V_D in (1). The variable x_i is one of the elements in a set V_D, and x_t is one of the measurement sampling poses. The variable $x_{nearest}$ is the nearest vertex in V_D from x_t.

sensor readings, and σ_Ω is the scaling parameter.

$$\alpha_i = 1 - e^{-\Omega_i^2/\sigma_\Omega^2} \tag{3}$$

$$r_i = \alpha_i \frac{R_i}{\Omega_i} + \{1 - \alpha_i\} r_0 \tag{4}$$

After the mean gas concentrations are obtained, the integrated weighted variance V_i and the variance map v_i can be computed by using Eq. (5). The value of $r_{t(i)}$ is the prediction of the mean concentration in $t(i)$. $t(i)$ is the nearest vertex to x_t, while v_0 is the average variance from every vertex.

$$\begin{aligned} V_i &= \Sigma_{t=1}^{T} \mathcal{N}(d(x_t, x_i), \sigma)(r_t - r_{t(i)})^2 \\ v_i &= \alpha_i \frac{V_i}{\Omega_i} + \{1 - \alpha_i\} v_0 \end{aligned} \tag{5}$$

If there are more sample concentration gas readings close to a vertex x_i, then the importance weight value Ω_i will be higher. The confidence value α_i is exponentially increased when the importance weight increases. Suppose the confidence value in a particular vertex is high. In that case, confidently, the mean and variance of the gas concentration will be closer to the norm of integrated weighted gas measurement. Otherwise, the mean and variance are approximated by the average of the sensor readings and the average variance from every vertex when the confidence is low.

The larger the global RRT graph is, the longer the computation time of this graph-based Kernel DM+V. This also causes a delay in the selection of the optimal goal. The computational complexity of graph-Kernel DM+V is $O(n|V_D|)$ (n is the number of gas samples and $|V_D|$ is the number of vertex elements inside D area). Therefore, we use three concurrent modules to calculate this graph-based Kernel DM+V. The

first module updates all of the mean and variance values in each vertex, while the second module updates only the mean and variance in the vertices near the robot. The third module is needed to collect and associate the data from modules one and two. As each edge has a distance cost, adding some costs from some edges is faster than calculating the Euclidean distance.

E. GAS EXPLORATION-EXPLOITATION STRATEGY

The graph-based Kernel DM+V provides the mean and variance, and their values are considered two important factors affecting gas exploration-exploitation strategy. A particular area with a high mean value in gas distribution mapping means that the gas concentration is high or dangerous. A specific area with high variance means no close gas sensor reading, or the set of gas sensor readings in that area has a high standard deviation due to wind disturbance. If there is a vertex where the mean value is high but the variance is still high, then the robot will go there, as it is interesting to exploit because the concentration is high. Moreover, the variance value will decrease as the amount of gas sample near that vertex increases.

If there is no vertex with a high mean value with high variance, the robot can explore another place, visiting one of the nearest frontiers. The robot will open a broader map and exploit the gas again if another vertex is interesting to visit (high mean and variance value). The Finite State Machine (FSM) diagram is shown in Fig. 4.

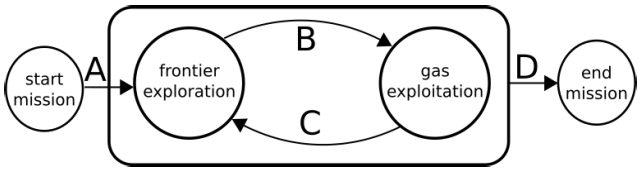

FIGURE 4. The finite state diagram of the proposed strategy. There are four states: (1) start mission, (2) frontier exploration, (3) gas exploitation, and (4) end mission. The four transitions are (A) if a frontier is detected, (B) if there is at least a vertex with high mean and high variance, (C) if all of the vertices with high mean have low variance and a frontier exists, (D) if all of the vertices with high mean have low variance and no frontier is detected anymore.

The mean threshold (m_{th}) is defined as a particular percentage value of the current maximum mean, while the variance threshold (v_{th}) is a constant. For instance, if the current maximum mean is 100 ppm and the percentage value is 5%, then the mean threshold is 5 ppm.

