Task-Directed Semantic Exploration with Sparse Sensing
Final Report

1 Motivation

The standard pipeline in robotic systems includes perception, planning, and control. Each step of the pipeline is typically implemented with a module that aims to maximize performance for its task, e.g., building a map of the environment with high resolution, planning a motion on this map that produces the shortest path to a goal, and moving along this path while considering all obstacles. However, not all steps of this pipeline require maximum accuracy to complete a specific task, and, in fact, this may not even be possible in a resource-constrained setting. In parallel, may environments (especially if man-made) present similarities across many different instances (e.g., apartment floor plans as those shown in Fig. 1); there is therefore the opportunity to distill these commonalities into a model which can then be used to complete a navigation task with reduced resource requirements.

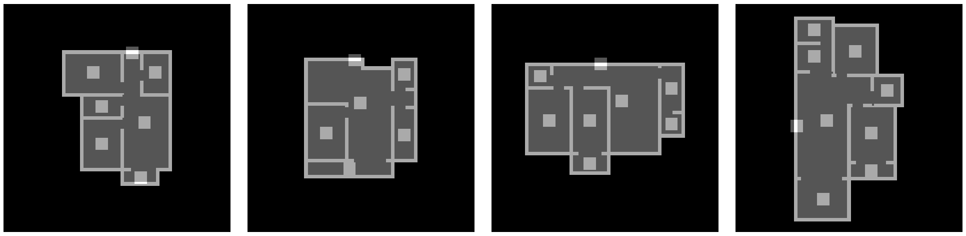


Figure 1: Floorplan examples from the RPLAN dataset (a dataset that contains residential apartments): patches indicate the centers of room and front door (on the boundary) [1].

Specifically, under this award, we have considered the problem of navigating to the exit door in an apartment. The robot is assumed to be equipped with a long-range 2-D laser scanner and a short-range exit-door detector (implemented, for instance, using a camera). The prior knowledge of the environment takes the form of an auto-encoder deep neural network that predicts a heat map of the exit location given a partial map of the environment.

2 Overview of the work

We proposed a method based on three steps (see Fig. 2 for an overview):

Mapping At each step of the algorithm, measurements (laser scans and, if deemed necessary, camera detections) are acquired and accumulated into two occupancy maps: one for the environment, and one for the exit.

Decision A local crop of the map is used as input to our trained neural network, which produces an estimate of the exit location (as the mode of a heat map). An interim goal is then selected at the location in the known space that is closest to the predicted goal.

[. Planning] We use the A* algorithm to find a feasible trajectory from the current position of the robot. Once the robot reaches the interim goal, it acquires a new laser scan to augment the map. If the robot is sufficiently confident of being at the exit location, a camera measurement is triggered to confirm the completion of the task.

Our method is complete (i.e., it is guaranteed to eventually complete the task)

2.1 Network architecture

To represent the prior knowledge about the environment, we use an auto-encoder architecture, as shown in Figs. 2♣ and 3. The encoder takes the concatenated laser scan and camera (exit detection) data to learn a

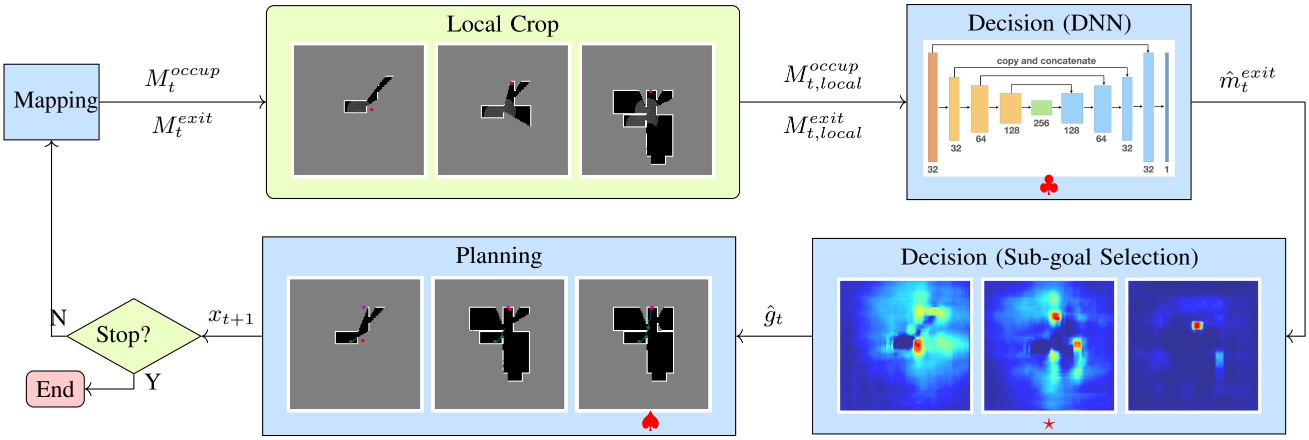


Figure 2: System workflow with main elements in blue boxes. The robot begins in an unknown environment (with the examples here occurring in that of Fig. **??**). Local Crop: a laser scan is acquired to update the occupancy grid map (three examples shown; the green dot is the current location). A camera measurement may also be acquired to update the exit location map (non-exits are superimposed in light gray). Decision: a local map is extracted and fed to a DNN (first blue box) to produce a heat map of the exit location (second blue box with a map for each example in the Local Crop box). Planning: a target location (third blue box; the red diamond is the target, and the purple star is the exit ground truth) is extracted and the robot navigates to the interim goal. The process is then iterated. (★ and ♠ will be used for a comparison in Fig. 4.)

common latent space representation of size $8 \times 8 \times 256$. Each layer uses a standard convolutional layer with Batch Normalization and Leaky ReLU.

