Learned Map Prediction for Enhanced Mobile Robot Exploration

Rakesh Shrestha[1,*], Fei-Peng Tian[1,2,*], Wei Feng[2], Ping Tan[1] and Richard Vaughan[1]

***Abstract*— We demonstrate an autonomous ground robot capable of exploring unknown indoor environments for reconstructing their 2D maps. This problem has been traditionally tackled by geometric heuristics and information theory. More recently, deep learning and reinforcement learning based approaches have been proposed to learn exploration behavior in an end-to-end manner. We present a method that combines the strengths of these different approaches. Specifically, we employ a state-of-the-art generative neural network to predict unknown regions of a partially explored map, and use the prediction to enhance the exploration in an information-theoretic manner. We evaluate our system in simulation using floor plans of real buildings. We also present comparisons with traditional methods which demonstrate the advantage of our method in terms of exploration efficiency. We retain an advantage over end-to-end learned exploration methods in that the robot's behavior is easily explicable in terms of the predicted map.**

I. INTRODUCTION

Modeling a previously-unknown environment is a canonical task in mobile robotics. The task of planning and executing robot trajectories to create a world model, henceforth 'map', is known as 'exploration'. Autonomous exploration is a component of many real-world robotic applications, including search and rescue [1], planetary exploration [2], visual inspection [3], 2D/3D reconstruction [4] and mining [5]. The completeness and efficiency of exploration is important to facilitate these applications. Here we describe a novel exploration method that exploits learned priors over maps to explore more efficiently than current methods.

Exploration is closely related to some well known NP-complete problems, such as the art gallery problem or the traveling salesman problem, with the important difference that the polygon to be covered or graph to be traversed is discovered dynamically and incrementally during run time. Previous methods for autonomous exploration can be broadly classified into two categories: frontier-based methods and information-theoretic methods. The frontier approach maximizes map coverage by moving to new frontiers - regions on the boundary between free space and unexplored space [6]. Path cost and expected information gain (i.e. utility) are commonly used to determine the next frontier to visit. The information-theoretic methods formulate the problem as map entropy minimization, i.e., information gain with probabilistic map representation [7]. Hence, a good information gain prediction plays a key role in both approaches.

[1]School of Computing Science, Simon Fraser University, Canada. {rakeshs,pingtan,vaughan}@sfu.ca

[2]School of Computer Science and Technology, Tianjin University, China.{tianfeipeng,wfeng}@tju.edu.cn

*The first two authors contributed equally to this work.

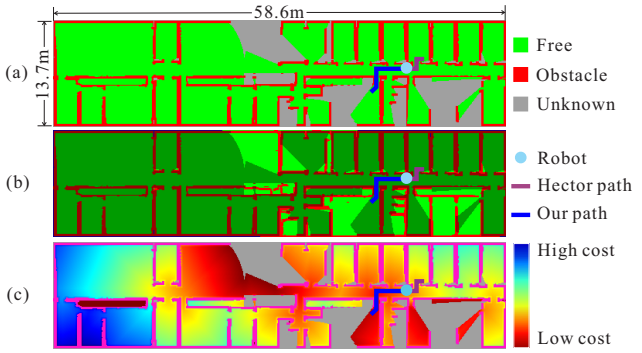

Fig. 1. **Deep learning based map prediction for autonomous exploration.** (a) Current incomplete map. (b) Predicted map using the network trained over many previous maps. (c) Cost transform for explored cells. Compared to Hector exploration [1], our method chose a more rewarding direction to explore due to a better estimate of information gain.

Estimating information gain amounts to predicting sensor measurements in unseen regions of the map. The general approach is to assume that the robot is faced with a map that is either somewhat typical for its application domain, or somewhat regular in itself, or both, so that partial views of the map can afford useful predictions of the unseen parts. Methods for achieving this vary in their advantages and limitations in map prediction and hence information gain estimation. While this problem is not new, there has been recent interest and new ideas. Pimentel et al. [8] proposed an elegantly simple heuristic whereby wall segments at frontiers are assumed to extend into the unseen area when computing expected information gain. Cost-utility based exploration [9], [10] simply estimates information gain by counting unknown cells within a predefined radius of a frontier point. While these methods may work locally on simple floor-plans, they are unlikely to perform well on more complicated floor-plans. Most recent information-theoretic methods [11], [12] predict the information gain based on Gaussian Mixture Models [13] or Gaussian process with Bayesian inference, by one-step sensor look-ahead measurements using sensor likelihood prediction models [14], [15], which tacitly assume the unknown areas are always free. However, this assumption does not hold in general, thus the final information gain estimation is often inaccurate.

We propose a data-driven approach that does not rely on explicit assumptions about the environment, but instead learns regularities from examples. Specifically, we employ Variational Autoencoders (VAE) [16] to predict unknown map regions beyond frontiers. In our chosen indoor setting this allows us to learn generalizations over many building floor plans to make informed decisions for faster exploration of novel floor plans. As illustrated in Figure 1, using our

Preprint submitted to 2019 International Conference on Robotics and Automation (ICRA). Received September 15, 2018.

