Fractal Trajectories for Online Non-Uniform Aerial Coverage

Seyed Abbas Sadat, Jens Wawerla and Richard Vaughan
Autonomy Lab, Simon Fraser University
{sas21, jwawerla, vaughan}@sfu.ca

***Abstract*— We propose a novel method for non-uniform terrain coverage using *Unmanned aerial vehicles (UAVs)*. The existing methods for coverage path planning consider a uniformly interesting target area and hence all the regions are covered with high resolution. However in many real world applications items of interest are not uniformly distributed but form patches of locally high interest. Therefore, with sparse sampling of uninteresting sections of the environment and high resolution sampling of interesting patches, the trajectory of the robot can become shorter. In this paper, we present a novel coverage strategy based on *Space-Filling Curves* that can accomplish non-uniform coverage of regions in the target area. Simulations and real robot experiments indicate that with the new strategy, travel time / cost of the task can be (almost) always less than a regular 'lawnmower' coverage pattern.**

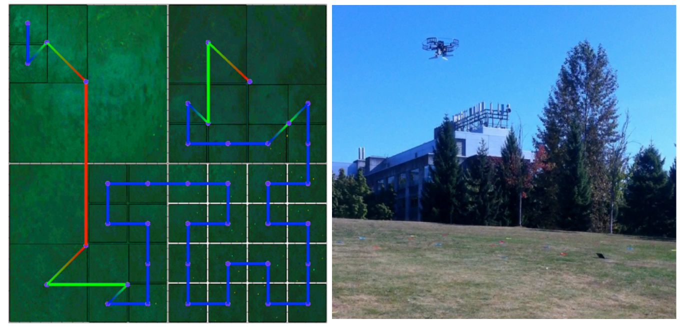


Fig. 1: Non-uniform coverage with real UAVs. Interesting regions (bottom right of the left figure) are covered with higher sensor resolutions than uninteresting area.

I. INTRODUCTION

Equipped with a light-weight camera, a UAV can be used to fly a path, while recording images, so that all the regions in the area of interest are covered. This coverage task is encountered in applications such as agriculture, surveillance, search and rescue, and vegetation monitoring. We recently proposed a method that uses a coverage tree structure and traversal strategies that accommodate non-uniform coverage of different sections of the environment and demonstrated its advantages over a regular uniform method [1]. If the environment is uniformly interesting, a coverage pattern of parallel stripes, i.e. the 'Lawnmower' pattern, provides optimal coverage. However, in many applications, as a result of non-uniformity in the environment, different parts of the target area can be covered with different resolutions, for example, by flying at different altitudes. This may allow the path planner to produce shorter paths due to the fact that the sensor footprint sweeps a bigger area as the distance between the sensor and the target surface increases [2]. In many real-world applications the distribution of interesting sub-areas is not known in advance. But we may be able to use the partial map collected online to classify a section as possibly interesting or uninteresting. At high altitude we can decide whether there is a need to cover a sub-region from closer viewpoints or not. Our previously proposed method was more efficient than Lawnmower when the percentage of interesting area was less than a threshold (about 50%) [1].

In this work, we propose a new online traversal of the coverage tree based on the Hilbert space filling curve (SFC) that improves the performance of the method through enhancing the local coverage plans and exploiting locality of the interesting regions. The new method out-performs all the previous strategies. It is also shown that with some distributions of interesting regions, our method generates shorter coverage paths compared to the regular Lawnmower pattern, independent of the total interesting area size. Simulation experiments and real robot experiments were used to compare the efficiency of the proposed approach.

II. RELATED WORK

UAVs have been used for aerial imaging in many projects. A single quad-rotor is used in [3] to cover an irregular area while satisfying some user-defined requirements (such as resolution and image overlapping). In [4] multiple UAVs are used to take georeferenced images of farm land. In order to create a full map of the area using image mosaicking techniques, grid-based coverage path planning was used to fully cover the target area [5]. Unmanned helicopters were used in [6] for automatic crop dusting. Simple back-and-forth motions are used to cover segments of the field after decomposition. A team of hex-rotors is used in [7] to take high quality images from farm fields by visiting a predefined set of waypoints. The images are then used by agricultural experts to locate weed pods. Carpin *et al.* have used a similar tree structure to solve a multi-target search problem using an aerial robot [8].