The decoder predicts a probability map of the centroid for the exit $\hat{m}_t^{exit}$. We consider also two additional but optional decoders: an occupancy map of indoor/outdoor/walls region (optional), and a probability map of room centroids (optional). In our results (see below) we have shown that these additional decoders do not significantly change the performance in the exit navigation task, but provide additional information that can be used for other auxiliary tasks (e.g., semantic mapping).

3 Implementation and testing

3.1 Evaluation in a simplified 2-D simulator

We built a simplified simulations for the training dataset and algorithm evaluations. The mapping is performed using PseudoSLAM [2]. The laser scanner has a 360° field of view and 9m range, and the camera has a 360° field of view and 1.7m range. The environment map is a 2D occupancy map with a resolution 0.14m per pixel. Since we assume zero odometry and observation errors, the laser map updates were implemented by ray-casting. The camera map updates were implemented by adding the detector results onto the current SLAM map. We compared against a standard frontier-based exploration method that does not use any prior knowledge of the environment [3] Furthermore, to demonstrate that our algorithm is robust against at least some level of unmodeled structure in the environment, we introduced random obstacles into the environments used for testing (Fig. 4).

The results are shown in Table 1.

The results show that:

- Our method is always more efficient than traditional uninformed frontier-based exploration.
- The use of camera measurements guarantees the correct termination of the search. Moreover, there is a trade-off between the number of measurements and the number of steps necessary to complete the search.
- The introduction of obstacles increases the traveled distance, but increases the number of camera

Table 1: Method performance comparison in terms of average steps (distance traversed before stopping) and average number of camera triggers. Also shown are the *(median, standard deviation)* over all runs. ↓ indicates that smaller value implies better performance. C.R. refers to the camera range, obst. refers to the inclusion of obstacles in the environment.

	Avg. steps ↓	Avg. camera triggers ↓
Frontier (C.R. 1.7m)	329.7 (277.3, 259.4)	14.22 (14.0, 8.41)
Frontier (C.R. 3.4m)	120.5 (87.6, 103.3)	4.65 (4.0, 2.79)
Method 1[†]	38.26 (35.69, 24.58)	0 (0,0)
Method 2	68.0 (59.45, 51.44)	1.66 (1.0, 0.95)
Method 3	65.67 (59.76, 44.91)	3.23 (3.0, 1.45)
Method 3 (with obstacles)	86.43 (67.51, 87.52)	3.94 (3.0, 2.38)

[†] Method 1 is not complete and has a success rate of 57%.

measurements only marginally.

3.2 Evaluation in a full 3-D dynamic simulator

To test the algorithm under more realistic conditions (a 3-D simulation with imperfect sensor measurements), we implemented our scheme in the Robot Operating System (ROS) using the Gazebo simulator with a Jackal robot model (Clearpath Robotics). Fig. 5 presents some images from one of these advanced simulations. In general, the algorithm behaved similarly as what found in 2-D simulations.

