Constructing Full-Coverage 3D UAV Ad-Hoc Networks Through Collaborative Exploration in Unknown Urban Environments

Narangerelt Batsoyol, YeonJin Jin, and HyungJune Lee
Department of Computer Science and Engineering
Ewha Womans University, Seoul, South Korea
Email: hyungjune.lee@ewha.ac.kr

***Abstract*—We consider a 3D network construction problem in the post-disaster scenario, where large urban areas are communication-wise isolated from the outside environment due to the severely damaged network infrastructure. Our main goal is to reconnect the isolated regions with the outside environment using unmanned aerial vehicles (UAVs) by building 3D aerial ad-hoc networks. Prior to network construction, we aim to capture the global map information over region of interests (RoI) by exploring all obstacles in the unknown region. We propose an efficient technique for collaborative 3D terrestrial exploration using multiple UAVs based on our distributed path planning algorithm, which finds collision-free exploration paths. Then, we present an optimal full-coverage 3D aerial ad-hoc network construction by deploying the minimum number of UAVs to indispensable spots while obtaining maximum network coverage. Simulation results demonstrate that our proposed exploration scheme outperforms several counterpart algorithms in terms of traversal time and redundant visit rate. Also, our network construction algorithm guarantees almost full coverage toward terrestrial space with only minimal UAV usage.**

I. Introduction

In a post catastrophic disaster situation, network infrastructure tends to be severely collapsed. Geographical regions in a large urban area occupied with various low-rise, mid-rise, and high-rise buildings can be communication-wise isolated, or totally secluded from the outside environment. It would be a promising approach to reconnect these isolated regions with the outside environment (toward base stations) by constructing ad-hoc networks using unmanned aerial vehicles (UAVs).

Since any global map information over the cluttered region is unknown, it is essential to explore the given region of interests (RoI) to capture all the obstacles' distribution status including their height information without missing any unvisited areas in a fast manner. UAVs can also be used as flexible communication relays to form an aerial ad-hoc network due to their less movement constraint and high maneuverability.

Regarding efficient area exploration, many researchers in the robotics field have conducted space exploration with obstacle avoidance mostly in two dimensions, whereas the space exploration in three dimensions (3D) has not been actively studied. Several algorithms were implemented in 3D environments using UAVs: randomly sampling search algorithms such as rapidly-exploring random tree and probabilistic roadmap, and optimal search algorithms [8], [9]. These algorithms, however, have not explicitly explored the area with obstacles.

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2015R1D1A1A01057902).

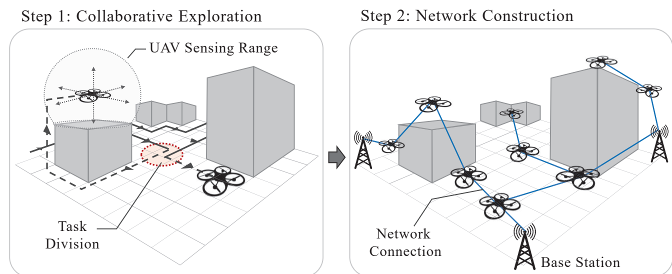


Fig. 1. System overview with collaborative exploration and network construction using UAVs

The coverage and connectivity problem in 3D ad-hoc networks have attracted a great attention among researchers in recent years. Especially, constructing a reliable network in atmospheric or underwater environments is a challenging task. Underwater acoustic sensor networks have been studied in [1], while climate monitoring and weather forecasting problems have been solved by deploying 3D aerial networks [2], [3], [6]. Most of previous works related with 3D coverage problem have used some space filling approach by dividing a space into equal polyhedrons based on transmission range [2], [10]. However, the environment is assumed to be free from obstacles, and thus, they have limitations in directly applying to more practical settings.

In this paper, we aim to solve two key problems of 1) how to explore an unknown complex 3D space occupied with obstacles in a fast manner using multiple UAVs, and 2) how to construct a full-coverage 3D ad-hoc network in an isolated cluttered environment.

We propose an efficient method for collaborative exploration over unknown urban environment using multiple UAVs based on a novel path planning algorithm. The proposed algorithm finds optimal and collision-free exploration paths in 3D space. The main goal is to completely explore the whole area over RoI and build a complete height map as soon as possible, while avoiding redundant visit at each specific location.

Then, we present a heuristic yet efficient full-coverage 3D aerial network construction algorithm. The challenge is to find the best 3D deployment location for each UAV such that UAVs can extend wireless coverage toward all the terrestrial buildings and spaces, and can also communicate with other UAVs and base stations. We aim to minimize the number of deployed UAVs, while guaranteeing maximum coverage.

Our paper is organized as follows: After presenting our system model in Sec. II, we describe our collaborative exploration with path planning in Sec. III and network construction in Sec. IV. We validate our algorithms in Sec. V and finally conclude our work in Sec. VI.

