Understanding Greediness in Map-Predictive Exploration Planning

Ludvig Ericson, Daniel Duberg, Patric Jensfelt

***Abstract*— In map-predictive exploration planning, the aim is to exploit a-priori map information to improve planning for exploration in otherwise unknown environments. The use of map predictions in exploration planning leads to exacerbated greediness, as map predictions allow the planner to defer exploring parts of the environment that have low value, e.g., unfinished corners. This behavior is undesirable, as it leaves holes in the explored space by design. To this end, we propose a scoring function based on inverse covisibility that rewards visiting these low-value parts, resulting in a more cohesive exploration process, and preventing excessive greediness in a map-predictive setting. We examine the behavior of a non-greedy map-predictive planner in a bare-bones simulator, and answer two principal questions: a) how far beyond explored space should a map predictor predict to aid exploration, i.e., is more better; and b) does shortest-path search as the basis for planning, a popular choice, cause greediness. Finally, we show that by thresholding covisibility, the user can trade-off greediness for improved early exploration performance.**

I. Introduction

Exploration is an essential capability in any truly autonomous robot dealing with an unknown or changing environment. It is particularly useful in radio-denied scenarios where time is of the essence, such as search-and-rescue in disaster zones [1]. The typical metric used is coverage over time or distance traveled [2], [3], though some recent work focuses on reconstruction error [4].

Although exploration implies an unknown environment, it is exceedingly rare that no prior information is available. For example, in man-made indoor environments, there is plenty of structure and common knowledge that a system should be able to exploit [5]. In spite of this, few state-of-the-art autonomous exploration methods make use of this.

Suppose a robot is tasked with exploring an indoor environment and is placed in a small room near a doorway to a long corridor. If the robot decides to explore the corridor first, it will have to return back to the room once it has reached the end of the corridor, thus resulting in unnecessary travel. Ideally, the robot would estimate what the rooms look like before they have been explored, and decide its course of action based on its estimation. This is the aim in *map-predictive exploration planning*.

Map prediction can be understood in multiple ways, from nearest-neighbor search in a patch-based database, to using a deep learning-based generative model to allow some degree of spatial intuition. In this paper, "predictive" is not restricted to the statistical sense of the word, but more generally "ahead of time". Predictions could be made from a longer-range sensor, e.g., using a LiDAR to plan the path of an RGB-D camera. If a partial or outdated map is available, as in multi-agent exploration or search-and-rescue in a collapsed building, prediction could mean using that knowledge.

Paper accepted for the 10th European Conference on Mobile Robots (ECMR 2021). All authors are with the Division of Robotics, Perception and Learning at KTH Royal Institute of Technology, Stockholm, SE-10044, Sweden. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP), the SSF project FACT and the VR grant XPLORE3D. For e-mail correspondence, contact ludv@kth.se.
978-1-6654-1213-1/21/$31.00

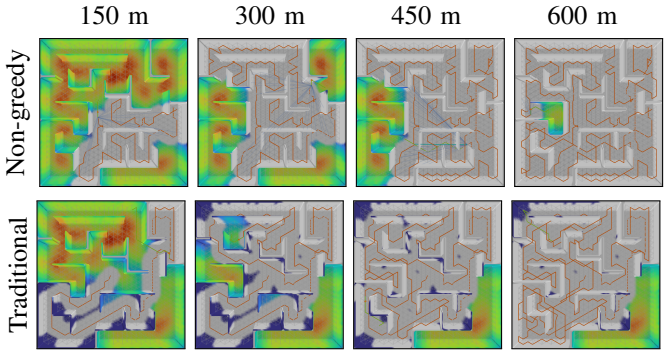

Fig. 1. An illustration of how traditional vs. non-greedy exploration planner performs in a map-predictive scenario. A snapshot is taken every 150 m of travel. Top: using a non-greedy planner, bottom: a variant using a traditional information gain formulation. Color shows the value of a proposed non-greedy scoring function that assigns high value in difficult-to-see regions. Red is lowest, dark blue is highest, and gray is already explored space. For the traditional planner, color illustrates what value the non-greedy scoring function would have assigned if it had been used. The traditional planner skips small regions early on which it must then revisit, allowing the non-greedy planner to finish earlier, after 620 m, compared to 1 067 m.