Spires *et al.* applied the Hilbert SFC to exhaustive geographic search with a swarm of robots [9]. Starting at random locations in the region, each robot follows the Hilbert curve and performs the sensing action at each grid cell. A robot has accomplished its task upon reaching the starting cell of another robot. However, in contrast to our approach, they only rely on a single order of Hilbert curve. SFCs have also been used to generate tool paths for CNC machines [10].

However in these applications, the regions of interest are known *a priori*.

Seabed coverage using an autonomous under-water vehicle (AUV), which resembles aerial coverage with UAVs, has also been studied in recent years. Galceran et. al. in [2], noticed that in lawnmower-like seafloor coverage, an AUV that stays at a constant depth, will lead to undesirable coverage overlap among the back-and-forth laps caused by the change in seafloor height. Their proposed method uses different inter-lap spacing in different regions to achieve shorter trajectory. Ho *et al.* evaluated various sampling path strategies for AUVs [11]. Although they assume no *a priori* knowledge of the underlying scalar field, the sampling paths are supposed to be static and hence different from our adaptive approach.

III. NON-UNIFORM COVERAGE

As mentioned earlier, we achieve non-uniform coverage by using the tree structure we proposed in [1]. In this tree each node covers a section of the area with a specific resolution. The resolution increases as one traverses down the tree so that a region covered by a parent node is completely covered by its children nodes with some higher resolution. The UAV can visit the tree nodes (based on the strategy presented in section III-B) which satisfy the resolution requirements, i.e. it needs to visit the leaf nodes (highest resolution) only in interesting regions. Therefore the robot travels in the environment visiting nodes at different depths of the tree in order to accomplish the coverage task. Here, in order to minimize the redundancy in coverage and hence reduce the overall trajectory length, we need a systematic ordering π of the nodes at each level so that two nodes appearing beside each other in π cover neighbour grid cells. This ordering is imposed on the nodes by the Hilbert curve as will be discussed in section III-B. We exploit the locality preserving feature of the Hilbert curve and introduce some opportunistic behaviours that improve the efficiency of coverage plans.

The next section reviews the coverage tree structure briefly. Then in section III-B we present our Hilbert-based coverage path planner in details.

A. COVERAGE TREE

Let us assume that the target area A is $m \times m$ meters and free of obstacles. Also, for simplicity, assume that the sensor direction is perpendicular to the ground and the shape of the sensor footprint is a square with length of $l(h)$ where h is the distance of the sensor to the ground. The coverage tree embedded in the metric space is recursively created as follows:

i The root of the tree R, is located at the center of A with a height of $h_R = l^{-1}(m)$ (where $l^{-1}(.)$ is the inverse of the function $l(.)$), i.e. the sensor footprint covers A at the root node.

ii Let h_n and A_n be the height of node n and the area covered by the sensor at node n respectively. Then, for a *branching factor* $b \geq 2$, A_n is decomposed into a $b \times b$ grid with cells of length $l_c = \frac{l(h_n)}{b}$ and therefore $h_c = l^{-1}(l_c)$. For some threshold h_t, if $h_c \geq h_t$ then there is a node for each grid cell, with node n as parent, at the center of the cell and with height h_c, i.e. the sensor covers the grid cell at the child node (see Figure 4a).

Note that no nodes with a height less than a threshold h_t exist in the tree. The parameter h_t determines the lowest height at which the sensor can sufficiently cover a region at the finest resolution. Therefore regular path planners for area coverage use this height to produce a lawnmower pattern (assuming no obstacles in the area) to sweep the whole area. This simple pattern is equivalent to visiting all the leaf nodes of the coverage tree in an order that yields the minimum path length. In this paper, we assume that we know h_t in advance and hence do not create nodes below it. However, if the sufficient sensing resolution can only be determined online, e.g. based on the patch type, the coverage tree can be created incrementally as the area is explored.