If we set the percentage of the mean threshold lower, then the robot will exploit more areas containing lower concentration gases and take a longer time. Instead of exploiting low "nonimportant" gas concentrations, exploring a new area is more important, even if the mission area is very wide. The variance threshold is also related to the trade-off. The robot waits until the variance is low enough before it goes for exploration. The lower the variance threshold, the longer the robot does the exploitation and the more limited the coverage area.

It needs to be clarified that the robot will not explore the same area that has been previously visited except for this particular case. The robot has to go to another frontier and pass the previously explored area. In that area, the gas variance increases due to measurement change. Measurement changes occur because of the dynamics of gas propagation. Because the gas variance increases above v_{th}, the robot will explore that area until the variance reduces to below v_{th}.

F. SYSTEM INTEGRATION

The complete system integration is explained below and illustrated by a diagram in Fig. 5.

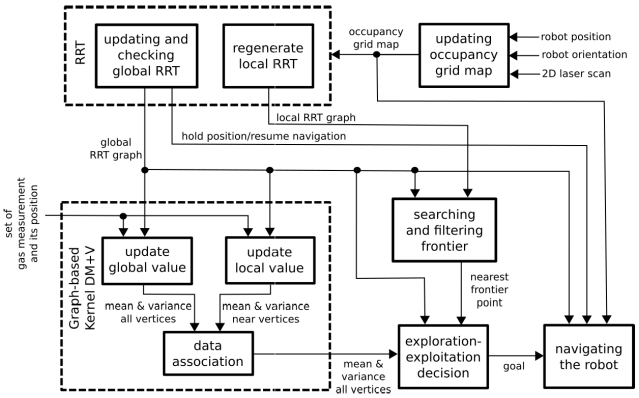

FIGURE 5. The whole system diagram.

As shown in this figure, the whole system consists of nine modules that a robot must run. A mapping algorithm is run to generate the occupancy grid map using 2D LIDAR and robot position and orientation information obtained by GPS and Inertial Measurement Unit (IMU) sensors. The occupancy grid map is then used as the RRT algorithm reference so that the global RRT graph can be used as the goal candidates, the graph-based Kernel DM+V vertices, and the navigation guidance. When a graph element coincides with an obstacle, the robot should hold the position so that the RRT module has to send a signal to the navigation module. Global and local graphs are also used to find the frontier points, which are filtered according to their information gain value [20].

The robot pose is used along with the gas sensor to update the set of gas readings using the proposed graph-based Kernel DM+V method. There are three modules according to the graph-based Kernel DM+V that are explained in the previous subsection. All mean and variance values are used to decide the exploration-exploitation problem to generate the best goal for the robot.

III. EXPERIMENTAL RESULTS

In this experiment, two different buildings are used to evaluate the method. The 3D buildings of the environment are shown in Figs. 6 and 7 which are a refinery and campus building, respectively. A static gas source is placed in each building environment. A rotary-wing UAV flying with static altitude is used as the robot and initially starts the mission

somewhere in the building. A simple anti-windup PID controller is used. The flight time is limited to 20 minutes. The sensor is mounted in the bottom center of the UAV-type robot. A simulation using a Robot Operating System (ROS) platform is conducted, whereas the environment and vehicle model are run in the PC with Gazebo, and the mapping, navigation, graph Kernel DM+V, and other processes related to the modules in Fig. 5 are processed in NVIDIA Jetson TX2. GADEN (a 3D gas simulator) and a Computational Fluid Dynamics (CFD) software are used to simulate the gas dispersion.

Two metrics are used to measure the performance of our proposed method: (1) the average of variances on the vertices that have a mean concentration greater than a threshold and (2) the coverage area in the regions where the gas concentration is high. The average of variance is calculated by using the estimated variance of graph-based Kernel DM+V. Two scenarios are conducted in different environments. Scenarios I and II are conducted by using a refinery (300 m $\times$ 200 m) and campus building (500 m $\times$ 700 m), as shown in Figs. 6 and 7, respectively.