Recent work on map-predictive exploration [6], [7] use map predictions to improve estimation of a utility function, and demonstrate improved performance. However, we find that because the estimation is more accurate, it has caused the exploration to become more greedy, maximizing short-term utility. In [8], up to 71 % of the total time is spent on covering the last 10 % of space. We find this phenomenon in most state-of-the-art exploration methods.

In light of these findings, we propose that non-greedy exploration is key to truly unlocking map-predictive capabilities for exploration. In this paper, a minimal model of autonomous exploration planning is defined, in order to study the causes of greediness in map-predictive settings. To this end, we substitute prediction with an oracle, giving the exploration planner knowledge of the environment a-priori. The contributions are:

1) We propose a scoring strategy that removes greediness by weighting the utility function by *inverse covisibility*, demonstrated in Fig. 1.

2) We investigate the use of an exhaustive local search as the basis for exploration planning, as RRT-based planning may cause greediness in map-predictive settings.
3) We investigate the impact of a *prediction horizon* on greediness, which is the distance beyond explored space that is known in advance.

The software used to produce the results of this paper is available at `https://git.lericson.se/fragmenot.git`.

II. Background & Preliminaries

Generally, the goal of *autonomous exploration* is typically to explore some real-world environment in the shortest time possible. In its simplest form, it is a cycle of the following:

1) plan where to go by some heuristic,
2) execute the selected plan, and
3) integrate sensor readings into a map.

The cycle continues until the map is sufficiently complete, or a predefined time limit is reached.

In *frontier-based exploration* such as [6], [7], [9]–[11], the planner selects which frontier to visit according to some utility function. A frontier is defined as a boundary between free and unexplored space. Another common formulation is *sampling-based exploration* such as [2], [4]. A planning tree T rooted at the current state is constructed in an RRT-like fashion [12], [13]. Specifically, T is iteratively constructed by sampling a point p, moving it to some maximum distance from the nearest node p^* in T, and adding the edge $p^* \to p$ to T. Once the planning tree is built, the best path is chosen according to some score function over tree nodes.

The score functions can be said to consist of two parts: gain g, and cost c. The gain is formulated as an integral over an uncovered surface or volume V

$$g(V) = \int_V \rho \, dV \tag{1}$$

where dV is a surface or volume element of V as appropriate, and ρ is a *value density function*. V is typically discretized, turning the integral into a sum. The cost is normally based on distance or trajectory execution time.

In [2], a sampling-based receding horizon approach is presented where a single step is taken before replanning. The planning tree can fail to reach unexplored space when the environment is difficult to navigate, or the nearest unexplored space is far away. AEP [3] solves this by a falling back to a frontier-based strategy when the information gain for the best sampled path is insufficient. Like [2], the AEP gain is volumetric. The method is evaluated both in simulation and on a drone with coverage as the metric. In [4] a sampling-based method is presented where the gain function is designed to minimize the reconstruction error. The authors also propose to normalize the score function by computing the sum gain per cost over each path in the planning tree in *global normalization*. Works [2]–[4] primarily rely on traditional information gain, i.e., $\rho = 1$.

In terms of map prediction, [5] presents a database of CAD models and is used to make predictions for topological maps. [14] present a map-predictive approach where the perspective is one of motion planning in the presence of unknown obstacles. Their contribution is improving maneuvering and safety in unknown environments.

A map-predictive exploration method is presented in [7]. Generative image inpainting based on deep-learning is used to predict a 2D grid map, providing better estimates of the gain by inferring V in Eq. (1). They use a standard frontier-based method and demonstrate using simulation that map predictions allow the robot to perform efficient coarse exploration.

In [6], a frontier-based method is presented that investigates how different types of prior knowledge and predictions can be used for exploration planning. Predictions of room layouts are made from a partial 2D grid map using line detection heuristics. The authors note that the better their prediction is, the more it leaves fragments of unexplored space behind. This result is consistent with [7], where the greediness results in incomplete coverage.

In [15], map-predictive exploration is cast as offline exploration planning with simple simulation in 2D grid maps. Exploration planning is accomplished by a frontier-based A* planner, including the exploration state as part of the search tree, which is not feasible beyond relatively small 2D grid maps. Notably, [15] explicitly prevent the planner from visiting "small clusters" in the interest of efficiency, which results in incomplete coverage even for the simple maps presented.