Here it is assumed that if the sensor footprint at node n is decomposed into a $b \times \ b$ grid, we are able to classify each cell (child node of n) as uninteresting or possibly interesting. In [1] we proposed the following strategies to traverse a coverage tree:

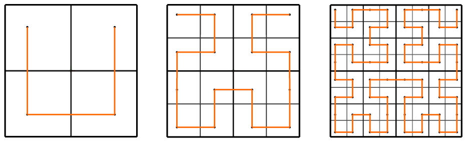
- **Depth-First (DF):** In the DF strategy the UAV starts performing the lawnmower trajectory at depth 1 of the tree. When visiting each node, the robot descends to the next depth to visit all interesting children (if any). This behaviour is repeated recursively upon visiting each child node.
- **Shortcut Heuristic (SH):** This strategy is the same as DF with one difference: Assume the UAV has visited a node and the next node to visit, n_{next}, is located at some height above the current node. According to DF the UAV flies directly to n_{next}. However in SH, the UAV visits the nearest descendant ($n_{nearest}$) of n_{next}. Now there are two possible situations: $n_{nearest}$ was interesting or it was uninteresting. In case it was interesting, we recursively visit all other unvisited children of n_{next}. Note that we do not have to visit n_{next}. Alternatively if the child was uninteresting, the UAV visits the parent of $n_{nearest}$.

For more detailed discussion on the described traversal strategies and the coverage tree and possible ways to relax some of the assumptions see [1].

B. HILBERT-BASED COVERAGE PATH PLANNING

The Hilbert curve, H, is a fractal space-filling curve shown in Figure 2. The Hilbert curve of Nth order, denoted by H_N, fills a grid of $2^N \times 2^N$. It is a useful curve since it provides a mapping between 1D and 2D (nD) spaces which preserves locality adequately well [12]. Informally speaking it means that a robot following a Hilbert trajectory stays close to the recent places that it has visited. This property exists in the Hilbert curve of any order.

We use the Hilbert curve to impose an ordering on the nodes at each level of the coverage tree. Note that H_1 provides an ordering on the 2×2 grid which corresponds to the nodes at depth 1 of the coverage tree when $b = 2$ (see



(a) 1st order (b) 2nd order (c) 3rd order

Fig. 2: Hilbert curves of 1st, 2nd and 3rd order.

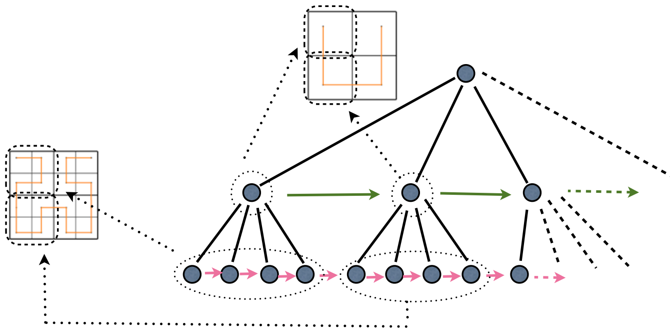


Fig. 3: Coverage tree with Hilbert-based ordering of nodes at each depth and its relationship to grids in different resolutions

Figure 4a, 4b) [1]. Similarly, H_2 and H_3 impose an ordering on nodes at depth 2 and 3 of the coverage tree (see Figure 4c, 4d). More generally, if $b = 2^s$, $H_{d \times s}$ can be used to sequentialize the nodes of depth d. Therefore a coverage tree can be seen as a tree with 1-D ordered nodes at each level as shown in Figure 3. Our approach to non-uniform coverage is to visit the nodes based on H. Upon visiting an interesting node, we increase the coverage resolution by descending to the next depth or stay in the same depth of the tree. If the visited node was uninteresting, the resolution of the coverage is decreased by going one level up (decrease depth) in the tree. In either case, we continue to visit the nodes based on the appropriate Hilbert ordering. The intuition is that, when an interesting region is observed, one can opportunistically assume that the next node will also be interesting due to the locality preserving feature of the Hilbert curve, i.e. if the interesting region forms a big patch, with high chance, the next node will remain in the patch and the overhead of visiting the parent nodes is avoided.

Our proposed traversal strategy is formally presented in Algorithm 1. It returns the next node that should be visited by the UAV according to the current state of the tree.