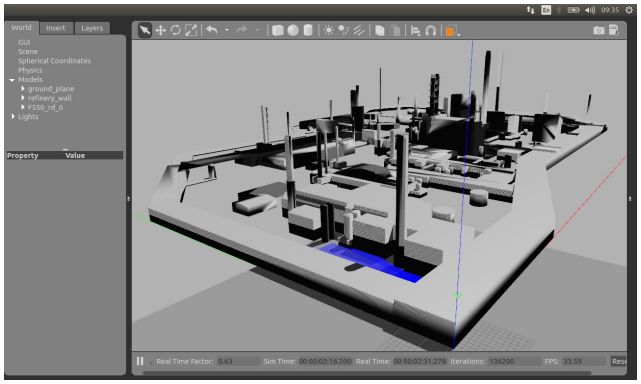

FIGURE 6. 3D illustration of the oil refinery building.

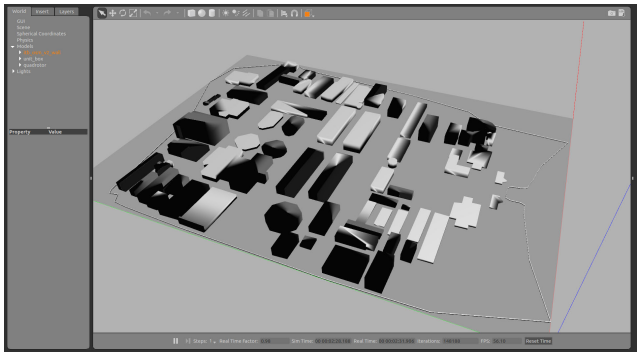

FIGURE 7. 3D illustration of Institut Teknologi Bandung (ITB) campus.

A. SCENARIO I

In this scenario, a simulated UAV-type robot starts the mission far from the gas source. The result of using exploration with frontier only and with a simple objective function with exploration-exploitation trade-off toward the global maximum of mean and variance are also shown to compare our proposed method. The objective function equation is shown in Eq. 6, whereas α and β are constant for weighting the mean and variance, respectively. Five simulations of each method are run and averaged to test the repeatability.

$$x_{goal} = \arg\max_{i} \left(\alpha r_i + \beta v_i\right) \tag{6}$$

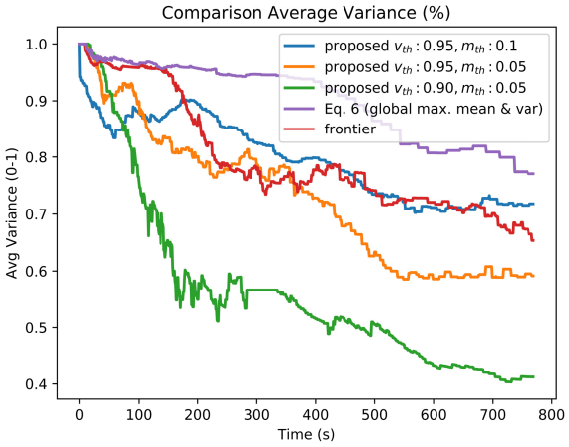

FIGURE 8. Comparison of average variance between five different decision-making strategies.

Fig. 8 shows the average variance only at the hazardous area against time. Sometimes, the average variance increases because the robot finds a new interesting vertex, but the average variance trend decreases. The coverage of the hazardous area against time is illustrated in Fig. 9.

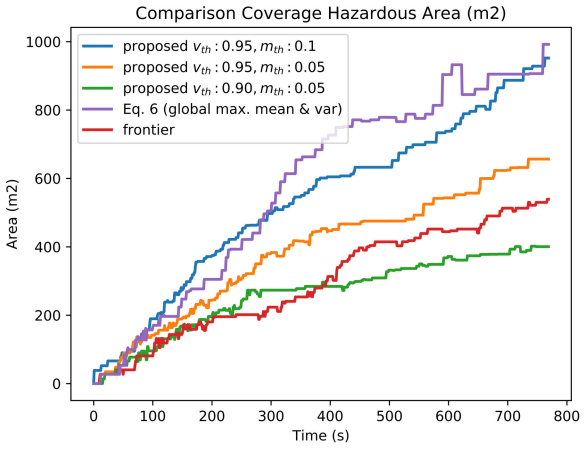

FIGURE 9. Comparison of hazardous coverage area between five different decision-making strategies.