II. System Model

We consider the problem of constructing full-coverage ad-hoc networks using UAVs in unknown urban environments. Our goal is to perform an efficient 3D exploration for capturing the height dynamics over the RoI and dispatch UAVs for extending wireless coverage toward terrestrial areas and also constructing aerial ad-hoc networks.

For this work, we employ a theoretical unit disk model for the communication model with communication range R. A UAV can communicate with other UAVs and base stations located at the corner of RoI via a wireless radio such as 802.11 or 802.15.4. We let UAVs traverse over virtual 3D grid cells to facilitate an otherwise complex problem.

The xy axis represents a cross-sectional 2D representation from the above, while the z axis is the height level in 3D space. Each individual grid cell's information is provided by 0 or 1 depending on whether an obstacle is located at the cell. It is assumed that UAVs can sense any obstacles in its directly adjacent cells within the sensing range using radar sensors or laser scanners [4].

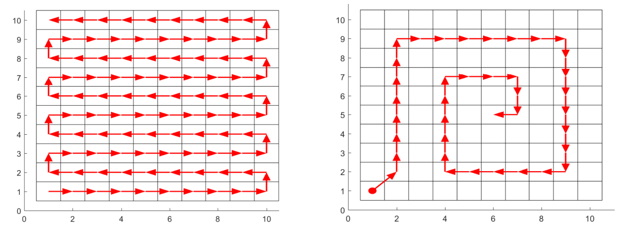
The problem of constructing full-coverage aerial ad-hoc networks can then be described with two sub-problems: 1) 3D exploration for height mapping by UAVs and 2) aerial ad-hoc network construction, as illustrated in Fig. 1.

III. Collaborative Exploration

To construct aerial ad-hoc networks with UAVs in an unknown territory occupied with various low-rise, mid-rise, and high-rise buildings, it is essential to explore the given RoI without missing any unvisited areas in a rapid manner. This can be achieved by finding the height of cluttered buildings in a geographically regular basis and constructing a height map over the RoI. Further, if we are allowed to use multiple UAVs for the exploration process, they would rather collaborate with each other to find optimal and collision-free exploration paths in 3D space for even faster execution. The main goal is to complete the height mapping process over the given RoI as soon as possible, while avoiding redundant visits at a specific location.

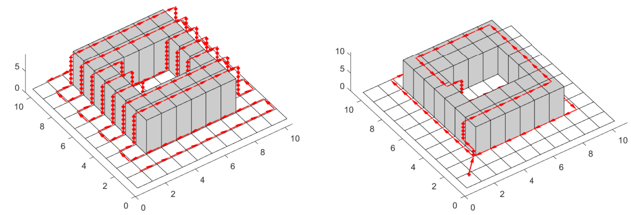
Each individual UAV agent follows its own distributed path planning decision. Whenever the UAV agent visits a certain cell as close as possible on the surface of 3D obstacles from the air, it updates its visited cell list and records the height of the obstacle located below at the same xy coordinate.

If each UAV agent follows its own distributed path planning without considering other agents' exploration progress, it is inevitable to have redundant cell visits, leading to inefficient coverage, and delayed exploration. However, since two or more UAV agents can communicate each other within the radio range and exchange information about their visited cells, the redundant visit can be reduced. For this, some negotiation protocol for sharing their visited cells and the height of them is necessary. Further, collaborative exploration based on almost equal task division of visiting the remaining cells among the UAVs would increase exploration efficiency. All these factors should work together to help other UAVs to minimize redundant visits in the future, and the overall exploration procedure can finish as early as possible.



(a) Naive Algorithm (b) Our Algorithm

Fig. 2. Basic movement trajectory in a space with no obstacle



(a) Naive Algorithm (b) Our Algorithm

Fig. 3. Basic movement trajectory in a space with obstacles

A. Path Planning

We aim to design an efficient path planning algorithm for a single UAV to construct a complete 2D height map by exploring the RoI on the 3D space in a cluttered city with buildings. We suppose that each UAV can detect an obstacle in its directly adjacent cells, which means it can sense its surrounding $3 \times 3 \times 3$ cells in 3D space. Our algorithm performs exploration with two modes: non-obstacle exploration and obstacle exploration depending on whether an obstacle is detected.

1) Non-obstacle Exploration: For exploration, each UAV sets its basic movement direction randomly from clockwise or counter-clockwise direction. The UAV starts exploring from the border region of the RoI, and travels toward the center in a spiral manner, following the basic direction under the non-obstacle circumstance. Since the UAV can sense directly adjacent cells, it may well follow not along the border line, but along non-border cells that are one cell away from the border line. A naive scanning exploration can visit each cell at a time in a regular basis for complete coverage as in Fig. 2(a), whereas our proposed exploration can reduce the exploration completion time by quickly covering the RoI area with a spiral horizontal movement as in Fig. 2(b).