In this algorithm, $Children(n)$ returns the children of node n and $Parent(n)$ returns the parent of node n. $Depth(n)$ returns the depth of the tree at which node n exists. The function $NeedVisit(n)$ returns $true$ unless visiting n is not necessary (note that $NeedVisit(null) = false$ and $NeedVisit(leaf) = false$ if $leaf$ has already been visited or classified as not interesting). This is the case when there is an ancestor node of n that is classified as uninteresting. Also,

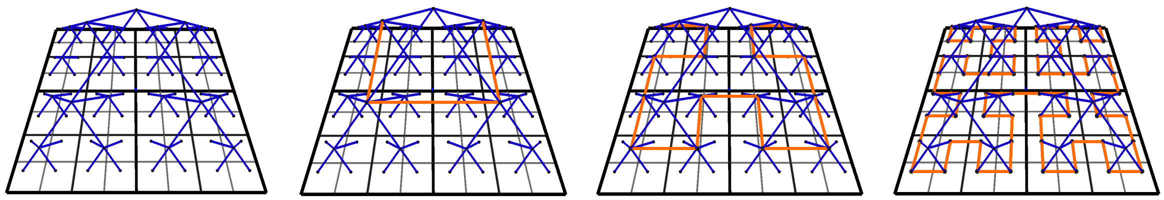
[1]We define the top left end of the curve as start and the top right end as the end of the curve.

Algorithm 1 Hilbert-based coverage path planning

1: **Input**: Last visited node n ($null$ if none), coverage tree T with the implicit order of the nodes.
2: **Output**: Next node that should be visited or $null$ if the traversal is finished.
3: **if** $n = null$ **then**
    return the first leaf of T
4: **end if**
5: **if** $Interesting(n)$ AND $NeedVisit(Children(n))$ **then**
6:   $n \leftarrow$ first child of n
7: **else if** $!Interesting(n)$ AND $Depth(n) > 1$ **then**
8:   $n \leftarrow Parent(n)$
9: **end if**
10: **if** $NeedVisit(n)$ **then**
    return n
11: **else if** n is the last child of $Parent(n)$ **then**
12:   $n \leftarrow Parent(n)$
13:   Go to 5
14: **else**
15:   **if** $Next(n) <> null$ **then**
16:     $n \leftarrow Next(n)$
17:     Go to 10
18:   **else**
    return $null$
19:   **end if**
20: **end if**

when none of the children of node n need to be visited then $NeedVisit(n) = false$. This function is required to prune out nodes that all their interesting descendant leaves have been visited. Additionally, $Next(n)$ returns the next node of the tree (or $null$ if n is the last node) at the same depth of n in the order imposed by $H_{Depth(n)}$. Finally, first (last) child of a node is defined as the child node that appears first (last) in the corresponding Hilbert ordering.

Using this algorithm, the traversal starts by visiting the leaf node on the top left corner of the environment (lines 3-4). Upon visiting a node n, if it is interesting and at least one of its children needs to be visited, it visits the interesting children according to H (lines 5-6). In case the node is uninteresting, it ascends to the parent node provided that n is not at the maximum height (lines 7-8). Then, lines 10-20 make sure that we prune the nodes that do not need to be visited. Here, as the nodes are being skipped, if the last child(ren) of a node is pruned (because it was classified as uninteresting when visiting the parent node) we go one level up to adhere to the general strategy. Figure 5 shows such a situation. At the top left, the 2×2 red square, s is pruned when the UAV visits the node marked by the triangle. Since s is the last child of the triangle node (by the Hilbert ordering), after covering the rest of the children, the UAV skips s and goes one level up in the tree and hence takes the route indicated by circles (which is a jump from depth 4 to depth 2). Note that we are always progressing in the Hilbert



(a) Coverage tree (b) H_1 (c) H_2 (d) H_3

Fig. 4: Coverage tree and Hilbert curve. H_N is used to impose ordering on nodes at depth N.

curves at any depth and we visit each node at most once. Given that we assume a maximum depth for the coverage tree, our algorithm terminates.