Three different constants are used while testing our proposed method. The trade-off clearly shows that the lower the variance value is, the lower the coverage area. This means that the more confident the gas concentration value obtained, the less hazardous the area mapped. The user can then define

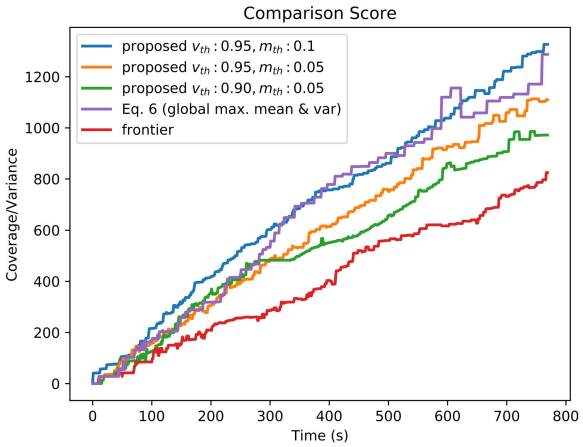

FIGURE 10. Comparison of scores between five different decision-making strategies.

the constant according to the needs. The lower the variance threshold (v_{th}) is, the more confident the gas distribution map obtained but the smaller the coverage area. The lower the mean threshold (m_{th}) is, the wider the coverage area, but the more uncertain the gas distribution map obtained.

From the average of variance and the coverage in a hazardous area, a score that is calculated by dividing the coverage area by the variance is obtained and shown in Fig. 10. The frontier-only exploration has the lowest score, and our proposed method and the global maximum optimization method have almost the same performance depending on the constants used.

B. SCENARIO II

In this scenario, a simulated UAV-type robot starts the mission near the gas source in the middle of the campus area. The performance graphs are shown in Figs. 12 and 13. The characteristics of the average variance and the coverage area obtained are the same as those in the prior scenario. However, the frontier, global maximum optimization and APF cannot cover a wide area. The frontier exploration strategy does not consider the gas mean and variance. The global maximum optimization strategy and APF do not revisit the hazard area and decrease the variance. Both APF and global maximum