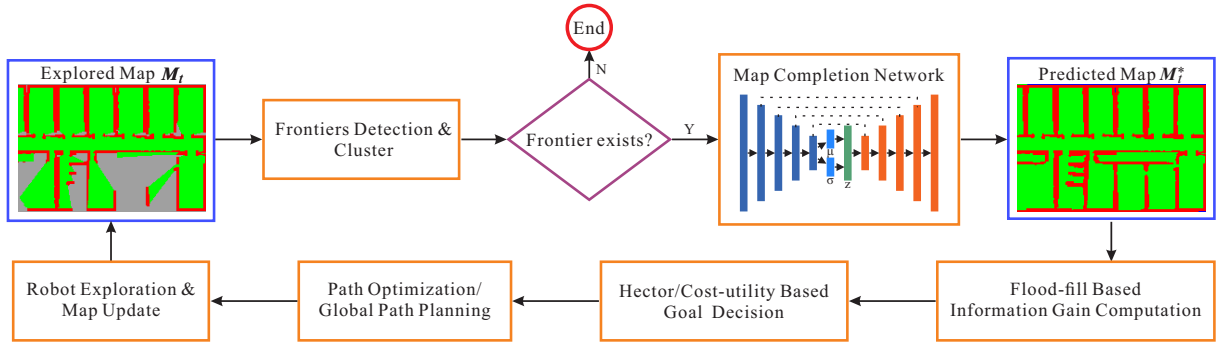

Fig. 2. System workflow.

VAE based map prediction, we estimate the potential areas that can be mapped from each frontier which we demonstrate can be more accurate and efficient than a single or multi-step lookahead in the sensor measurements.

Our system addresses the exploration problem in two steps: 1) predicting unknown regions to identify navigation goals that maximize map coverage; 2) navigating to those goal positions using well-established techniques. We employ deep learning to tackle the former in isolation, given the recent success of deep generative neural network models [17], [16]. Navigation is a well-studied problem in robotics where the traditional methods still outperform deep learning based methods [18], [19], [20]. By separating the problem into these two steps, we are able to utilize the strengths of both deep learning and traditional methods while also maintaining the comprehensibility of the overall system pipeline.

II. Related Work

Frontier-based exploration methods are easy to understand and implement and work fairly well in practice. These methods maintain a list of boundaries between explored and unexplored regions in the partial map, known as 'frontiers'. The robot iterates over choosing the currently most promising frontier, planning and executing a path that drives through it while extending the map to eliminate the frontier, until no frontiers remain. Yamauchi et al. [6] proposed a well-known Nearest-Frontier Exploration method which always chooses the closest frontier as the next goal. Bourgault et al. [9] proposed to use a cost-utility function based on expected information gain and path cost. There are several extensions to this idea in [21], [10], [22], [23]. For efficient information gain computation, Umari et al. [10] used an RRT tree for frontier detection and counted unknown cells surrounding a frontier point within a predefined radius to estimate information gain. Ström et al. [22] presented a method to match the area beyond the frontiers with the most similar map in a database, and compute the expected information gain based on the retrieved match. Pimentel et al. [8] proposed a simple heuristic map prediction by linearly extending the walls or turning the walls by 90 degrees to compute expected information gain. While such simple local heuristic map completion and information gain computations can work well on simple floor-plans, we could hope to do better using more sophisticated priors over maps.

The information-theoretic methods use information theory to minimize uncertainty of the map. Several authors like [24], [25], suggest the use of information gain (also called mutual information in some contexts) as a measure of the reduction of map uncertainty. In recent work, Vallvé et al. [26] proposed to compute map and path information gain densely for the entire configuration space and apply a grid-step gradient on the potential fields to directly optimize a path. Jadidi [11] proposed a Gaussian Process (GP) based method to build an information field for the entire configuration space, but the final decision-making is still based on a utility function that chooses a goal which balances the path cost and expected information gain from frontiers extracted from the GP map. Bai et al. [12] present a method to predict the information gain in the surrounding partial map of the current robot pose based on Gaussian process and Bayesian inference. Their method chooses the point with the largest expected information gain as the goal for the next step. These methods, while providing a new perspective on information gain estimation, often suppose unknown space as free or suppose Gaussian distribution of occupancy probabilities which does not hold in real floor-plans, thus the accuracy of the final information gain is limited.

Researchers have also proposed to use deep learning and reinforcement learning techniques to solve the autonomous exploration problem. Bai [27] presented a supervised learning method that chooses the next step which has a fixed distance to the robot and maximizes the expected information gain from 36 candidate actions. Lei et al. [28] proposed a reinforcement learning based method aiming to avoid obstacles in exploration.

In contrast to these methods, we propose using deep learning to auto-complete unseen map regions in geometric detail and use this filled-in map to measure expected information gain. We then combine this prediction with different exploration strategies to improve exploration efficiency.