Let us assume that the number of grid cells at the finest resolution (leaves of the coverage tree) is k. The creation of the tree at the start of the mission takes $O(k^2)$. The memory space of each node in the tree is constant which leads to $O(k^2)$ total memory consumption. At each waypoint, the quadrants of the sensed image are classified as being interesting or uninteresting ($O(1)$). Then, Algorithm 1 is executed, iterating over some nodes to find the next goal waypoint. At each iteration of the loop $NeedVisit(n)$ is the most expensive operation. It needs traversing to an ancestor node ($O(log(k))$) or calling $NeedVisit()$ on each child ($O(k^2)$). Note that traversing up to find an uninteresting ancestor is executed only in the first call of $NeedVisit(n)$ and not in the recursive calls. This is because $NeedVisit()$ checks all the descendants from top of the subtree to the bottom and consequently the uninteresting ancestor is found either in the first call of $NeedVisit(n)$ or in the recursive calls before its descendants. The loop runs at most for $O(log(k))$ since at each iteration we go to the parent node (either directly or after iterating over all the 4 children) or have started a sequence of descending moves which ends in an unvisited interesting leaf node. Therefore the computational cost of the planning at each waypoint is in the order of $O(k^2 log(k))$ (which is $O(Klog(K^{1/2}))$ where K is the number of nodes in the tree). Note that this complexity is the worst case scenario in an abstract situation. In a concrete case, the loop runs very few times to find the next goal.

IV. EXPERIMENTS AND RESULTS

In the following sections we discuss the results of the simulations and field trials.

A. SIMULATIONS

In these experiments, a 128×128 m^2 area, free of obstacles, is considered. The simulated UAV is equipped with a sensor pointing downwards with $l(h) = h$, i.e., a field-of-view of $90°$. In order to generate many different configurations of interesting patches, we introduce two parameters to represent a distribution of interesting regions: p, the percentage of the whole area that is interesting and c, the number of interesting segments. We assume that the interesting regions are rectangles (of any form) but have the same size. For instance, $(40, 3)$ refers to all environment configurations in which 40% of the whole area is interesting

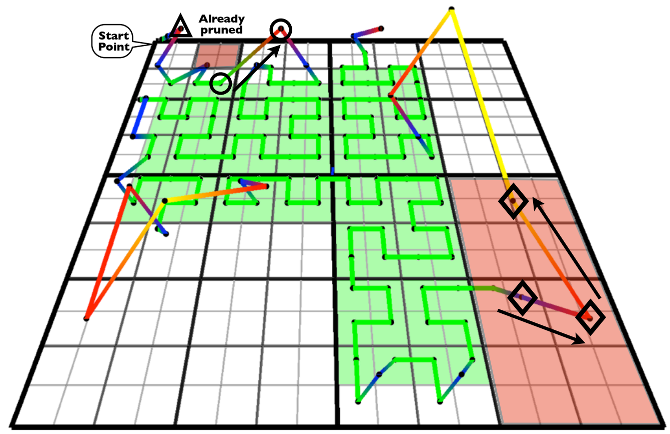


Fig. 5: Final coverage plan in simulation. The color of the trajectory changes with height. Red squares on bottom-right are pruned when the UAV ascends to visit the nodes indicated by diamonds. (best seen on screen)

and in form of 3 rectangles with equal area. For each pair of (p, c), $p \in \{10, .., 90\}$ and $c \in \{1, .., 4\}$, 10 random environments are generated and the four strategies (H, DF, SH, lawnmower) are used to cover the area. For the traversal strategies we use $b = 2$, $l_{max} = 1$ and coverage tree with maximum depth of 5. For each strategy, the average of the total distance that the robot travelled is used as a performance metric. The results are shown in Figure 6. In all experiments the variance was below 192 m (the error bars are very small and difficult to see in the figures).

The graphs indicate that our new Hilbert based traversal of the coverage tree outperforms Depth-First and Shortcut strategies in all the tested configurations of the environment. Moreover, the cost of the Lawnmower pattern which is independent of interestingness, is almost always higher than our approach. Consequently, if the estimated distribution of interesting regions falls in the wide range of the tested configurations, using our method, one can perform the coverage task in shorter time compared to Lawnmower. Note that we have also computed the lower-bound of the trajectory length which is obtained when the interesting regions are fully known and covered by Lawnmower pattern. Comparing the Hilbert strategy with the lower-bound, we can see that our Hilbert coverage strategy is performing well considering that it is not relying on any *a priori* information.