We observe that in each of [2]–[4], [6], [7], [15], the system quickly gets to 85–95 % coverage, then spends most of its time on the last 5–15 %, if at all covered. This finding is the key motivation to our study of greediness in map-predictive exploration. We take a similar approach to [15], using a simplified model of exploration as the basis of investigation.

III. Model

Recent work in autonomous exploration typically relies on high-fidelity simulators for evaluation. In this work, we study the theoretical aspects of map-predictive exploration. A simple simulator is therefore not only sufficient, but to be preferred, as it allows us to eliminate extraneous error sources, such as sensor noise, complex motion models, and heavy simulation environments.

A. World Representation

We represent occupied space as a mesh of K triangular faces, denoted $S_\Omega = \{f_1, f_2, \ldots, f_K\}$, and explored space as a subset $S \subseteq S_\Omega$ of those K faces. This lets us define completion exactly as the percentage of area seen, and have 100 % explored if and only if $S = S_\Omega$. A face $f \in S$ is *seen*, and a face $f \notin S$ is *unseen*.

This representation is efficient enough to also enable us to do sensor integration in the planning step, and to thus determine the exploration state (see Eq. (2)) along every path inside the planning algorithm.

Motion is simplified by being confined to a precomputed roadmap, represented as an undirected graph $G = (V, E)$

with the states $V \subseteq \mathbb{R}^3$. The roadmap is distributed as a connected lattice, so that each edge is of equal length, and there is a path between every pair of states. The motion model is deterministic. The *visible faces* $S_{\{u,v\}} \subseteq S_\Omega$ along the edge $\{u, v\} \in E$ are the set of unoccluded faces within sensor range between u and v.

The exploration state is thus parameterized by the tuple (S, v) with $S \subseteq S_\Omega$ the set of seen faces, and $v \in V$ the current state. Its successor function is

$$(S, v) \leftarrow \left(S \cup S_{\{v,v'\}}, v'\right) \tag{2}$$

where $v' \in V$ is the successor state.

In summary, our world representation differs from the typical exploration literature in these regards:

1) The state space is essentially finite, defined by V, and we can therefore plan by a graph search on G.
2) Occupied space is represented as a triangular mesh, and explored space is a subset of occupied space, hence eliminating reconstruction errors.
3) While many exploration methods "double count" uncovered space when there is sensor overlap in the planned path, we do not, since the same state successor function is used in planning and execution.

These deviations allow us to isolate the planning part of exploration planning and also to close the gap between the plan and the execution in terms of using sensor information.

B. Path Planning with Map Predictions

RRT-based planners are not a well-motivated choice for map-predictive exploration planning, since they compute the shortest path which is not generally desirable in unexplored space. To illustrate this point, suppose a perfect maze is to be explored in a map-predictive setting, starting at the maze entrance. A perfect maze is equivalent to a tree where junction points are tree nodes, and the tree leaves are dead ends. The optimal exploration path is then equivalent to a depth-first search of the tree. An RRT-based planner, however, would tend to select "the longest shortest path", i.e., the path to the furthest-away leaf node in the maze tree, since such a path explores the most space. This phenomenon is visible in Fig. 1, where the traditional planner first explores the bottom left corner, then the top right, then the top left, and so on.

We therefore compare two path planners, which we call *backtracking search* and *shortest-path search*. Shortest-path search models RRT-based path planning and is equivalent to a perfect RRT-based path planner, as it computes the actual shortest path in G.

In backtracking search, an exhaustive local search on G is performed by generating paths via breadth-first search. It revisits the same state multiple times, allowing backtracking paths such as $v_1 - v_2 - v_1 - \cdots$ to form. Since the set of possible paths grows exponentially with path length, the search is limited to the first N paths so generated.

Temporary *jump edges* are inserted into the roadmap G from the current state to each state in the frontier set $F \subseteq V$ at the start of path planning. A state is in F if and only if it has at least one seen and one unseen face along its edges. A jump edge is equivalent to traversing the shortest path over already-explored edges, so all jump edges have $S_{\{u,v\}} \subseteq S$ by definition, and consequently zero gain. Jump edges are approximately equivalent to starting a sub-search from each state in F. Jump edges also guarantee that the planner finds unseen faces, since the distance to at least one unseen face is two edges at most.