III. Problem Statement

Consider a nonholonomic wheeled robot equipped with a limited visibility laser range scanner, which is operating in bounded 2D environment $E \subset \mathbb{R}^2$. It incrementally builds an occupancy grid representation $\boldsymbol{M}_t$ from the exploration of environment E, where t is the time step. The goal of autonomous exploration is to plan a sequence of actions $\mathcal{A}$

Preprint submitted to 2019 International Conference on Robotics and Automation (ICRA). Received September 15, 2018.

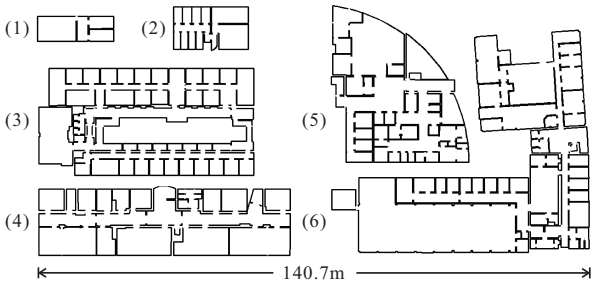

Fig. 3. Some floor-plan examples in KTH dataset.

by which the area of $\boldsymbol{M}_t$ is maximized in every time step t. Here each action $a \in \mathcal{A}$ belongs to Lie Algebra [29] of special Euclidean group $SE(2)$. Due to the NP-hardness of this problem, we cannot obtain the optimal solution in polynomial time, thus we relax it as a minimization of an immediate cost $\boldsymbol{C}$ which is a function of path cost $\boldsymbol{C}_p$, possibility of colliding with obstacles $\boldsymbol{C}_o$ and expected information gain $\boldsymbol{I}$ to determine the actions $a \in \mathcal{A}$.

IV. System Overview

Figure 2 shows the work-flow of the proposed method. Based on the current explored map $\boldsymbol{M}_t$, we first detect all the frontier points $\mathcal{F}_t$, then classify them into clusters $\boldsymbol{F}_t^i \subseteq \mathcal{F}_t$ using the DBSCAN algorithm [30], where $i = 1, \ldots, N$. N is the number of clusters. Then we use the VAE network to predict the occupancy of cells in the unseen regions beyond each frontier cluster $\boldsymbol{F}_t^i$, to obtain a predicted map $\boldsymbol{M}_t^*$. We then compute the information gain $\boldsymbol{I}_t^i$ for each frontier cluster $\boldsymbol{F}_t^i$ using the predicted map $\boldsymbol{M}_t^*$, as input to the otherwise-standard Hector exploration [1] or cost-utility exploration [10], [31] which generates a feasible path for the robot's exploration.

V. Map Completion Network

Recently, generative networks have shown impressive performance in semantic inpainting and inferring large missing regions in images with high accuracy [32], [33], [34]. Similar ideas have been applied to predict missing parts of partial 2D maps [35], [36] where a local set of robot-centric sensor measurements are used to infer a single missing observation.

We use the VAE network [16] for map completion because of its training/testing speed and the ability to predict large unknown regions. Alternatives like Generative Adversarial Networks [17] are too computationally demanding for online planning. [35], [36] use Deep Sum-Product Networks [37] to get performance similar to GANs, but the prediction is limited to a narrow Field of View (FOV).

A. Dataset

We use the KTH dataset [38] (see Figure 3 for some examples) to generate partial maps and ground truth for training the map completion network. The KTH dataset provides 165 floor plans with 6,248 rooms on the KTH campus. We manually cleaned some repeated floor plans and randomly split them into training and testing sets with a proportion of 3:1. We use Hector exploration [1] to explore the map and record a 256×256 region centered around each frontier cluster center. We encode obstacles, free space, and unknown space into different color channels and append a 80×80 mask to specify the region we want to predict. Figure 4 shows an example of a generated dataset for the proposed network.

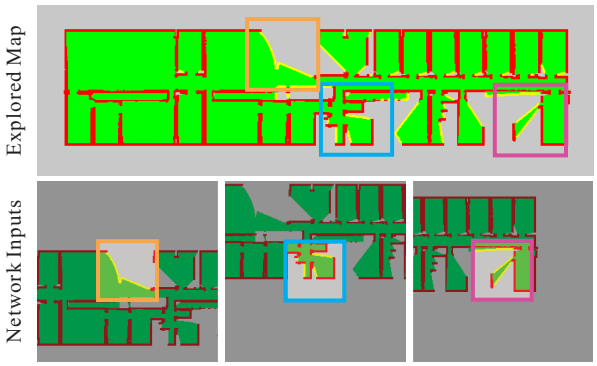

Fig. 4. **Network inputs generation.** The upper image shows the current explored map, frontiers are marked as yellow. Lower row shows three inputs for the map completion network, whose sizes are all 256×256 with mask (prediction area) size at 80×80 centered in frontier cluster centers. The network is trained to generate the true contents of the masked center region given the input.

B. Network Structure

Figure 5 shows the architecture of our map prediction network. The encoder part of the network learns to output a latent representation z in a lower dimensional space and the decoder reconstructs the missing parts of the map using the compact latent representation.