In the Hilbert-based traversal of nodes at a specific depth, the UAV flies from one node to the other while observing

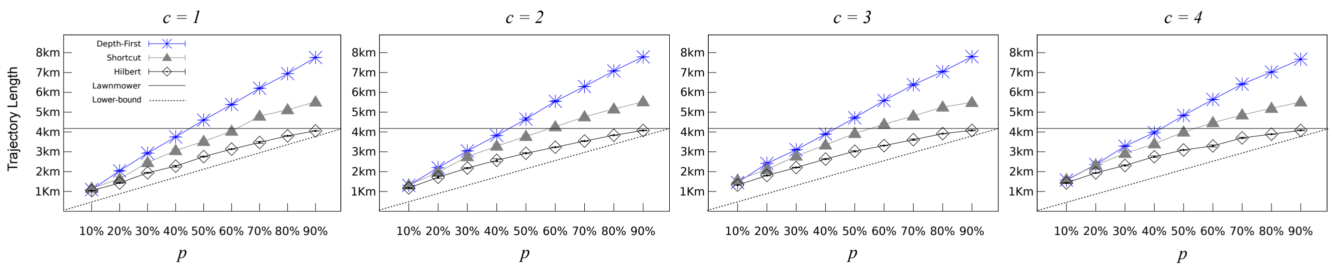


Fig. 6: Results of the simulation experiments in different environment configurations (p : ratio of the whole area that is interesting, c : number of patches) (See IV-A).

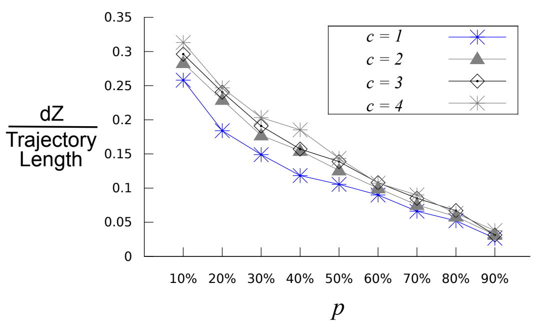


Fig. 7: The ratio of total displacement along Z-axis to the total length of the trajectory in environments with different number of patches.

grid cell neighbouring criteria, i.e. it moves from a grid cell to one of its 4 (2 or 3 if on the grid boundary) neighbours. This means that the local coverage paths at one tree level is optimal in terms of path length. Similarly, in the sequence of nodes that Algorithm 1 generates, when the UAV ascends from node n to n', n' is either the parent of n or a sibling of an ancestor of n. Therefore the neighbouring criteria is held at all the levels of the tree. On the contrary, in previously proposed strategies i.e. Depth-First and Shortcut, no such criteria was imposed and a constant ordering among the children nodes was used which was not optimal at different levels. The other property of our new approach is that it exploits the patchiness of interesting regions by travelling between nodes while preserving locality. Both Shortcut and Hilbert-based traversal opportunistically visit an unclassified neighbour of the current interesting grid cell to avoid visiting nodes at higher levels while still being in that interesting section of the environment. However, the Hilbert curve exhibits high locality and as a result the UAV exits the interesting regions very few times, compared to greedy Shortcut. Figure 5 demonstrates how Hilbert curve covers two interesting patches with few crossings of the boundary of the patches.

Figure 7 depicts the ratio of mean motion along Z-axis to the mean of trajectory length. It implies that with low total interesting area, the UAV tends to visit nodes at lower depth of the tree (high altitude). This is due to the fact that uninteresting regions cause the UAV to ascend in the tree, and those sections are consequently sampled sparsely (by pruning out all their uninteresting children nodes). In cases where most of the area is interesting the vehicle stays at leaf level as is suggested by the graph. Moreover, with a single interesting patch in the area it will be more likely that a node at high altitude (low tree depth) is uninteresting compared to the situation where multiple patches (with the same total interesting area as one patch) exist. Hence, in the former situation larger uninteresting segments can be pruned out at once with a series of successive single ascending.