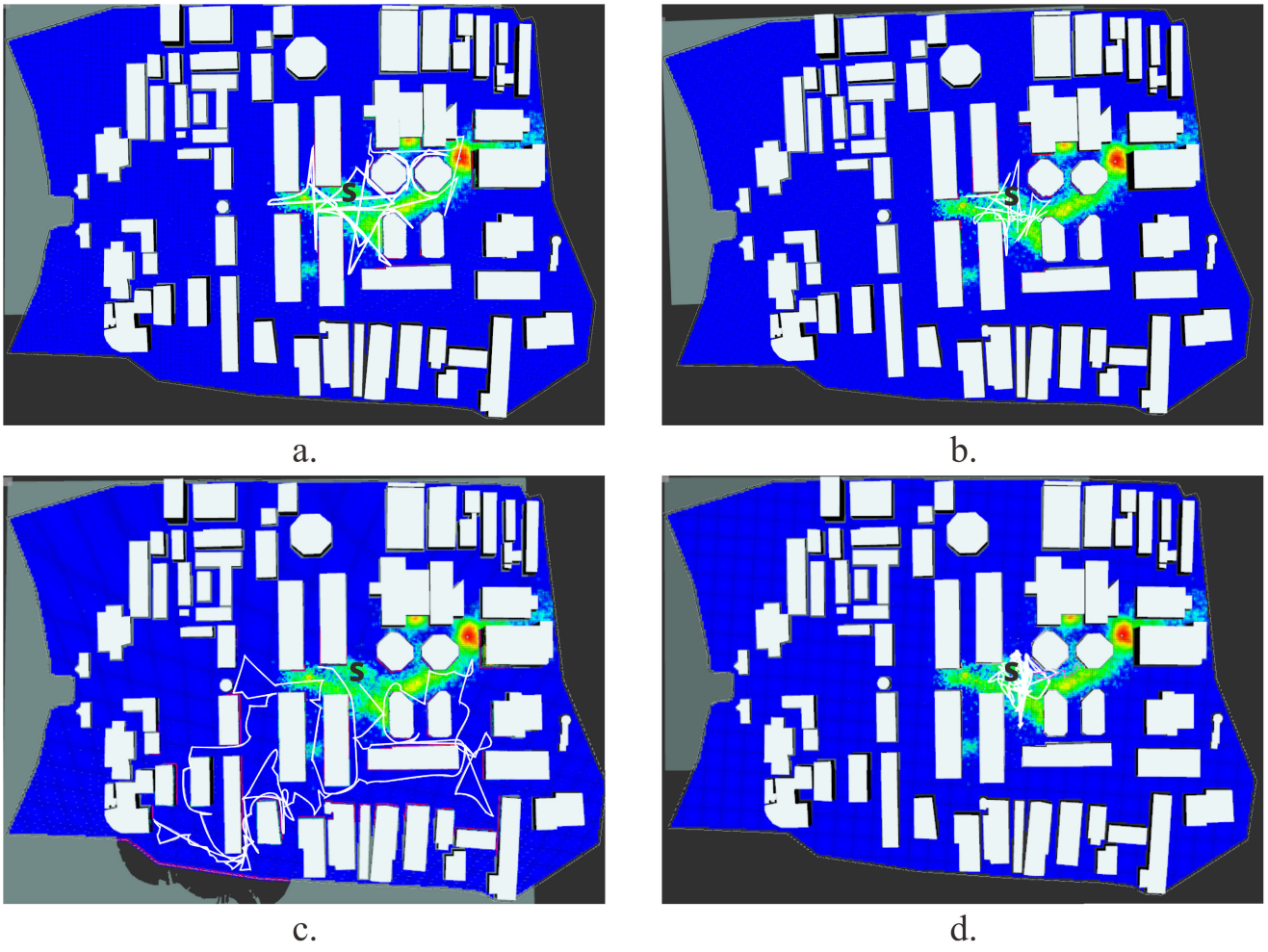

FIGURE 11. Some typical robot trajectories using: (a.) switching, (b.) global maximum optimization, (c.) frontier only and (d.) APF. White paths are the robot trajectories, purple to red color is the ground truth concentration of gas distribution, white polygons are the obstacles. S is the start position of the robot.

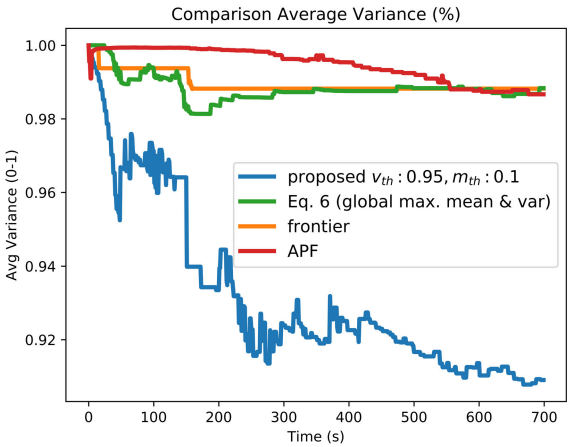

FIGURE 12. Comparison of average variance between five different decision-making strategies on the ITB campus.

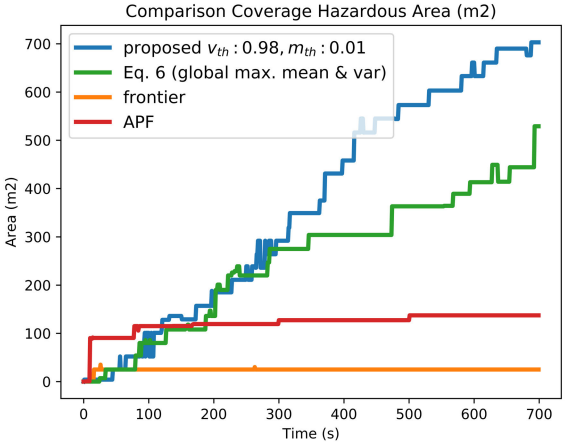

FIGURE 13. Comparison of hazardous coverage area between five different decision-making strategies on the ITB campus.

optimization strategy do not have a mechanism to open a new territory rapidly as the frontier exploration does.

Fig. 11 shows how the robot moves by different strategies. The switching strategy keeps the robot in a hazardous area. The frontier strategy always opens a new area, while the global maximum optimization strategy is slow to explore. The decision-making computation in the global maximum optimization strategy is slower because it should use all of the gas samples.