For the vertical exploration, the UAV starts exploring from the second level since it can sense cells in the lower and the upper levels. In case of detecting an obstacle during this procedure, the UAV changes to the obstacle exploration mode and gradually flies upward up to the top of the currently detected obstacle.

A UAV can move in North, South, West, and East direction while mapping, and saves its sensed cells in the *sensedList*. To determine its movement direction among these four directions, it counts the number of unsensed cells in its North, South, East, and West direction, and goes to the direction with the largest number of unsensed cells. If there are multiple candidate directions, the UAV selects a more highly prioritized direction based on its initially selected basic movement direction at the beginning of the non-obstacle exploration. For each direction, we hold some priority order for directions among *North*, *South*, *East*, and *West*. We prioritize the clockwise direction with the following order of *North* $\rightarrow$ *East* $\rightarrow$ *South* $\rightarrow$ *West*, while the counter-clockwise direction has the following priority order of *South* $\rightarrow$ *East* $\rightarrow$ *North* $\rightarrow$ *West*. The UAV continues to move based on the selected direction until it meets a border cell or an already-sensed cell. Upon encountering one of these cells, it finds a new direction based on the aforementioned procedure and follows the trajectory until it encounters an obstacle or completes the mapping process (for obtaining the height information for all the cells).

2) Obstacle Exploration: Once a UAV encounters an obstacle located at adjacent cells within its sensing range during the non-obstacle exploration, it changes the mode to the obstacle exploration mode. It first tries to fully encircle the base of a building of which the obstacle is a part. After finishing exploring the base of the building (before moving upward), the UAV obtains the knowledge of how large the obstacle group is and how its belonging cells are located. Except for directly visited cells during the encircling exploration, the UAV can list up a set of inner cells, *innerList* that need to be explored afterwards.

Upon the completion of encircling at the lowest level, it ascends up to the fourth level since the information of cells located at the third level has already been obtained during the lowest encircling. Then, the next cell to visit is determined by choosing a cell with the most adjacent unexplored cells from the *innerList* at the current level. If there are several candidate cells, we choose the closest one among them.

Then, the UAV generates a path from the currently visiting cell to the selected destination cell and starts moving horizontally and then vertically, or vertically and then horizontally. We choose one of two movement behaviors that has the larger number of unvisited cells from *innerList*. In case of encountering another even higher obstacle during the journey on the way, it performs the hill climbing and descending until reaching the current destination cell. The UAV climbs the hill until there is no obstacle cell found within its sensing range and follows its path. In case that there is no obstacle cell right below, the UAV descends until finding an obstacle cell right below or reaching the second level. These steps are repeated until all the cells from *innerList* are explored. Illustrative examples are shown in Fig. 3.

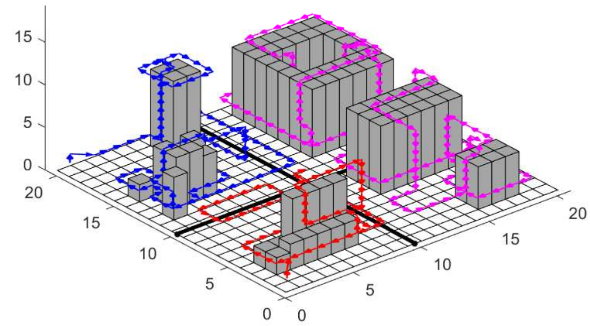


Fig. 4. Collaborative exploration using 3 UAVs for faster height mapping with 20×20 grid size and 4 sub-areas of 10×10 grid size where communication range R is 10

Once a group of obstacles for a building are fully explored, the UAV descends back to the second level and continues to explore all other cells over RoI, while transitioning back and forth to the non-obstacle exploration mode.

A more detailed path planning algorithm is described in Algorithm 1.

B. Task Division

We extend our path planning algorithm for a single UAV to the one with multiple UAVs. By letting multiple UAVs explore RoI in a collaborative way, the overall area exploration for obtaining all the height information of cells over RoI can be reduced. We propose a hierarchical task division method that divides a territory area into smaller sub-areas and assigns sub-areas to visit almost equally to communicable UAVs. Considering the communication range R, we divide the RoI into multiple sub-areas with the size of $R \times R$ so that a UAV located at the center of a sub-area can communicate with another at that of adjacent sub-areas.

