Goal-driven Autonomous Exploration

Team #: 1

Name: Albert Chih-Hen Lee / Yize Liu / Jiatong Yao / Sean Bang-Li Weng

Background

The topic of robots exploring and mapping the environment has been prevalent in these years.

In this project, we focused on the navigation system in daily scenarios and aimed to validate the effectiveness of the method in paper "Goal-Driven Autonomous Exploration Through Deep Reinforcement Learning".

Goal-Driven Autonomous Exploration Through Deep Reinforcement Learning

Reinis Cimurs, Il Hong Suh, *Fellow, IEEE*, and Jin Han Lee

Related Work

SLAM: Simultaneous Localization and Mapping, which is an attempt to solve localization and mapping problems at the same time. In the paper, it talks about several SLAM methods like EKF(Extended Kalman Filter)-SLAM, FastSLAM 1.0 and other Well-Known Nonlinear Filtering Methods.

However, EKF/FastSLAM 1.0 or other methods have their limitations.

Overall, the expense of collecting sufficient information for the SLAM is significant due to the cost of devices, the time consumed, and the hazardous environment for gathering data.

Introduction

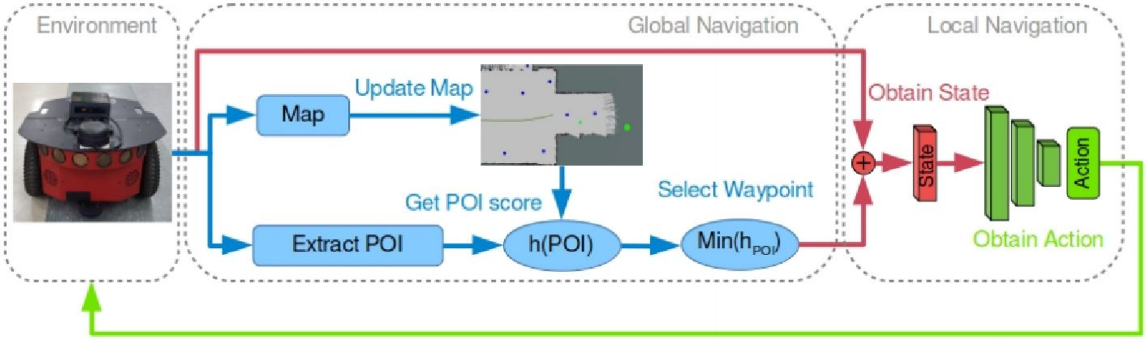
Two major concerns:

1. how to guide the robots to reach the global goal with an optimal path in the conditions of lacking comprehensive information beforehand
2. local optimal decision based on the data gathered from the sensors in real-time

To find a compromise between the global optimum solution and the expense of establishing the database of SLAM, we will use and validate the approach which introduces the deep reinforcement learning (DRL) in an unknown environment and the architecture of Twin Delayed Deep Deterministic Policy Gradient (TD3) to deal with the local optimum problem in the limited conditions.

Method

The structure for fully autonomous goal-driven exploration.



Global Navigation: optimal waypoint selection from POI(point of interests) and mapping.
The exploration robot needs to make a decision on where to go and how to have the highest possibility of arriving at the global goal.

Local Navigation: a deep reinforcement learning-based local navigation.
Give a optimal robot motion policy with TD3.

Method

Global Navigation waypoint selection strategy

available POI selection:

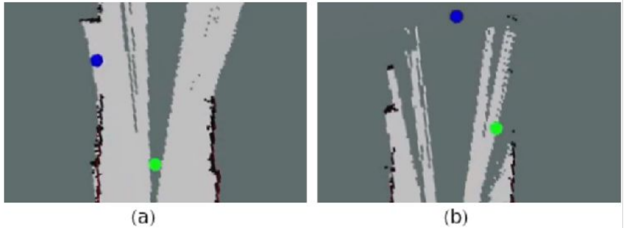


Fig. 2. POI extraction from the environment. Blue circles represent an extracted POI with the respective method. Green circles represent the current waypoint. (a) Blue POI is obtained from a gap between laser readings. (b) Blue POI extracted from non-numerical laser readings.