Using the global maximum optimization and APF requires all gas concentration samples, including low values outside the hazardous area, as it has to decrease the variance in an area where the robot takes the sample. If the low value is filtered, the robot will remain in a place where the variance does not decrease. By using the global maximum optimization strategy, the Kernel DM+V process is slower than the switching strategy, although there is a local update mechanism. Therefore, one of the advantages of the switching strategy is that it does not need a low gas concentration

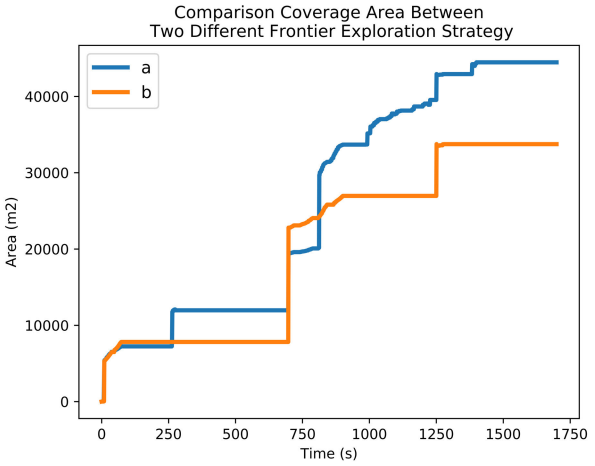

FIGURE 14. Comparison of coverage area between two different frontier exploration strategies. (a.) using method from [20]. (b.) using proposed algorithm in Algorithm 3.

sample so that gas extrapolation can be performed faster. Global maximum optimization and APF are not suitable for large and unknown areas because they do not consider how to open unknown areas as frontier exploration does.

In this scenario, the result of using Algorithm 3 as the frontier exploration is shown and compared with the standard frontier selection without handling the new near frontier. As shown in Fig. 14, the coverage area of using Algorithm 3 is wider.

C. COMPARISON BETWEEN GRID AND GRAPH KERNEL DM+V IN NONCONVEX AREA

A graph-based approach to estimate the gas distribution map is proposed, named the graph Kernel DM+V. In this subsection, the grid and graph Kernel DM+V performance in a nonconvex area with a map size is 20 $\times$ 20 meters with one-meter grid size.

It is shown both in Fig. 15 and Table 1 that computing the grid-based Kernel DM+V in a nonconvex environment is very slow. The computation time of the grid Kernel DM+V is significantly longer than that of the graph-based model, even with fewer gas samples.

From the original paper on the Kernel DM+V [17], in which the gas distribution map is estimated in a free-obstacle map, the computation complexity is $\mathcal{O}[n(\frac{\sigma}{c})]$ with n, σ and c are the number of gas samples, the kernel distance and grid cell size, respectively. In a building which is not free from obstacles, the Dijkstra algorithm should be used to compute the distance as the obstacle exists so that the complexity is $\mathcal{O}[n(\frac{\sigma}{c})^2]$. With a graph-based approach, the mean gas concentration of a vertex influences other vertices, and the cost of each edge is initially available by the RRT algorithm. Therefore, the complexity computation of the graph-based Kernel DM+V is $\mathcal{O}[n.|V_D|]$ with V_D being a set of vertices inside the kernel.

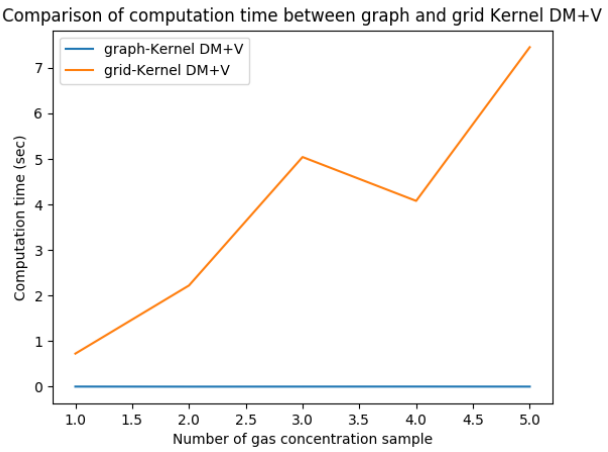

FIGURE 15. Comparison between grid and graph Kernel DM+V in nonconvex area.

TABLE 1. Comparison of computation time between graph and grid Kernel DM+V.