At the start, each UAV thinks that it is responsible for exploring every sub-area in RoI and follows its own exploration path planning according to Sec. III-A. Each UAV maintains *assignedList* that saves the sub-area IDs that it needs to explore, and *visitedSubareaList* that updates already-visited sub-area IDs. Each UAV is subject to completely explore one sub-area first, and then is allowed to move to another. To determine which sub-area to visit next time, the UAV finds the nearest sub-area from *assignedList*. It continues to visit the remaining sub-areas from the list one by one until it meets another UAV, or every sub-area from the list is explored.

When several UAVs get encountered each other in the air within their communication range, they exchange the information of *visitedSubareaList* and which sub-area is currently being explored by each own UAV. Based on the shared information, the UAVs perform the task division procedure that distributes the remaining sub-areas to visit into them with the similar workload by updating *assignedList*. It should be noted that if there exists any obstacle between two UAVs within the communication range, we consider them non-communicable each other.

Algorithm 1 Distributed Path Planning Algorithm

Input: *Starting position of UAV, 3D cartesian coordinate*
Output: *2D height map H*
//Set Initial State
//$BasicDir$ shows priority of direction
//$Clockwise$ = [North-East-South-West]
//$Counter\text{-}Clockwise$ = [South-East-North-West]
//If the cell at xyz is an obstacle, then $C_{xyz} = 1$, else $C_{xyz} = 0$
$z_{initial}$ of UAV = 2;
$BasicDir$ = random($Clockwise$, $Counter\text{-}Clockwise$);
while (H is not completed) **do**
  // I. Select direction Dir to move
  Dir = Direction (N/S/E/W) of max(# of unexplored cells);
  if # of $Dir \geq 2$ **then**
    Choose Dir which comes first in $BasicDir$ list;
  end if
  // II. Follow trajectory
  while (H is not completed && no obstacle cell detected && not reached border cell && not reached already-sensed cell)) **do**
    Follow the path in Dir;
    Update $sensedList$
  end while
  // III-i. Obstacle detection
  if (At least one $C_{xyz} = 1$ in sensor range) **then**
    // i) Obstacle base encircling
    G= current position (starting location of obstacle exploring);
    while *next_cell* != G && *next_cell* != border cell **do**
      Find Cardinal Direction (of UAV's sensing area) which has $C_{xyz} = 1$;
      if ($C_{xyz} = 1$ is at East or NorthEast side) **then**
        Go to North;
      else if ($C_{xyz} = 1$ is at South or Southeast side) **then**
        Go to East;
      else if ($C_{xyz} = 1$ is at West or Southwest side) **then**
        Go to South;
      else if ($C_{xyz} = 1$ is at North or Northwest side) **then**
        Go to West;
      end if
      Update $sensedList$
      Save the detected obstacle cell in $obstacleList$;
    end while
    // ii) Obstacle height exploration
    Increase z to 4 & update $sensedList$;
    $innerList$ = cells encircled by $obstacleList$;
    S = xy location of cell in $obstacleList$ (which has max(# of unexplored adjacent obstacle cells));
    while (S exist) **do**
      $destCell$ = [x of S, y of S, current z level];
      Count # of cells in $obstacleList$ on the way if we go to $destCell$'s x first with current y then go to $destCell$'s y ; or go to $destCell$'s y first with current x then go to $destCell$'s x. A $path$ with max # of cells is chosen, else selected in random.
      while (Go to $destCell$ following the $path$) **do**
        if There exist higher level obstacles on the way to $destCell$ **then**
          Do hill climbing and descending until there is an obstacle right below;
          $H_{xy} = z - 1$, with xy of current position;
        end if
        Update $sensedList$
      end while
      Update S (from unvisited cells $\in \{obstacleList \cup innerList\}$);
    end while
    // iii) Descend to the lowest level
    If there is unvisited border/near border cell nearby, UAV goes to that cell. Else should go to the closest unvisited cell at current z.
    Decrease UAV's z value until $z = 2$ & Update $sensedList$;
  end if
end while

We represent the allocation of sub-areas to available UAVs as making bids at each iteration step. At each iteration, each UAV k is allocated to one sub-area and registers the sub-area's ID at its $assignedList_{UAV_k}$. We calculate a weight measure for each sub-area i to prioritize sub-areas to allocate to a better UAV as follows:

$$weight_{S_i} = D(Loc_{UAV_k}, C_{S_i}) + \sum_j (D(C_{assignedList_{UAV_k}(j)}, C_{S_i}) \tag{1}$$

where S_i denotes sub-area i, Loc_{UAV_k} is the current location of UAV k, C_{S_i} is the center of sub-area i, $assignedList_{UAV_k}(j)$ is the jth sub-area from *assignedList* of UAV k, and $D(A, B)$ denotes geographical distance between A and B. This weight measure means the cumulative distance measure for all possible combinations from the current UAV location, or already-assigned, but not-visited-yet sub-areas to a candidate sub-area.