B. FIELD TRIALS

We also performed some experiments in outdoor environment using a Pelican quadrotor from Ascending Technologies. GPS is used to estimate the robot position which is then fed into a PID controller to generate the required velocity commands. The task was to cover an area of 30×30 m^2. A number of frisbees were spread out in the area forming a single patch that occupied about 31% of the target environment as shown in Figure 9 with a yellow polygon. Using a gimbal mounted colored camera with 90° field-of-view, we detect frisbees from different heights (Figure 9). A query region in an image with non-empty intersection with a frisbee is classified as interesting. Due to light changes and other sources of noise, there are false positives in frisbee detection. The experiment was repeated 6 times performing Lawnmower-based coverage and 8 times with our proposed method. The results are shown in Figure 8. In this figure, we report the length of the UAV trajectory along with the overall ratio of perceived interesting cells, i.e. false positives plus true positives.

The mean and variance of the Lawnmower performance are shown in pink. The shaded part indicates the possible performance of Lawnmower with different interestingness ratio of the environment. The variance of the Lawnmower trajectory indicates the amount of noise in the position estimation and control of the robot. Despite that, it can be observed that for situations where the perceived interestingness is below 50% our method generates a shorter trajectory, except in one trial (indicated by a dashed circle). In that trial false positive interesting middle nodes which were apart from each other led to longer trajectory.

Note that as the width m (length) of the target area increases, the length of the Lawnmower coverage pattern grows with $O(m^2)$. On the other hand with our approach, the

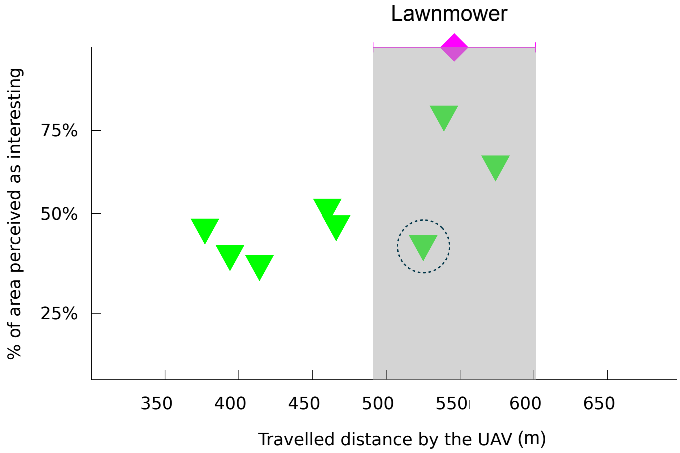


Fig. 8: Results of the experiments with real UAV. The performance of the Lawnmower pattern is shown in terms of mean and variance of 6 trials. The performance of our method in each trial is reported individually.

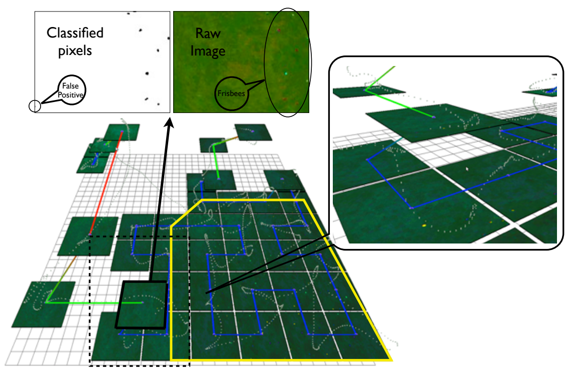


Fig. 9: The trajectory of the UAV. The yellow line approximately shows the true interesting area (best seen on screen).

length of a trajectory that ascends up to eliminate (classify as uninteresting) a node N will be of $O(m')$ where m' is the width (length) of $A_{parent(N)}$ (sensed area at the parent node of N). This can be seen on the bottom right of the Figure 5 where, traveling from the leaf node to its ancestor at depth 1, prunes out (at least) $\frac{1}{4}$ of $A_{parent(N)}$. Therefore as the size of the area grows the performance benefits of traveling to higher nodes will be larger and the difference between Lawnmower and Hilbert-based coverage becomes more clear. Due to lack of suitable flying sites, we could not perform the outdoor experiments in a larger environment and left it for future work.