Number of sample	Computation time (sec)	
	Graph Kernel DM+V	Grid Kernel DM+V
1	0.003	0.72
2	0.001	2.22
3	0.002	5.04
4	0.003	4.08
5	0.002	7.45
100	0.07	174.74

IV. CONCLUSION

We have proposed a strategy for a robot to conduct area exploration, especially in a 2D spatial gas distribution. By incorporating an RRT graph, frontier selection and graph-based Kernel DM+V, the robot can accomplish the mission in a very wide environment.

Overall, the performance obtained in the experimental results shows that our proposed method is slightly better than the others but with selected appropriate constants. The methods that are used for generating the RRT graph, frontier selection and goal decision making still need constants selection. This problem must be addressed in future work, as the constants need to be found with empirical simulation.

Generally, our approach can be used not only for UAV, but also unmanned ground vehicle or unmanned surface vehicle as long as they carry a 2D LIDAR, RTK-GPS and gas sensor. In a real mitigation mission, instead of driving the robot manually or based on waypoints, our approach is more effective as the robot does not have to explore areas that are not contaminated. The most important thing that should be set before the mission is the gas mean and variance threshold. If the operation time of the robot is short, it is suggested to choose relatively big values of both gas mean and variance threshold to allow the robot covers a vast area with a short time although the confidence of the gas distribution map will be relatively low.

Our future work is also focused on the experiment with a real UAV in a real field. Moreover, an adaptive strategy for the gas exploration-exploitation trade-off will eliminate some constants related to the mean and variance value of the gas concentration. However, it is challenging to apply it to a spatial gas distribution, as this is very dynamic. This work will be extended by using a multi-robot by keeping one global RRT graph and one global Kernel DM+V update and some local RRT graphs and local Kernel DM+V updates corresponding to the number of robots.

An extension to 3D for the gas exploration in a very large area is another issue. However, the use of graphs in a 3D space will make a significant difference compared to the use of grids. Also, if the UAV needs to fly with low altitude (1-2 meters), some dynamic obstacles must be considered. This may be done by modifying our algorithm by efficiently reconstructing the graphs and frontiers according to the appearance of the dynamic obstacles.

REFERENCES

[1] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 33, no. 3, pp. 367–385, Jun. 2003.

[2] S. Sriramachari, "The Bhopal gas tragedy: An environmental disaster," *Current Sci.*, vol. 86, no. 7, pp. 905–920, Apr. 2004.

[3] J. J. Sharples, G. J. Cary, P. Fox-Hughes, S. Mooney, J. P. Evans, M.-S. Fletcher, M. Fromm, P. F. Grierson, R. McRae, and P. Baker, "Natural hazards in australia: Extreme bushfire," *Climatic Change*, vol. 139, no. 1, pp. 85–99, Nov. 2016.

[4] K. H. Low, J. M. Dolan, and P. Khosla, "Adaptive multi-robot wide-area exploration and mapping," in *Proc. 7th Int. Joint Conf. Auton. Agents Multiagent Syst.*, vol. 1, May 2008, pp. 23–30.

[5] K. Tiwari and N. Y. Chong, *Multi-robot Exploration for Environmental Monitoring: The Resource Constrained Perspective*. New York, NY, USA: Academic, 2019.

[6] A. Carron, M. Todescato, R. Carli, L. Schenato, and G. Pillonetto, "Multi-agents adaptive estimation and coverage control using Gaussian regression," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2015, pp. 2490–2495.

[7] P. Neumann, S. Asadi, J. H. Schiller, A. J. Lilienthal, and M. Bartholmai, "An artificial potential field based sampling strategy for a gas-sensitive micro-drone," in *Proc. IROS Workshop Robot. Environ. Monit. (WREM)*, San Francisco, CA, USA, 2011, pp. 34–38.

[8] P. P. Neumann, V. H. Bennetts, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, "Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms," *Adv. Robot.*, vol. 27, no. 9, pp. 725–738, 2013.

[9] C. Rhodes, C. Liu, and W.-H. Chen, "Informative path planning for gas distribution mapping in cluttered environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2020, pp. 6726–6732.

[10] K. Kamarudin, A. Y. Shakaff, and V. H. Bennetts, "Integrating SLAM and gas distribution mapping (SLAM-GDM) for real-time gas source localization," *Adv. Robot.*, vol. 32, no. 17, pp. 903–917, Sep. 2018.