Encoder: Our encoder is based on the ResNet architecture [39]. We use the ResNet architecture on account of its demonstrated performance in the task of image classification on benchmark datasets and short training time.

The encoder outputs a mean (μ) and a log variance $log(\sigma)$ of the encoding of size 8×8×512. The final encoding fed to decoder network is sampled from the Gaussian distribution $\mathcal{N}(\mu, \sigma)$. We do not use fully connected layers as they explode the number of parameters in the network and did not yield a significant improvement in our results.

Decoder: The decoder network is essentially a mirror image of the encoder. While the encoder reduces the size of the feature plane while increasing their number, the decoder does the opposite. This is achieved by transposed convolution layers (also known as deconvolution layers) [40]. The output from the decoder network is a probability map of obstacles, with pixel values in the range 0 to 1. We then apply a threshold of 0.5 to determine obstacles and free spaces.

The network weights are randomly initialized without pre-training, because our training data is significantly different from standard benchmark datasets such as ImageNet [41].

C. Network Loss Functions

1) Prediction Loss: The prediction task in our case can be viewed as a pixel-wise binary classification: for each pixel we have to predict whether it is an obstacle (label 1) or free

Preprint submitted to 2019 International Conference on Robotics and Automation (ICRA). Received September 15, 2018.

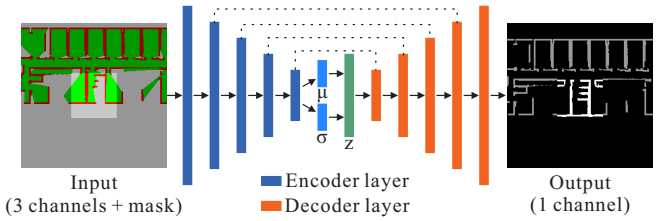

Fig. 5. **The architecture of our map prediction network.** The input is a four-channel image containing obstacles (red), free space (green), unknown space (blue) and a mask for prediction the region. The output is a single-channel image representing the probability of obstacles. We use a threshold of 0.5 to binarize output into obstacle and free space.

space (label 0). Hence, we choose Binary Cross Entropy loss between network output x_n and its true label y_n for pixel n. The loss for a single pixel n is given as:

$$l_n = -[y_n \cdot log x_n + (1 - y_n) \cdot log(1 - x_n)] \quad (1)$$

The total loss of the prediction is the sum of all the pixel-wise losses for the masked 80×80 region,

$$L_{prediction} = \sum l_n. \quad (2)$$

2) Kullback-Leibler (KL) Divergence Loss: The KL-divergence on the latent encoding is used as a loss function to penalize the divergence of the encoding from a Standard Normal distribution The loss is given as:

$$L_{kld} = \sigma^2 + \mu^2 - log(\sigma) - 1 \quad (3)$$

Please refer to [16] for its justification.

The final loss is the weighted sum of the two losses $L_{prediction}$ and L_{kld} from Equation (2) and Equation (3).

$$L_{final} = \gamma L_{prediction} + (1 - \gamma) L_{kld} \quad (4)$$

where $\gamma \in (0, 1)$ controls the relative importance of two losses. We use $\gamma = 0.99$ for training our network.

VI. Map Completion Augmented Exploration

A. Information Gain Computation

For estimating the information gain we need to find the regions that can be immediately explored from a frontier cluster, which is achieved by using the flood-fill algorithm. Starting from a frontier point, a Depth First Search is performed to find all the connected pixels that are unknown in the current map $\boldsymbol{M}_t$, but are free according to the predicted map $\boldsymbol{M}_t^*$. Figure 6 illustrate the flood-fill regions with and without map prediction.

The canonical definition of information gain for each frontier cluster $\boldsymbol{F}_t^i$ is:

$$\boldsymbol{I}_t^i(\boldsymbol{M}, x_i) = \boldsymbol{H}(\boldsymbol{M}) - \boldsymbol{H}(\boldsymbol{M}|x_i), \quad (5)$$

where $\boldsymbol{H}(\boldsymbol{M})$ is the *current entropy* of the map at time t and $\boldsymbol{H}(\boldsymbol{M}|x_i)$ is the *posterior entropy* of the predicted map with the new flood-filled map information x_i. The Shannon's entropy [42] over an occupancy grid map $\boldsymbol{M}$ is defined as

$$\boldsymbol{H}(\boldsymbol{M}) = -\sum_j p(m_j) \log p(m_j) + (1 - p(m_j)) \log(1 - p(m_j)), \quad (6)$$

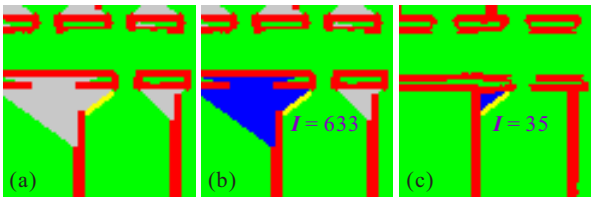

Fig. 6. Comparison of flood-filled region without and with map prediction. Frontiers and flood-filled regions are marked as yellow and blue respectively. (a) Current explored map. (b) and (c) are flood-fill regions without and with map prediction. The information gain of the frontiers can be truthfully reflected from the map prediction in (c).

where $p(m_j)$ is the probability of the cell m_j being occupied. An unknown cell m_j in $\boldsymbol{M}$ decreases the entropy by 1, since $p(m_j) = 0.5$. In comparison, the known cells do not contribute to the information gain. The information gain $\boldsymbol{I}_t^i$ is then equal to the number of unknown cells that will become known which is in keeping with our flood-fill based method.