In case that several UAVs select the same sub-area with the nearest distance, a UAV with the smallest number of visited sub-areas is chosen. If there still exist ambiguities, a UAV is randomly chosen and allocated to the sub-area. From the second iteration, the weight measure is used to quantify the distance measure from a group of its already-allocated sub-areas to a sub-area candidate and to allocate the remaining sub-areas to visit to UAVs.

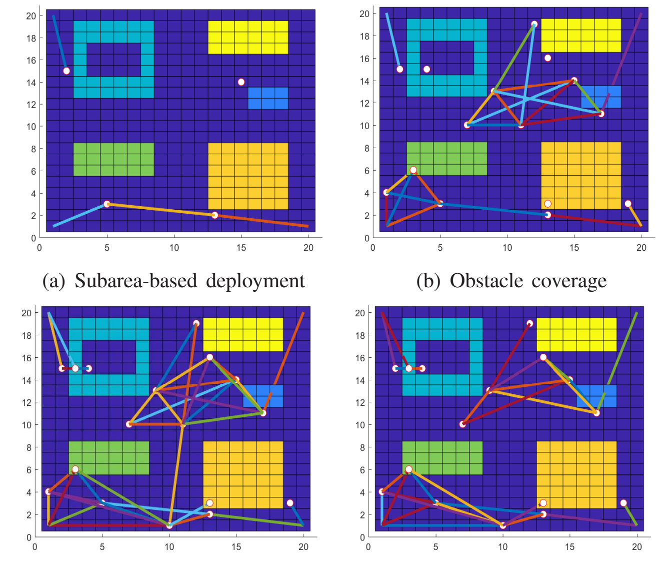
It should be noted that if multiple UAVs become aware of exploring the same sub-area within the communication range, we let a UAV with the larger number of already-visited cells in the sub-area be allocated to the current sub-area, whereas other UAVs stop exploring at the current sub-area and leave for another sub-area from their own *assignedList* by task division. An example of collaborative exploration is illustrated in Fig. 4.

After finishing exploring all the sub-areas from *subareaList*, each UAV goes back to the nearest corner cell where a base station is located and finishes its height mapping process via exploration. All the UAVs share their local map information through their connected base station and merge into a global height map over RoI.

IV. Network Construction

Once the height mapping process for the entire cells over RoI is completed, we focus on constructing a full coverage 3D ad-hoc network using UAVs. The challenge is to find the best location for each UAV to be deployed in 3D space such that UAVs can extend wireless coverage to all terrestrial buildings, and can communicate each other where at least one UAV is connected to a base station. We aim to minimize the number of deployed UAVs, while guaranteeing maximum wireless coverage. We suppose that a UAV can cover all nearby cells within the communication range R where there is no obstacle between them.

Motivated by some previous works on wireless coverage in 3D space [2], [10], we rely on the same hierarchical sub-area structure used for task division in Sec. III-B. The network construction problem using UAVs are divided into three steps:



(a) Subarea-based deployment (b) Obstacle coverage

(c) Network hole filling (d) Refinement

Fig. 5. Evolutionary network construction optimization after necessary steps where the circles represent deployed UAVs, and solid lines show valid communication links where base stations are located at four corners of RoI

1) sub-area coverage for fully covering the ground space with or without obstacles, 2) network hole coverage by deploying UAVs into the detected network holes, and 3) refinement process for discarding unnecessary UAVs.

A. Sub-area Coverage

Our network construction procedure is initiated with the first step of sub-area coverage. By deploying only one UAV to each sub-area, we check whether one UAV is enough to cover its responsible sub-area. In case that there exist non-reachable cells shadowed by nearby obstacles from the current UAV's location at the sub-area, we deploy more UAVs for guaranteeing 100% wireless coverage with a more aggressive manner. Further, to extend wireless coverage even to mid-rise and high-rise obstacles themselves, the deployment positions for additional UAVs should be determined.

1) Ground Space Coverage: The ground space coverage consists of iterative steps to fully cover each sub-area by finding out the best UAV positions where they can cover the maximum number of cells in each sub-area. For the ground space coverage, a UAV is deployed at $R+1$ level, and extends their wireless coverage down to the bottom level and up to $2R+1$ level.

At the first iteration, we assign each UAV to every sub-area, and each UAV counts all possible coverage cells considering obstacle shadowing for each deployment cell position inside its sub-area. In this stage, we need the same number of UAVs as the number of sub-areas as shown in Fig. 5(a).

We further perform a few more iterations to achieve 100% full coverage by deploying more UAVs to unfulfilled sub-areas. We find out the best positions to cover the obstructed cells in the ground space and deploy additional UAVs at the locations where the number of obstructed cells can be minimized. We continue to run this iterative procedure until there is no obstructed cell on the ground space.