C. Prediction Horizon

Map prediction will realistically only be able to accurately predict at some maximum distance ahead of explored space. We call this distance the *prediction horizon*. Such a horizon is sure to impact exploration performance, and we therefore impose a horizon artificially by cropping the environment at d meters from explored space.

Specifically, we remove states in the roadmap G that are more than d meters plus the sensor range away from the nearest visited state. This means that planning is restricted to explored space plus d meters beyond.

D. Evaluating Greediness

Greediness can be quantified by the surface-to-volume ratio of unexplored space. It is preferable to have the remaining unexplored space concentrated in one dense area, as opposed to having small disconnected clusters spread out over the entire environment. The latter is a consequence of greedy exploration, so a non-greedy planner should have a lower surface-to-volume ratio of its unexplored space. We therefore evaluate greediness by the *isoperimetric ratio* (IPR). For the unexplored faces $U = S_\Omega \setminus S$,

$$\mathrm{IPR}(U) = L_U^2 A_U^{-1} \tag{3}$$

where L_U, A_U is the perimeter and area of U, respectively. IPR is a convenient metric, as it is dimensionless, and invariant to similarity transforms such as rotation or scaling.

IV. Non-Greedy Exploration

Map predictions lead to ballooning gain in the predicted-but-unexplored space. The consequence is that the planner prefers moving towards these high-gain regions, leaving comparatively low-gain fragments and slivers of unexplored space behind. Eventually, these low-gain fragments are all that remain, and the robot must travel long distances to revisit areas it has already been in, to visit the remaining fragments and complete the exploration.

Adding or increasing a distance-based cost is not sufficient to prevent this phenomenon. Consider a robot that is approaching a corner of an unexplored room. It can either visit the entire corner, or just come near it and continue away from the corner. The latter plan almost surely has higher gain, regardless of how high the distance cost is set.

The solution is to "finish what you started" by weighting those small regions higher. These regions are visible from places where the sensor would see few other faces simultaneously. Consequently, *sensor covisibility* is low for such

faces. By inversely weighting each face by the covisibility, greediness can be avoided.

A. Sensor Covisibility

We define *sensor covisibility* as a binary relation $C_{ij} \in \{0, 1\}$ between a pair of faces f_i and f_j. The pair is covisible if at some edge in G, they are both visible to the sensor:

$$C_{ij} = 1 \iff \exists \{u, v\} \in E : \{f_i, f_j\} \subseteq S_{\{u,v\}} \quad (4)$$

with E and $S_{\{u,v\}}$ as before. Naturally, the relation is symmetric, so $C_{ij} = C_{ji}$.

C_{ij} forms a relation matrix $\boldsymbol{C} \in \{0, 1\}^{K \times K}$ which is used to find the total unseen covisible area c_i for each face with the matrix-vector product

$$\begin{bmatrix} c_1 & c_2 & \dots & c_K \end{bmatrix}^{\mathrm{T}} = \boldsymbol{C} \begin{bmatrix} a_1 & a_2 & \dots & a_K \end{bmatrix}^{\mathrm{T}} \quad (5)$$

where a_i is the area of the ith face, but zero for seen faces. Since seen faces do not contribute, c_i will decrease as its covisible faces are explored. We weight inversely by c_i, which encourages further exploration in the neighborhood of ith face, and thus drives the planner to finish exploring a region before moving on.

B. Inverse Covisibility Scoring

The gain function $g(\cdot)$ is formulated as a surface integral over uncovered space similar to [4], [16]. In our case, it is a sum over the set of new faces seen when traversing from u to v, i.e., $\Delta S = S_{\{u,v\}} \setminus S$, and

$$g(\Delta S) = \sum_{f \in \Delta S} \rho_f \quad \text{with} \quad \rho_f = \exp\left(-\alpha \sqrt{c_f}\right). \quad (6)$$

The parameter α determines the strength of preference for low covisibility faces. Note that traversing an already visited edge has a gain of zero, since $\Delta S = \emptyset$ in that case.