B. Augmenting Exploration Planners

The expected information gain $\boldsymbol{I}_t^i$ from the proposed map prediction can be used to augment different kinds of goal planners or path optimizers instead of just a specific method. Here we choose an exploration transform [43] based planner: Hector exploration planner [1] and cost-utility function based planner [10], [9] as base methods, which are two representative methods for path optimization and navigation goal planning respectively.

1) Augmented Hector Planner: Original Hector exploration [1] builds a costmap known as an exploration transform [43] which assigns a cost to every explored free cell in the current map $\boldsymbol{M}_t$. The cost has two parts, i.e. the path cost to the nearest frontier point, and a obstacle cost to the nearest obstacle. Inspired by this idea, we build our cost transform $\boldsymbol{C}(m)$ for the free cell m by additionally considering our information gain estimate as follows:

$$\boldsymbol{C}(m) = \min_{\mathbf{F}_t^i \subseteq \mathcal{F}_t} \left\{ \min_{f \in \mathbf{F}_t^i} \boldsymbol{C}_p(f) - \alpha \sqrt{\boldsymbol{I}_t^i} \right\} + \beta \boldsymbol{C}_o, \quad (7)$$

where $\boldsymbol{C}_p(f)$ is the path cost from cell m to frontier cell f, $\boldsymbol{I}_t^i$ is the information gain for frontier cluster $\boldsymbol{F}_t^i$, and $\boldsymbol{C}_o$ is the collision cost of the cell m formulated as a thresholded distance to nearest obstacle. α and β are the weights for information gain and collision cost respectively. Since information gain is proportional to expected unmapped free areas (m^2) while path cost is related with length (m), we use square root for information gain $\boldsymbol{I}_t^i$ to keep their units consistent. The square root response function for information gain cost also helps to dampen the high information gain estimates due to potential mispredictions. Note that the cost transform $\boldsymbol{C}(m)$ for each cell m can be computed efficiently in an incremental way by propagating the costs starting from the frontiers points.

After evaluating our cost transform, we can plan a path from the current robot pose by following the steepest gradient of the transform until it reaches a frontier point.

Preprint submitted to 2019 International Conference on Robotics and Automation (ICRA). Received September 15, 2018.

2) Augmented Cost-Utility Planner: The cost-utility exploration scheme [10], [9], [44] assigns a cost to each frontier point based on its travel cost $\boldsymbol{C}_p(f)$: length of the shortest path to the frontier f, and utility (information gain), which is $\sqrt{\boldsymbol{I}(f)}$ in our case.

$$\boldsymbol{C}(f) = \boldsymbol{C}_p(f) - \lambda\sqrt{\boldsymbol{I}(f)} \tag{8}$$

where $\boldsymbol{I}(f) = \boldsymbol{I}_t^i$ for $f \in \boldsymbol{F}_t^i$ and λ is a weight to adjust relative importance between the cost and the utility. The frontier with the minimum cost is then chosen to be the next navigation goal and a global path planner (Dijkstra algorithm in our implementation) is used to generate a path to the goal.

VII. Experiments

A. Experimental Setup

1) Simulation: The simulations for both dataset generation and evaluations were performed using the Stage simulator [45] Pioneer P2-DX mobile robot and SICK LMS scanning laser rangefinder (LRF) models. The FOV and range of the LRF were limited to $270°$ and $5m$ respectively with 512 samples. The sensor update rate was 10 Hz in simulation time. The system is implemented in ROS [46].

The environment map is a 2D occupancy grid with a resolution of 0.1m per pixel. Map updates are done by ray-casting range measurements for each laser range reading to label free and obstacle cells. To allow fast dataset generation and evaluations, we use ground truth localization from the simulation rather than running SLAM. Since we assume good localization/mapping and do not aim to optimize the path for lower pose uncertainty [47], the use of ground truth localization is justified.

2) Baselines for Comparison: The comparison baselines we use for our evaluations are: Nearest Frontier Exploration ("nearest_frontier") [6], Hector exploration planner ("original_hector") [1] and cost-utility exploration ("original_cost_utility") [9].