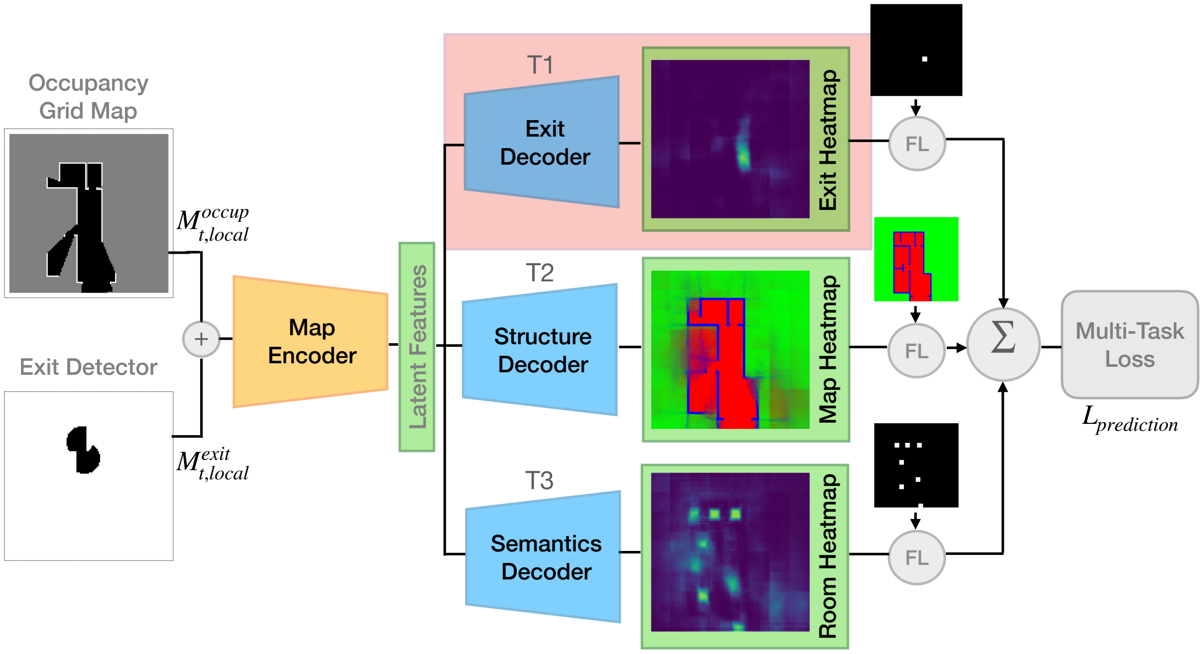
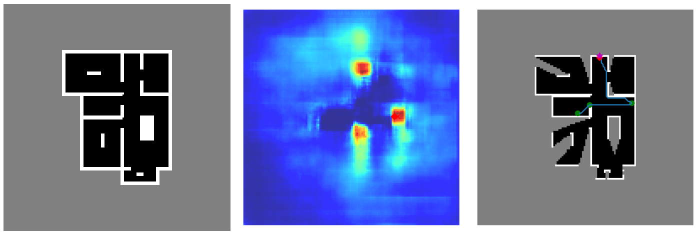
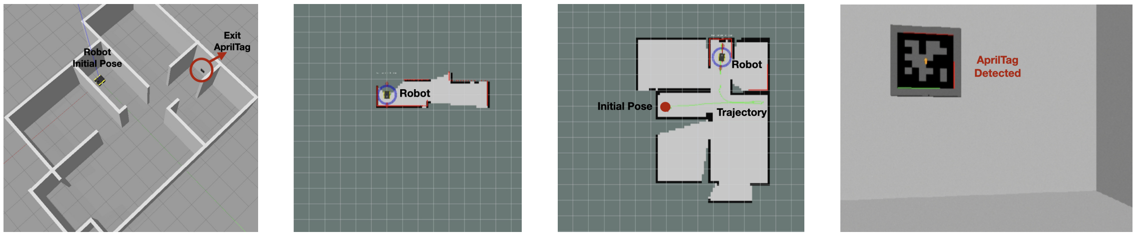


Figure 3: Network diagram: the local occupancy grid map and grip map built from the exit detector are fed into an auto-encoder to estimate the exit location in the floorplan. Without loss of performance, the additional tasks on map completion and semantic prediction can be trained simultaneously. FL means focal loss.



(a) Environment (b) Exit heat map (c) Final trajectory

Figure 4: Proposed method in an environment with obstacles. (a) The same environment as in Fig. **??** with randomly placed obstacles. (b) Exit heat map in the second iteration (compare to Fig. 2 ⋆). (c) (green line) Found trajectory to the exit location (compare to Fig. 2 ♠).



(a) Gazebo (b) Intial Map (c) Final Trajectory (d) Exit Detected

Figure 5: ROS Simulation: (a) A typical environment in Gazebo with the Jackal placed on a random initial location. (b) The occupancy grid after one scan. (c) Robot trajectory (green line) after two iterations with the robot driven to a hallucinated exit location. (d) Camera image taken at the end of the trajectory in (c) and showing the detected Apriltag.

4 Students supported and research products

The award mainly supported one student, Zili Wang (System Engineering), and also involved a MSc student, Drew Threatt (Master in Robotics and Autonomous System). The work has resulted in one workshop extended abstract and presentation [4], and a paper submission [5] (currently under review).

5 Extramural funding

The PIs have submitted a proposal to the Sony Research Award Program with title *"Efficient and Robust Top-Down Autonomous Navigation"* ($150,000), which is currently under review.

References

[1] W. Wu, X.-M. Fu, R. Tang, Y. Wang, Y.-H. Qi, and L. Liu, "Data-driven interior plan generation for residential buildings," *ACM Transactions on Graphics*, vol. 38, no. 6, pp. 1–12, 2019.

[2] T. Li, D. Ho, C. Li, D. Zhu, C. Wang, and M. Q.-H. Meng, "Houseexpo: A large-scale 2d indoor layout dataset for learning-based algorithms on mobile robots," in *IEEE International Conference on Intelligent Robots and Systems*, 2020, pp. 5839–5846.

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

[4] Z. Wang, S. B. Andersson, and R. Tron, "Task-driven navigation: Leveraging experience using deep learning," in *Workshop on Robotic Perception and Mapping: Emerging Techniques, IEEE International Conference on Robotics and Automation*, 2022.

[5] Z. Wang, D. Threatt, S. B. Andersson, and R. Tron, “Task-driven navigation: Leveraging experience using deep learning,” in *IEEE International Conference on Robotics and Automation*, 2023 (submitted).