Similar to [3], cost is modelled by exponential decay $c(D) = \exp(-\lambda D + b)$ where D is distance traveled from the current state, and λ is the decay rate. The bias b is introduced to improve numerical stability. The per-edge score function is

$$q(\Delta S, D) = g(\Delta S) c(D) \quad (7)$$

and the score of a path is a sum of q over the path's edges.

The prediction horizon d affects covisibility scoring in that we mark faces that are more than d meters away from the nearest seen face as non-covisible with all other faces, i.e., $c_f = 0$ for such faces.

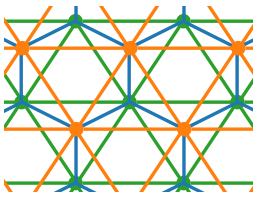

Fig. 2. Illustration of the roadmap lattice for two layers. The orange layer is sitting on top of the green layer, with blue edges connecting the two. The distance between the layers is such that all blue, orange, and green edges are of equal length.

TABLE I
PARAMETER SETTINGS IN OUR EXPERIMENTS

Algorithm		**Model**	
α	1.2	Sensor Range [m]	3.0
λ	0.35	Edge Length [m]	1.5
b	7	Branching Factor	~ 6
N	30 000		

TABLE II
ENVIRONMENTS USED IN OUR EXPERIMENTS

Environment	**Office [3]**	**Labyrinth [4]**
Bounding Box	$34 \times 25 \times 5$	$40 \times 40 \times 3$ m^3
Surface Area of S_Ω	1 725	2 942 m^2
Faces K	24 386	30 819
Face Area	~ 0.07	~ 0.095 m^2
$\lvert V \rvert$	573	1 449
$\lvert E \rvert$	1 540	4 822
Pairwise Path Length	~ 7.6	~ 10.5 m

V. Implementation

The covisibility matrix C is computed by ray-casting radially outwards at regular intervals of 25 cm along every edge $\{u, v\}$. The sensor geometry is thus a sphere, limited to a radius of 3 m. The sphere is approximated by an icosphere of 2 562 vertices, with one ray cast at each vertex. Since G is constant, ray casting is done once.

The roadmap is constructed by multiple layers of triangular tilings in the XY plane, shown in Fig. 2. Each layer is offset from the one below so that its vertices end up at the triangle centers of the layer below. Each vertex in the upper layer is connected to the three vertices making up the triangle below it. The layers are stacked Z-wise so that the inter-layer edges are the same length as the sides of the triangles. This means that all edges are equally long, in our case, 1.5 m. We remove edges that intersect the environment mesh. Faces that are not visible from anywhere in G are discarded from S_Ω.

With a budget of $N = 30\,000$ paths and a branching factor around 6, we search all paths of length 3 and below, and about half the paths of length 4. The selected path is most often of length 4 since Eq. (7) is non-negative. We take a single step of the best path, thus moving 1.5 m before the planning cycle begins again. All pertinent parameters are presented in Table I.

VI. Experiments & Evaluation

In this section, we first show that non-greedy planning does improve long-term performance for map-predictive exploration. Following that, we answer the two questions posed at the beginning of this paper: what is the effect of the prediction horizon, and does an RRT-based planner exhibit more greediness given map predictions. We then show that a hybrid approach can be taken where long-term and short-term benefits are gained from map predictions. We run our experiments in two indoor environments: first, the maze environment from [4] seen in Fig. 1, and second, the office environment from [3] depicted in Fig. 4. Environment

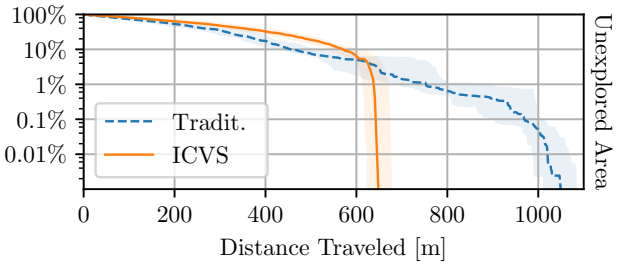

Fig. 3. Unexplored surface area over distance with the same settings as in Fig. 1, in the maze environment. Shaded area is $\pm 2\sigma$. "ICVS" is using the inverse covisibility score described in Section IV with backtracking search, and traditional uses shortest-path search with uniform information gain. In both cases, the prediction horizon $d = \infty$. The non-greedy variant lags behind until around the 5 % mark, and reaches completion shortly after.