[11] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI/IAAI*, Jul. 2002, pp. 593–598.

[12] M. Peasgood, J. McPhee, and C. Clark, "Complete and scalable multi-robot planning in tunnel environments," *IFAC Proc. Volumes*, vol. 39, no. 20, pp. 26–31, 2006.

[13] J. Vallvé and J. Andrade-Cetto, "Active pose SLAM with RRT," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 2167–2173.

[14] U. Orozco-Rosas, K. Picos, and O. Montiel, "Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots," *IEEE Access*, vol. 7, pp. 156787–156803, 2019.

[15] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, "Mobile robot path planning using membrane evolutionary artificial potential field," *Appl. Soft Comput.*, vol. 77, pp. 236–251, Apr. 2019.
[16] M. Reggente and A. J. Lilienthal, "The 3D-kernel DM+V/W algorithm: Using wind information in three dimensional gas distribution modelling with a mobile robot," in *Proc. IEEE SENSORS*, Nov. 2010, pp. 999–1004.
[17] A. J. Lilienthal, M. Reggente, M. Trincavelli, J. L. Blanco, and J. Gonzalez, "A statistical approach to gas distribution modelling with mobile robots–The kernel DM+V algorithm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, Oct. 2009.
[18] J. G. Monroy, J.-L. Blanco, and J. Gonzalez-Jimenez, "Time-variant gas distribution mapping with obstacle information," *Auto. Robots*, vol. 40, no. 1, pp. 1–16, Jan. 2016.
[19] C. Stachniss, C. Plagemann, and A. J. Lilienthal, "Learning gas distribution models using sparse Gaussian process mixtures," *Auto. Robots*, vol. 26, nos. 2–3, pp. 187–202, Apr. 2009.
[20] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1396–1402.
[21] J. Monroy, V. Hernandez-Bennetts, H. Fan, A. Lilienthal, and J. Gonzalez-Jimenez, "GADEN: A 3D gas dispersion simulator for mobile robot olfaction in realistic environments," *Sensors*, vol. 17, no. 7, p. 1479, Jun. 2017.
[22] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Dept. Comput. Sci., Iowa State Univ., Ames, IA, USA, Tech. Rep. TR 98-11, Oct. 1998. [Online]. Available: http://janowiec.cs.iastate.edu/papers/rrt.ps

YAQUB A. PRABOWO received the B.Eng. and M.Eng. degrees in electrical engineering and intelligent systems and control engineering from the Institut Teknologi Bandung (ITB), Indonesia, in 2015 and 2019, respectively. He is currently pursuing the Ph.D. degree in robotics with ITB.

From 2019 to 2020, he was chosen as a Research Scholar with the Centre for Autonomous System (CAS), University of Technology Sydney. Since 2012, he has been begun to learn robotics in a robotics club with ITB. He attended some humanoid robot competitions, from 2013 to 2015, with the national and international levels. His team was awarded the 2nd winner of the international robot competition in FIRA Robo World Cup in Shijingsan Stadium, Beijing, China.

BAMBANG R. TRILAKSONO (Member, IEEE) received the Graduate degree in electrical engineering from the Bandung Institute of Technology, and the master's and Ph.D. degrees in electrical engineering from Waseda University, Japan. He is currently a Professor with the Control and Computer System Research Group, Bandung Institute of Technology. His research interests include optimal control, robust control, intelligent control and systems, discrete event systems, control applications, robotics, and embedded control systems.

EGI M. I. HIDAYAT received the Graduate degree in electrical engineering from the Bandung Institute of Technology, the Master of Science degree in control and information system from Universitat Duisburg Essen, Germany, and the Ph.D. degree in electrical engineering from the University of Uppsala, Sweden. He is currently a Lecturer with the School of Electrical Engineering and Informatics, Bandung Institute of Technology. His research interests include modeling and identification systems, control systems, and robotics.

BRIAN YULIARTO received the Graduate degree in physics engineering from the Bandung Institute of Technology, and the master's and Ph.D. degrees in quantum engineering and systems science from The University of Tokyo, Japan. He is currently a Professor with the Advanced Functional Material Group, Bandung Institute of Technology. His research interests include nanoenergy materials and gas sensor development.