According the sequential laser readings,
a POI is added when it navigation through a presumed gap when two sequential laser readings is larger than a threshold in (a);

a POI is also added as a free space when laser reading outside the maximum range and return a non-numerical value in(b).

optimal waypoint selection from available POI:

Information-based Distance Limited Exploration (IDLE) evaluation method.
A POI with the smallest IDLE score is selected as the optimal waypoint.

IDLE score

at the time step t score *h* of each candidate POI *c* with index *i*

$$h(c_i) = \boxed{tanh\left(\frac{e^{\left(\frac{d(p_t,c_i)}{l_2-l_1}\right)^2}}{e^{\left(\frac{l_2}{l_2-l_1}\right)^2}}\right) l_2} + \boxed{d(c_i,g)} + \boxed{e^{I_{i,t}},}$$

map information score

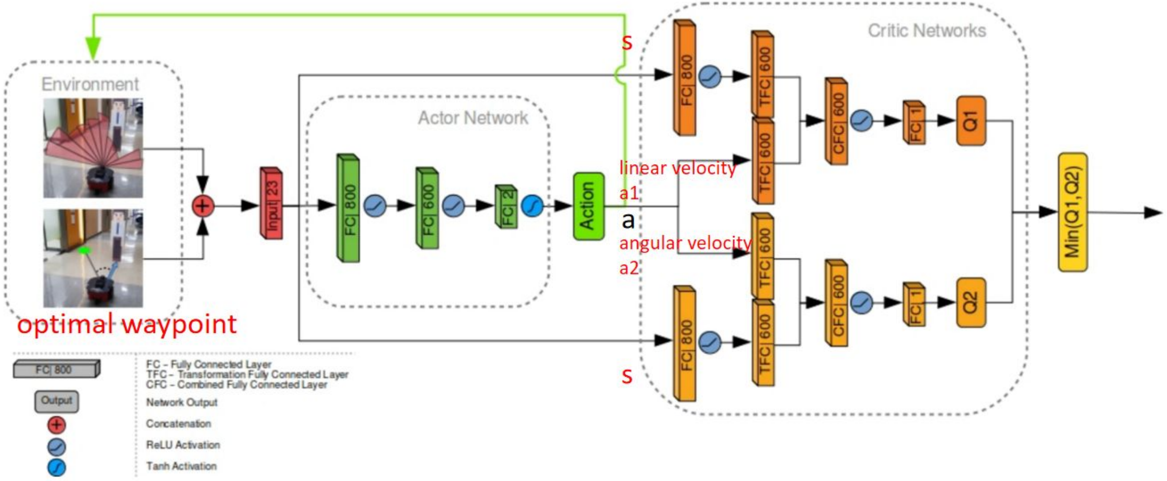
The Euclidean distance component between robots position *p* at *t* and candidate POI is expressed as a hyperbolic tangent *tanh* function.

Euclidean distance between the candidate and the global goal *g*

Method

Local Navigation

TD3 network structure including the actor and critic parts.



Input data:
the laser readings is combined with polar coordinates of the waypoint with respect to the robot's position.

The combined data is used as an input state *s* in the actor-network of the TD3.

The actor-network consists of two fully connected (FC) layers. ReLU activation follows after.
The last layer is then connected to the output layer with
two action parameters *a* that represent the linear velocity a_1 and angular velocity a_2 of the robot.

Two critic-networks have the same structure but their parameter updates are delayed allowing for divergence in parameter values.

The minimum *Q* value of both critic-networks is selected as the final critic output to limit the overestimation of the state-action pair value.