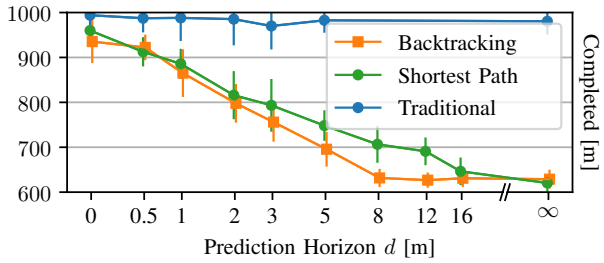

Fig. 5. Distance at completion in meters for some choices of prediction horizon d, comparing the two path planning variants in the office environment, both using inverse covisibility scoring. For reference, a traditional information gain with shortest-path planning is also shown. The horizontal axis is logarithmic. Vertical lines show $\pm 2\sigma$. Backtracking search improves until $d \geq 8$, while shortest-path search improves until $d = \infty$ but slower. The traditional formulation is unaffected by the prediction horizon.

statistics are summarized in Table II for comparison. An important difference between the two environments is that the office environment has *outlier faces* that can only be seen from particular edges in the roadmap. The outlier faces are artifacts from reconstructing the mesh from noisy sensor data using [3]. The maze environment, on the other hand, has no outlier faces as it is not a reconstruction. Unless otherwise stated, each experiment is repeated 10 times.

A. Traditional Vs. Non-Greedy Planning

Our first experiment establishes whether or not a non-greedy scoring function actually outperforms a traditional formulation in the long run, as we have hypothesized. The traditional formulation uses shortest-path search, with uniform information gain, i.e., $\rho = 1$, and both are given knowledge of the full map, $d = \infty$. Figure 3 shows the aggregate statistics of that comparison.

In Figure 1, eight snapshots are presented from one such run: four from the non-greedy planner, and four from the traditional planner. The snapshots show that the traditional planner leaves corners and other smaller regions unexplored throughout the map. The covisibility score is encoded in color, showing that these regions would have had high gain if covisibility scoring had been used in the traditional setting.

Figure 3 shows unexplored space over distance traveled for the same runs. As expected, the traditional formulation is fast to cover ground early on, but as exploration nears completion, the non-greedy planner outpaces it since it has not left any fragments behind. Meanwhile, the traditional planner must travel an additional 400 m to finish, 63 % longer. This mirrors the findings in [2]–[4], [6], [7], and therefore validates the model.

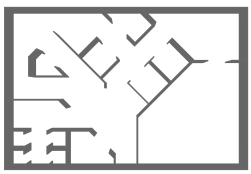

Fig. 4. The office environment from [3] used in the experiments of Section VI-C. Even though it is considerably smaller, completion distances are comparable to those of the maze environment for both traditional and non-greedy exploration.

B. Planning Style and Map Prediction

Figure 5 shows a comparison of the two planning styles using inverse covisibility scoring along with a traditional formulation using a uniform gain in the office environment for a selection of prediction horizons. Backtracking search performs better than shortest-path search, except when the prediction horizon is either very near, or very far away. Backtracking search reaches its optimum at $d = 8$, which is approximately the planner's reach beyond explored space, and also approaching the environment size. Notably, the optimum for shortest-path search is reached much later, at $d = \infty$, where the two planners perform equally well. Traditional exploration is unaffected by map prediction.

C. Effects of the Prediction Horizon

Figure 6 (top) shows completion over distance for a selection of d values, when using inverse covisibility scoring and backtracking search. Lower d values lead to more rapid exploration at first, suggesting more greediness. These are caught up by the high d conditions as more of the environment is explored. The higher d conditions finish abruptly, while the low d conditions do not because of the considerably fragmented unexplored space they have left behind as evidenced by the IPR.

The greediness thus decreases with the prediction horizon d, and higher d conditions finish with a shorter total distance. For $d \geq 8$, the performance is the same as for $d = \infty$. This is the limit of the planner's ability to take advantage of map predictions due to limited search depth.