2) Mid-rise and High-rise Obstacle Coverage: After finishing the ground space coverage, we start covering obstacle buildings higher than $2R+1$ level. We define $m = R+1$, which is the lowest level of a UAV's deployment for the ground space coverage, and $n = 2R+1$, which is the maximum coverage level from the level m by the UAV. In general, if we deploy a UAV at $m+i\times n$ level (where i is a natural number), its resulting wireless coverage ranges from $i\times n$ to $(i+1)\times n$ level.

Depending on a building's height level, we deploy the UAV either to the top of the building or at the side. If the building's height is between $i\times n$ and $m+i\times n$ level, the UAV can be placed on the top of the building. Otherwise, if the building height is between $m+i\times n$ and $(i+1)\times n$ level, the UAV cannot be physically placed inside the building, and thus, is relocated at the side. In this case, we need $(i-1)$ additional UAVs to deploy at the levels of m, $m+n$, $\ldots$, $m+(i-1)\times n$. To determine its xy coordinate for the UAV deployment, we check which position leads to covering the largest number of obstacle cells. In case that multiple UAVs are assigned to the same building at the side, they are deployed at the different levels with the same xy coordinate to get connected each other. If a UAV cannot fully cover a building with wide cross-section located at the same level, other UAVs are deployed to locations with different xy coordinates at the same level.

At this point, we achieve 100% sub-area coverage towards ground space and even building areas by dispatching additional UAVs to the sectored sub-area regions.

B. Network Hole Coverage

Although all the deployed UAVs are responsible for extending wireless coverage within their own subarea, the problem of constructing communication links among those UAVs and connecting to at least one base station have not been addressed yet. In this section, we present network construction among UAVs by finding out network holes and deploying additional UAVs as relays.

At the base station side, we list up all possible connections among UAVs and base stations. Based on the information, we run the Dijkstra's algorithm [7] to construct shortest path-based communication networks.

Based on the extracted network information, we first find out isolated UAVs or base stations within $2R$ range to connect via additional UAV deployment. In case that an isolated UAV is located together with other UAVs and base stations within $2R$ range, respectively, it chooses the closest base station to make a direct connection to the base station. If there is no base station to connect, but exist multiple non-isolated UAVs within the range, we choose one UAV among them that has the smallest number of hops toward its connected base station. Otherwise, if there exist only isolated UAVs to connect, we select the closest UAV.

After choosing candidates to connect, we explore where to deploy additional UAVs as relays. We create a virtual 3D cube where an isolated UAV, and its selected candidate are located at diagonal vertexes. Then, we search over all possible cells within the cube to find out one cell location that does not have an obstacle cell and not interrupted by any obstacle cells for direct communication link. We place an additional UAV at the chosen cell as relay. If this additionally deployed UAV becomes connected not only with the original isolated UAV, but with other isolated UAVs, we do not place more UAVs for the other UAVs since they have already get connected with luck. We continue this procedure until all the isolated UAVs get connected toward at least one base station.

C. Refinement

During the previous steps presented in Secs. IV-A and IV-B, we have constructed complete communication networks that connect all the UAVs to at least one base station while extending wireless coverage to all the 3D spaces in the RoI as illustrated in Fig. 5.

We perform the last optimization step to minimize the number of UAVs deployed for network construction, while preserving the almost full wireless coverage property. We want to differentiate some redundant UAVs from core ones that play key roles in connecting many other connection points.

We first check all closely located UAVs within $R/2$ range and pairs of UAVs that have at least 60% wireless coverage overlap. If a UAV has the smaller number of communication links, and does not create any network hole upon discarding it, we remove the UAV from our network construction. We continue to do this refinement process, and find the optimal number of UAVs and complete the whole network construction.

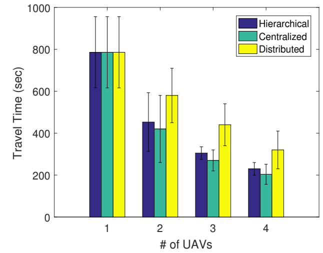
V. Evaluation

We validate our proposed algorithms in a simulated environment. Our 3D environment is located over 30×30 grid cell topology where communication range R is up to 10 cells, and the sub-area size is 10×10 grid size. We use the UAV's relative flying speed of 1 $cell/sec$. In our experiments, we have prepared 100 different basic building patterns and have randomly generated test datasets with different grid size and also randomly chosen height of the building.

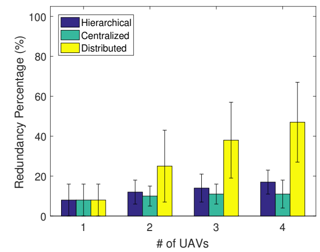
Our evaluation is divided into two parts: 1) collaborative exploration performance and 2) network construction performance. For collaborative exploration, we quantify total travel time for full 3D exploration as the farthest travel time among UAVs, and the average redundant visit rate among UAVs by varying the number of UAVs. For network construction, we measure total coverage rate based on how many cells are covered by UAVs.