The Hector exploration planner augmented with our information gain is referred to as "ig_hector" and the cost-utility planner is "ig_cost_utility". As an upper bound we also compare the performance of the Hector planner augmented by information gain from ground truth map ("ig_hector_gt"): the impossible perfect oracle for map prediction. Note that the use of ground truth information gain does not guarantee a perfect exploration as it is still a greedy heuristic that in general will not lead to a globally optimal solution for the coverage of whole map. Thus, this provides a tight upper bound for a well-informed local exploration method.

3) Configuration Settings: For fair comparison, we set the weights for information gain in "ig_hector" and "ig_cost_utility" the same, viz. $\alpha = 3$ (Equation (7)) and $\lambda = 3$ (Equation (8)). And we set $\beta = 5$ in Equation (7).

B. Map Completion Network Evaluation

1) Evaluation Metrics: Following prior authors [8], [12], we measure the average percentage of area coverage against exploration time to evaluate exploration efficiency. We also report the average time and travel distances taken to achieve 85% map coverage. This approach is similar to the one proposed in [8] where the exploration is stopped after a fixed duration of time. This strategy allows us to ignore some minor unmapped corners to better reflect the exploration efficiency. Since the size of our maps vary, selecting a constant time threshold will not lead to a fair comparison. Hence, we choose percentage of area covered instead.

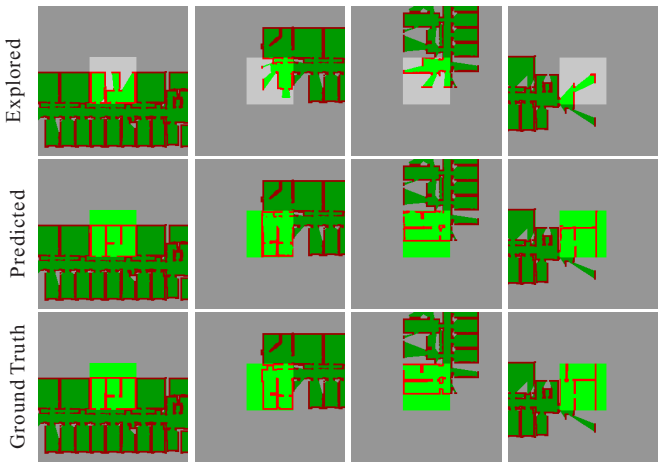

Fig. 7. Four map prediction results by the VAE network. In the first three columns, the topology of the map is correctly predicted while the fourth column shows an incorrect but plausible prediction.

We present the results for 12 different maps from the testing dataset of the KTH floor-plans. For each map and each method, we run 20 tests from different initial robot poses chosen at random and evaluate their performance.

For training the map completion network, we first run 50 explorations using the original Hector planner for each map in the training dataset and crop a 256×256 region around each frontier cluster. This results in a training dataset of 1.5 million partial maps. Then we test the map prediction performance on the testing maps with about 140,000 inputs. The network takes an avergae of 5 milliseconds with 1 millisecond standard deviation to predict one batch of frontiers (32 inputs) on a Nvidia GTX980 GPU. We obtain overall map cell contents prediction accuracy of 92.5%; with precision and recall for free space prediction of 95.0% and 96.3% respectively. The precision/recall for obstacle predictions are 77.0% and 70.6% respectively. Some examples of our network's predictions are shown in Figure 7. From these evaluations and Figure 7, we can see the proposed VAE network can complete maps with high accuracy.

C. Analysis of Exploration Efficiency

1) Area Coverage Over Time: Figure 8 shows the percentage of total map coverage over exploration time. We compare our "ig_hector" and "ig_cost_utility" methods with baselines as well as the Hector planner [1] augmented by oracular information gain from ground truth maps ("ig_hector_gt"). The curves of Figure 8 show that the map-prediction augmented methods explore the map more quickly than the baselines in the exploration process. All methods eventually converge

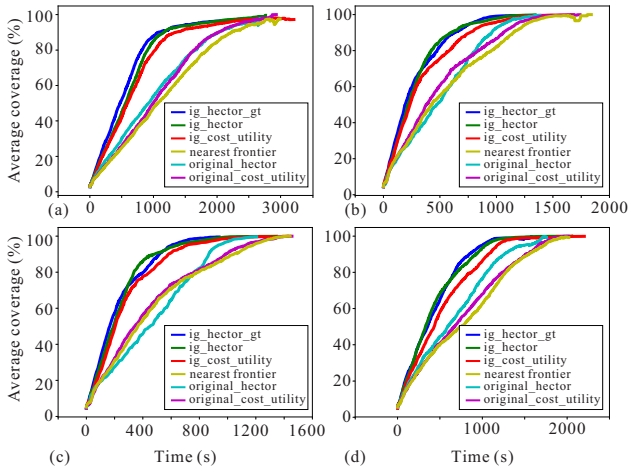

Fig. 8. Percentage map coverage against time for 4 floor plans. Results are averaged over 20 experiments.

to near-perfect coverage, but the augmented methods do so earlier than the baselines. We notice that the nearest frontier planner and original cost-utility methods show similar performance while the original Hector is slightly better than the other two. In contrast, the proposed "ig_hector" perform much better, reaching nearly the same exploration efficiency of the upper bound oracle-informed method.