D. Leaving Some Things Behind

Figures 3 and 6 shows that improved late-stage performance has come at the cost of early-stage performance, which is the intended behavior. Non-greedy exploration finishes exploring every corner and crevice before moving on, therefore delaying exploring high gain regions. However, since the office environment is a reconstruction from noisy sensor data, it contains outlier faces that are only visible from a single edge in the roadmap. The proposed scoring function can be amended to ignore such outliers by setting

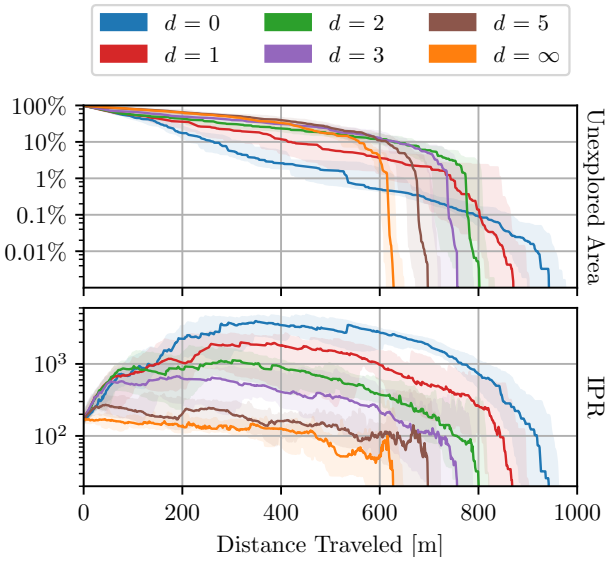

Fig. 6. Completion and isoperimetric ratio at different prediction horizons d meters. Solid lines represent median, shaded regions represent a 90 % confidence interval. Low d values approach completion faster but are caught up by higher d values towards the end. The IPR explains why: low d values get ahead by being less thorough, while high d values have left no unexplored space behind when they approach completion.

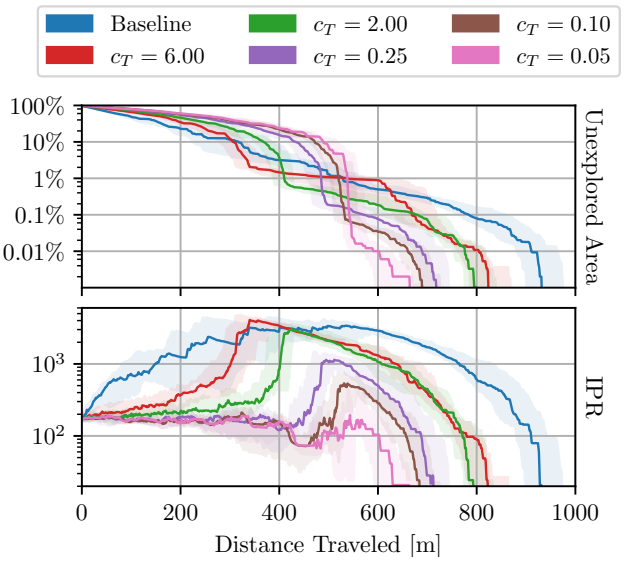

Fig. 7. Using gain Eq. (8) we achieve a trade-off between speed and thoroughness. Smaller values of c_T means that smaller areas are left behind. Higher values allow skipping the most difficult-to-see faces, which are among the most costly to get to. An inflection point occurs when only faces with $c_f < c_T$ remain. IPR increases *before* that point, as unexplored space starts to become fragmented.

their score low. Specifically, we substitute the covisible area with a thresholded variant in the gain function of Eq. (6),

$$\hat{g} = g[c_f \leftarrow \hat{c}_f] \quad \text{where} \quad \hat{c}_f = \begin{cases} c_f + P & c_f < c_T \\ c_f & \text{otherwise} \end{cases} \tag{8}$$

where P is a penalty incurred for faces of below-threshold covisibility, and c_T is the threshold itself. Since c_f is an exponent in Eq. (6), the penalty drastically lowers the gain for such faces, allowing the planner to skip them until only such below-threshold faces.