A. Collaborative Exploration

To validate our path planning algorithm, we compare it against a naive scanning algorithm that follows the zigzag trajectory [5]. The naive scanning algorithm is considered as the basic motion planning method used mostly in agriculture, search and rescue, and mapping task using UAVs as in Fig. 2(a). We adapt this algorithm to fit into the 3D space with obstacles as shown in Fig. 3(a).



(a) Travel time performance



(b) Redundant cell-visit rate

Fig. 6. Travel time and redundant visit rate performance according to task division by varying the number of UAVs on 30×30 grid size test sets

We analyze exploration performance based on total 100 randomly generated test sets where 20 test sets for each different grid size from 10×10 to 50×50 are used. As in Fig. 7(a), our 3D path planning algorithm outperforms a naive scanning algorithm with a factor of up to 2. Both algorithms lead to full exploration over all the cells, and the constructed height topology through each exploration exactly matches the ground-truth in our test datasets.

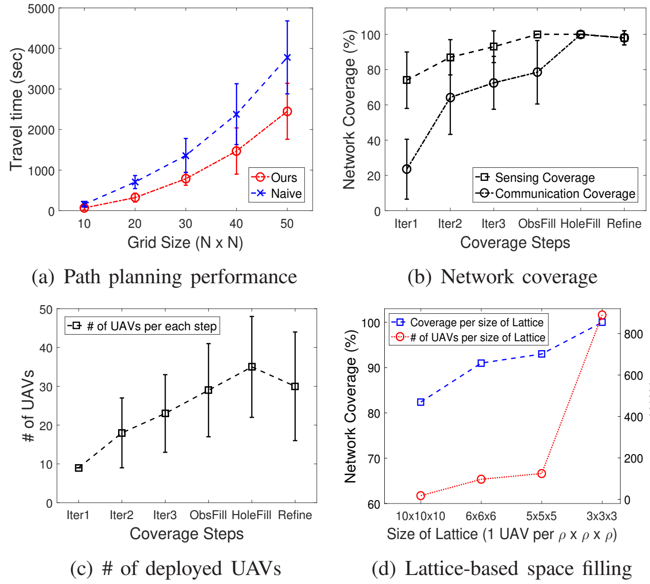
We investigate how task division improves exploration efficiency in terms of travel time and redundant visit rate. We compare three different task division mechanisms: 1) our hierarchical task division, 2) a centralized task division, and 3) a distributed task division. The centralized task division algorithm, considered as a theoretically optimal bound, divides the exploration space into pre-determined sub-areas of which each one is assigned to a responsible UAV. The distributed task division allocates unvisited cells using K-means clustering to the encountered UAVs within communication range after they exchange the visited cell information, without using any sub-area division.

We use 20 randomly generated test sets with 30×30 grid size and 9 sub-areas. As in Fig. 6, the centralized task division performs best in terms of travel time and redundant visit rate because the regions to explore with each UAV are pre-determined in an optimal way using a centralized manner. On the other hand, our hierarchical task division also performs well comparable to the centralized task division, whereas the distributed task division performs worst. This demonstrates that exploiting not a cell, but a sub-area as the minimal allocation unit and exploration based on it have indeed avoided possible redundant path overlap among UAVs.

B. Network Construction

We evaluate network construction performance in terms of network coverage rate and the number of deployed UAVs. We categorize network coverage in two aspects: 1) sensing coverage on how deployed UAVs extend their wireless coverage toward terrestrial areas, and 2) communication coverage on whether deployed UAVs can communicate each other toward at least one base station.

As in Fig. 7(b), our network construction algorithm keeps optimizing over iterations in terms of both sensing and communication coverage, reaching 100% coverage at the network hole filling step. Through the last refinement process, the