The results suggest the VAE network has learned to predict unseen map areas well enough to usefully improve exploration efficiency.

2) Travel Distance and Time: Figures 9(a) and (c) show the median time taken and distance traveled to cover 85% of the total area using different methods for the 12 test floor plans.

To assess the statistical significance of the difference between map completion time and distance traveled for the baseline methods against our methods, we use Welch's t-test [48]. We choose the proposed "ig_hector" as our basis for comparison. A higher t-value implies a higher confidence that baselines' map completion times or traveled distances are longer than the proposed "ig_hector" method. Figures 9(b) and (d) show the t-values for map completion times and travel distances of all the tests in 12 floor-plans. The t-value corresponding to 95% confidence level is shown in the graph as a dotted line.

Figures 9(a)–(d) show that the proposed methods "ig_hector' and "ig_cost_utility" always have lower map completion time and distance traveled compared to the baseline methods they were based on viz. "original_hector" and "original_cost_utility" respectively. The measured performance distribution of the enhanced methods are better, and differs with a large confidence interval, compared to their baselines.

Our method "ig_cost_utility" has similar travel distance to "original_hector" in some of the maps. This is likely explained by the superior performance of hector exploration compared to cost-utility exploration canceling out the advantage of better predictions.

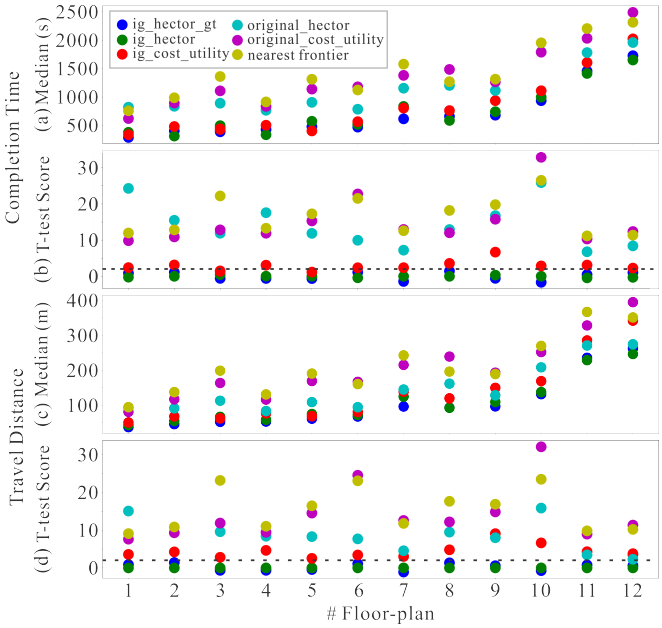

Fig. 9. For each method, we show the median task completion time and trajectory length for 12 different floor-plans, over 20 trials each with different initial robot pose. Smaller values indicate better performance.

Overall, these experiments show that the map prediction from the proposed VAE network results in a performance increase in travel time and distance traveled compared to previous methods in the maps we tested.

VIII. Conclusion and Future Work

We describe a VAE deep neural network that learns to predict unseen regions of building floor plans. This model is used to enhance exploration performance by improving estimates used in path planning, leading to increased efficiency. We suggest that the navigation behavior of our system is more intelligible than that of end-to-end deep learning techniques as it splits the system into a learned generative model with a map output that is easily interpreted by humans, within a well known information-theoretic framework.

In future work, we aim to implement our system on real robots and evaluate its performance. The work can be extended to 3D reconstruction rather than 2D, using ground or aerial robots. Furthermore, we assume accurate localization and mapping so far. Removing this assumption and actively planning for efficient exploration and with good quality localization and mapping would be an interesting extension.

Reproduction, code and data products

All code, data and models used in the experiments are available at `https://git.io/fAX7k`.

Acknowledgements

Supported by the NSERC Canadian Field Robotics Network.

REFERENCES

[1] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, O. von Stryk, and U. Klingauf, "Hector open source modules for autonomous mapping and navigation with rescue robots," in *Proceedings of RoboCup Symposium*, ser. Lecture Notes in Artificial Intelligence (LNAI). Springer, 2013, p. to appear.

[2] K. Otsu, A.-A. Agha-Mohammadi, and M. Paton, "Where to Look? Predictive Perception with Applications to Planetary Exploration," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 635–642, 2018.

[3] A. Ahrary, A. A. Nassiraei, and M. Ishikawa, "A study of an autonomous mobile robot for a sewer inspection system," *Artificial Life and Robotics*, vol. 11, no. 1, pp. 23–27, 2007.

[4] D. Klimentjew, M. Arli, and J. Zhang, "3D scene reconstruction based on a moving 2D laser range finder for service-robots," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2009, pp. 1129–1134.

[5] T. Neumann, A. Ferrein, S. Kallweit, and I. Scholl, "Towards a mobile mapping robot for underground mines," in *Proceedings of the PRASA, RobMech and AfLaI Int. Joint Symposium, Cape Town, South Africa*, 2014.