Figure 7 shows the completion and IPR of this formulation in the office environment with a selection of thresholds c_T, with $d = \infty$ and $P = 200$. Baseline is using traditional information gain. With $c_T \in \{0.05, 0.10, 0.25\}$, only outlier faces are skipped. Consequently, results are similar to $d = \infty$ of Fig. 6, but early-stage performance has improved: at 400 meters of travel, about 12 % more space is explored.

With $c_T = 6.00$, we reintroduce some greediness by allowing to skip larger regions. This condition gets to 10 % unexplored at the same time as the baseline, but manages to finish earlier, showing it has improved late-stage performance without sacrificing early-stage performance. The IPR curve also shows the planner is not simply reverting to traditional exploration, as IPR rises later, and decreases faster than the baseline. This is because even though some unexplored regions are left behind, once only such regions remain, they are explored in a non-greedy manner as before.

VII. Conclusion

In traditional exploration planning, space is either explored and known, or unexplored and unknown. With map predictions, a third state arises: unexplored but predicted space. This third state of space calls into question the use of shortest-path search as the basis for exploration planning. We have demonstrated that a more flexible path planning can indeed improve performance, but these benefits are modest compared to the use of inverse covisibility scoring. This suggests that similar heuristics should be sufficient to enable map-predictive capabilities in RRT-based exploration planners. We have also shown that using such a heuristic, greediness decreases as the prediction horizon increases, indicating that even a relatively weak map predictor may be useful. Finally, we have shown that it is possible to combine early and late stage performance with a thresholded inverse covisibility scoring function, allowing the planner to skip small details without losing its non-greedy behavior.

The next step in this line of work is to design an inverse covisibility score for a state-of-the-art map-predictive method. We stress that it must be map predictive, as we have not demonstrated any advantage to inverse covisibility scoring without map prediction. A second line of inquiry is to predict volumetric data, e.g., [17], [18], as exploration state typically is represented.

References

[1] H. Ardiny, S. Witwicki, and F. Mondada, "Autonomous exploration for radioactive hotspots localization taking account of sensor limitations," *Sensors*, vol. 19, no. 2, p. 292, 2019.

[2] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "Next-Best-View" planner for 3D exploration," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1462–1468.

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

[4] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020.

[5] A. Aydemir, P. Jensfelt, and J. Folkesson, "What can we learn from 38,000 rooms? reasoning about unexplored space in indoor environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4675–4682.

[6] M. Luperto, M. Antonazzi, F. Amigoni, and N. A. Borghese, "Robot exploration of indoor environments using incomplete and inaccurate prior knowledge," *Robotics and Autonomous Systems*, vol. 133, 2020.

[7] R. Shrestha, F.-P. Tian, W. Feng, P. Tan, and R. Vaughan, "Learned map prediction for enhanced mobile robot exploration," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 1197–1204.

[8] M. Luperto, M. Antonazzi, F. Amigoni, and N. A. Borghese. (2020) Exploration of indoor environments predicting the layout of partially observed rooms. Accessed 2020, Oct. [Online]. Available: https://youtu.be/OQz-N8xAR6Q?t=121

[9] W. Gao, M. Booker, A. Adiwahono, M. Yuan, J. Wang, and Y. W. Yun, "An improved frontier-based approach for autonomous exploration," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018, pp. 292–297.

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

[11] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, "Fast frontier-based information-driven autonomous exploration with an mav," *arXiv preprint arXiv:2002.04440*, 2020.

[12] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[14] A. Elhafsi, B. Ivanovic, L. Janson, and M. Pavone, "Map-predictive motion planning in unknown environments," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 8552–8558.

[15] A. Q. Li, F. Amigoni, and N. Basilico, "Searching for optimal off-line exploration paths in grid environments for a robot with limited visibility," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 26, no. 1, 2012.

[16] L. Yoder and S. Scherer, "Autonomous exploration for infrastructure modeling with a micro aerial vehicle," in *Field and service robotics*. Springer, 2016, pp. 427–440.

[17] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D euclidean signed distance fields for on-board MAV planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 1366–1373.

[18] D. Duberg and P. Jensfelt, "UFOMap: An efficient probabilistic 3D mapping framework that embraces the unknown," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6411–6418, 2020.