V. LIMITATIONS AND FUTURE WORK

In future, we will consider different patch types with various levels of interestingness. Furthermore, we will use probabilistic sensor models to deal with false positives and false negatives in the patch classification. Also, the assumption that the UAV should stop at each waypoint in order to perform the sensing might not hold in some situations. In that case, Lawnmower pattern has the advantage of less acceleration than our method. Moreover, although the amount of height changes in our method is low compared to the total trajectory, the cost of ascending and descending might be different than the cost of movement in the x-y plane. Considering different energy consumption models for these movements can be interesting future research.

VI. CONCLUSION

We proposed a novel method for online non-uniform coverage of unknown terrains based on Hilbert space filling curve. Our method can be used to provide aerial coverage in larger areas compared to a regular Lawnmower pattern. It achieves this by spending its battery charge to fly over interesting regions and only sparsely sampling uninteresting sections. Simulations show the method is effective in patchy environments, and real-world trials verify that the benefits can be seen even with imperfect sensing and control.

ACKNOWLEDGMENT

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) through the NSERC Canadian Field Robotics Network (NCFRN).

REFERENCES

[1] S. A. Sadat, J. Wawerla, and R. T. Vaughan, “Recursive non-uniform coverage of unknown terrains for uavs,” in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS'14)*, (Chicago, USA), September 2014.

[2] E. Galceran and M. Carreras, “Efficient seabed coverage path planning for asvs and auvs,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 88–93, Oct 2012.

[3] J. Valente, D. Sanz, J. Del Cerro, A. Barrientos, and M. de Frutos, “Near-optimal coverage trajectories for image mosaicing using a mini quad-rotor over irregular-shaped fields,” *Precision Agriculture*, vol. 14, no. 1, pp. 115–132, 2013.

[4] A. Barrientos, J. Colorado, J. d. Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente, “Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots,” *Journal of Field Robotics*, vol. 28, no. 5, pp. 667–689, 2011.

[5] J. Valente, A. Barrientos, J. Cerro, C. Rossi, J. Colorado, D. Sanz, and M. Garzn, “Multi-robot visual coverage path planning: Geometrical metamorphosis of the workspace through raster graphics based approaches,” in *Computational Science and Its Applications - ICCSA 2011*, vol. 6784 of *Lecture Notes in Computer Science*, pp. 58–73, Springer Berlin Heidelberg, 2011.

[6] S.-W. Moon and D. H.-C. Shim, “Study on path planning algorithms for unmanned agricultural helicopters in complex environment,” *International Journal of Aeronautical and Space Sciences*, vol. 10, no. 2, pp. 1–11, 2009.

[7] J. Cerro, A. Barrientos, D. Sanz, and J. Valente, “Aerial fleet in rhea project: A high vantage point contributions to robot 2013,” in *ROBOT2013: First Iberian Robotics Conference* (M. A. Armada, A. Sanfeliu, and M. Ferre, eds.), vol. 252, pp. 457–468, 2014.

[8] S. Carpin, D. Burch, and T. H. Chung, “Searching for multiple targets using Probabilistic Quadtrees,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4536–4543, Sept. 2011.

[9] S. V. Spires and S. Y. Goldsmith, “Exhaustive geographic search with mobile robots along space-filling curves,” in *Collective robotics*, pp. 1–12, Springer, 1998.

[10] J. J. Cox, Y. Takezaki, H. R. Ferguson, K. E. Kohkonen, and E. L. Mulkay, “Space-filling curves in tool-path applications,” *Computer-Aided Design*, vol. 26, no. 3, pp. 215–224, 1994.

[11] C. Ho, A. Mora, and S. Saripalli, “An evaluation of sampling path strategies for an autonomous underwater vehicle,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5328–5333, IEEE, 2012.

[12] H. Dai and H. Su, “On the locality properties of space-filling curves,” in *Algorithms and Computation* (T. Ibaraki, N. Katoh, and H. Ono, eds.), vol. 2906 of *Lecture Notes in Computer Science*, pp. 385–394, Springer Berlin Heidelberg, 2003.