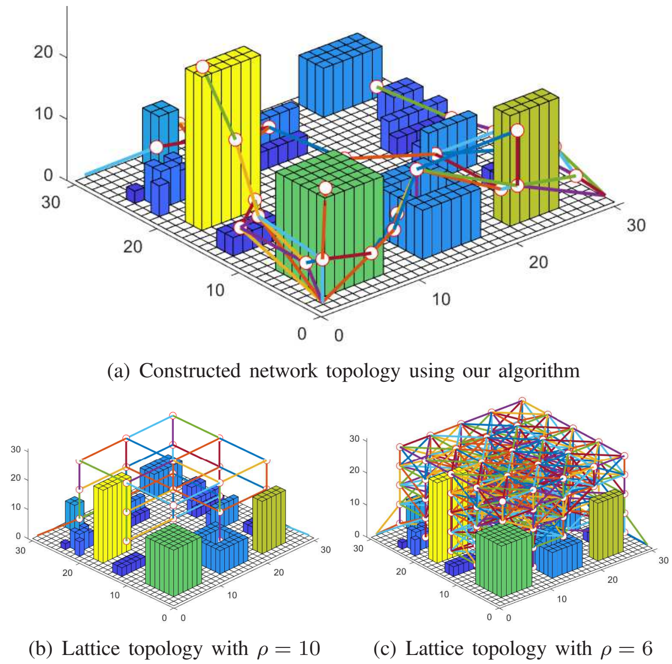
(a) Path planning performance

(b) Network coverage

(c) # of deployed UAVs

(d) Lattice-based space filling

Fig. 7. Path planning efficiency and network coverage performance



(a) Constructed network topology using our algorithm

(b) Lattice topology with $\rho = 10$

(c) Lattice topology with $\rho = 6$

Fig. 8. Network topology comparison between lattice-based space filling and our algorithm

coverage performance slightly decreases to 98% due to the removal of 4 less effective UAVs as in Fig. 7(c).

Lastly, we compare our network construction against a space filling approach in 3D space with obstacles after we adapt from [2]. We let one UAV deployed to one lattice cube consisting of $\rho \times \rho \times \rho$ cells. As UAVs are more densely deployed by decreasing the lattice size ρ from 10 to 3 in Fig. 7(d), both sensing and communication coverage have improved. To achieve full coverage around 100%, however, a tremendously large number of UAVs (i.e., 952) are required. Even for the coverage rate of 91%, 116 UAVs are required for the lattice-based space filling, whereas ours requires only 30 UAVs for the coverage of 98%. Fig. 8 shows the visualization of each obtained network topology.

VI. Conclusion

We have presented an optimal full-coverage 3D aerial ad-hoc network construction method using multiple UAVs in an urban environment. We have proposed ways to deploy the minimum number of UAVs to connect UAVs and base stations all together, while guaranteeing maximum network coverage. Prior to the network construction procedure, we have proposed an efficient technique for collaborative 3D terrestrial exploration using multiple UAVs based on distributed path planning. Our collaborative exploration algorithm finds efficient collision-free exploration paths in 3D space in a rapid manner by reducing redundant visit as much as possible via aerial task division.

Our experiment results show that our path planning algorithm for height mapping significantly outperforms a baseline naive scanning algorithm. Also, our collaborative exploration procedure using multiple UAVs finds an effective way to perform task division, while reducing the overall traversal time and the redundancy percentage. We have also demonstrated that our network construction technique guarantees almost full coverage toward terrestrial space with only minimal UAV usage.

For future work, to prepare some possible network failure and extend the current problem, we may design a more conservative wireless coverage by covering a specific location with at least k UAVs. Also, it would be interesting to reflect realistic problems such as battery outage and sensing failure during the mission to design a more adaptive network algorithm.

References

[1] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: research challenges. *Ad hoc networks*, 3(3):257–279, 2005.

[2] S. Alam and Z. J. Haas. Coverage and connectivity in three-dimensional networks. In *ACM MobiCom*, pages 346–357. ACM, 2006.

[3] H. M. Ammari and S. Das. A study of k-coverage and measures of connectivity in 3d wireless sensor networks. *IEEE Transactions on Computers*, 59(2):243–257, 2010.

[4] N. Gageik, P. Benz, and S. Montenegro. Obstacle detection and collision avoidance for a uav with complementary low-cost sensors. *IEEE Access*, 3:599–609, 2015.

[5] D. Jeong, S.-Y. Park, and H. Lee. DroneNet: Network reconstruction through sparse connectivity probing using distributed UAVs. In *IEEE PIMRC*, pages 1797–1802, 2015.

[6] D. Pompili, T. Melodia, and I. F. Akyildiz. Three-dimensional and two-dimensional deployment analysis for underwater acoustic sensor networks. *Ad Hoc Networks*, 7(4):778–790, 2009.

[7] S. Skiena. Dijkstras algorithm. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley*, pages 225–227, 1990.

[8] K. Yang and S. Sukkarieh. 3D smooth path planning for a UAV in cluttered natural environments. In *IEEE/RSJ Intelligent Robots and Systems (IROS)*, pages 794–800. IEEE, 2008.

[9] L. Yang, J. Qi, J. Xiao, and X. Yong. A literature review of uav 3d path planning. In *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*, pages 2376–2381. IEEE, 2014.

[10] C. Zhang, X. Bai, J. Teng, D. Xuan, and W. Jia. Constructing low-connectivity and full-coverage three dimensional sensor networks. *IEEE Journal on Selected Areas in Communications*, 28(7), 2010.