[6] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997, pp. 146–151.

[7] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[8] J. M. Pimentel, M. S. Alvim, M. F. Campos, and D. G. Macharet, "Information-driven rapidly-exploring random tree for efficient environment exploration," *Journal of Intelligent & Robotic Systems*, pp. 1–19, 2017.

[9] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, 2002, pp. 540–545.

[10] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1396–1402.

[11] M. G. Jadidi, J. V. Miro, and G. Dissanayake, "Mutual information-based exploration on continuous occupancy maps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 6086–6092.

[12] S. Bai, J. Wang, F. Chen, and B. Englot, "Information-theoretic exploration with Bayesian optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1816–1822.

[13] P. Pfaff, C. Plagemann, and W. Burgard, "Gaussian mixture models for probabilistic localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 467–472.

[14] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of artificial intelligence research*, vol. 11, pp. 391–427, 1999.

[15] S. Thrun, "A probabilistic on-line mapping algorithm for teams of mobile robots," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.

[16] D. P. Kingma and M. Welling, "Auto-encoding Variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[17] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[18] J. Oh, V. Chockalingam, S. Singh, and H. Lee, "Control of memory, active perception, and action in minecraft," in *Proceedings of the International Conference on International Conference on Machine Learning (ICML)*, 2016, pp. 2790–2799.

[19] T. D. Kulkarni, A. Saeedi, S. Gautam, and S. J. Gershman, "Deep successor reinforcement learning," *arXiv preprint arXiv:1606.02396*, 2016.

[20] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu *et al.*, "Learning to navigate in complex environments," *arXiv preprint arXiv:1611.03673*, 2016.

[21] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters." in *Robotics: Science and Systems (RSS)*, vol. 2, 2005, pp. 65–72.

[22] D. P. Ström, I. Bogoslavskyi, and C. Stachniss, "Robust exploration and homing for autonomous robots," *Robotics and Autonomous Systems*, vol. 90, pp. 125–135, 2017.

[23] D. P. Ström, F. Nenci, and C. Stachniss, "Predictive exploration considering previously mapped environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2761–2766.

[24] A. Elfes, "Robot navigation: Integrating perception, environmental constraints and task execution within a probabilistic framework," in *Reasoning with Uncertainty in Robotics*, 1996, pp. 91–130.

[25] P. Whaite and F. P. Ferrie, "Autonomous exploration: Driven by uncertainty," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 193–205, 1997.

[26] J. Vallvé and J. Andrade-Cetto, "Potential information fields for mobile robot exploration," *Robotics and Autonomous Systems*, vol. 69, pp. 68–79, 2015.

[27] S. Bai, F. Chen, and B. Englot, "Toward autonomous mapping and exploration for mobile robots through deep supervised learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2379–2384.

[28] T. Lei and L. Ming, "A robot exploration strategy based on Q-learning network," in *IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2016, pp. 57–62.

[29] W. M. Boothby, *An introduction to differentiable manifolds and Riemannian geometry*. Academic press, 1986, vol. 120.

[30] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "Density-based spatial clustering of applications with noise," in *International Conference on Knowledge Discovery and Data Mining*, vol. 240, 1996.

[31] H. H. González-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.

[32] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics*, vol. 36, no. 4, p. 107, 2017.

[33] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," *arXiv preprint arXiv:1801.07892*, 2018.

[34] R. A. Yeh, C. Chen, T.-Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models." in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, no. 3, 2017, p. 4.

[35] A. Pronobis and R. P. Rao, "Learning deep generative spatial models for mobile robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 755–762.

[36] A. Pronobis, F. Riccio, and R. P. Rao, "Deep spatial affordance hierarchy: Spatial knowledge representation for planning in large-scale environments," in *International Conference on Automated Planning and Scheduling Workshop*, 2017.

[37] H. Poon and P. Domingos, "Sum-product networks: A new deep architecture," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2011, pp. 689–690.

[38] 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 (IROS)*, 2012, pp. 4675–4682.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[40] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[42] C. E. Shannon and W. Weaver, "The mathematical theory of communication. 1949," *Urbana, IL: University of Illinois Press*, 1963.

[43] S. Wirth and J. Pellenz, "Exploration transform: A stable exploring algorithm for robots in rescue environments," in *IEEE International Workshop on Safety, Security and Rescue Robotics*, 2007, pp. 1–5.

[44] M. Juliá, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Autonomous Robots*, vol. 33, no. 4, pp. 427–444, 2012.

[45] R. Vaughan, "Massively multi-robot simulation in stage," *Swarm intelligence*, vol. 2, no. 2-4, pp. 189–208, 2008.
[46] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, 2009.
[47] H. Carrillo, I. Reid, and J. A. Castellanos, "On the comparison of uncertainty criteria for active SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 2080–2087.
[48] B. L. Welch, "The generalization of student's' problem when several different population variances are involved," *Biometrika*, vol. 34, no. 1/2, pp. 28–35, 1947.