Experiments

Introduction

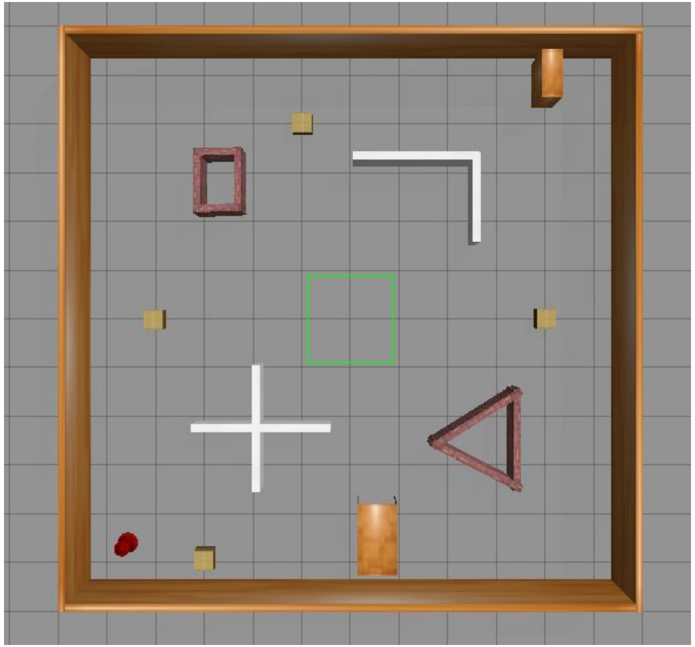
Goal: Train the model on different maps to validate its effectiveness

Environment: Ubuntu 20.04, ROS noetic, Gazebo

Hardware: NVIDIA GTX 1660 graphics card, 16 GB of RAM, and AMD Ryzen 5 3600 6-core Processor CPU

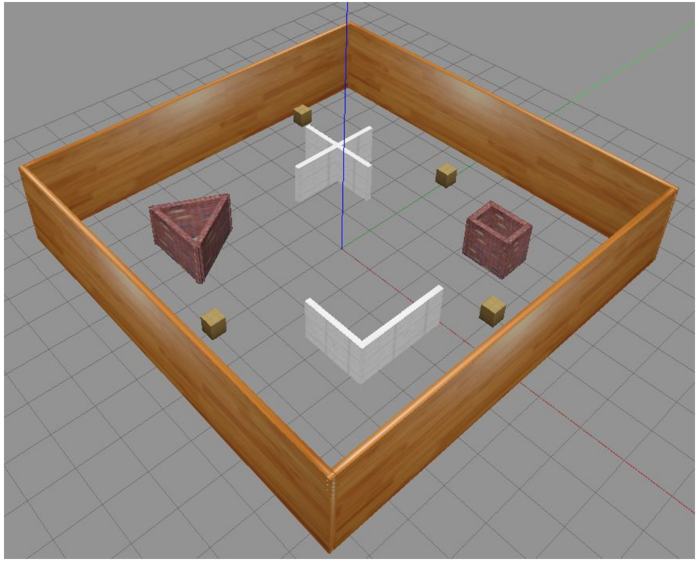
Experiments

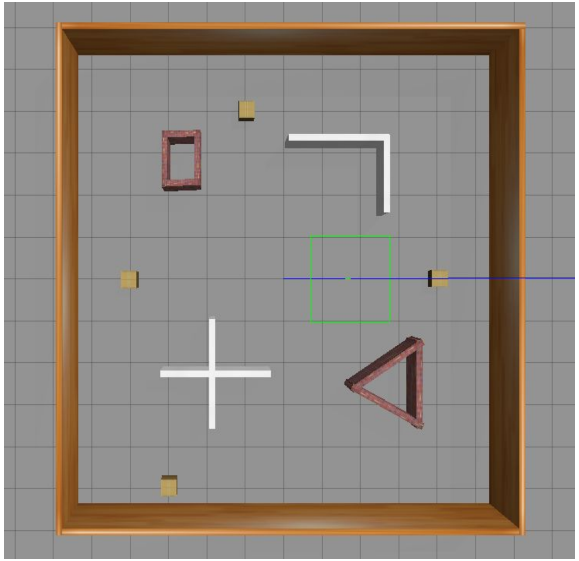
Map 1 - Default map provided by the authors



Experiments

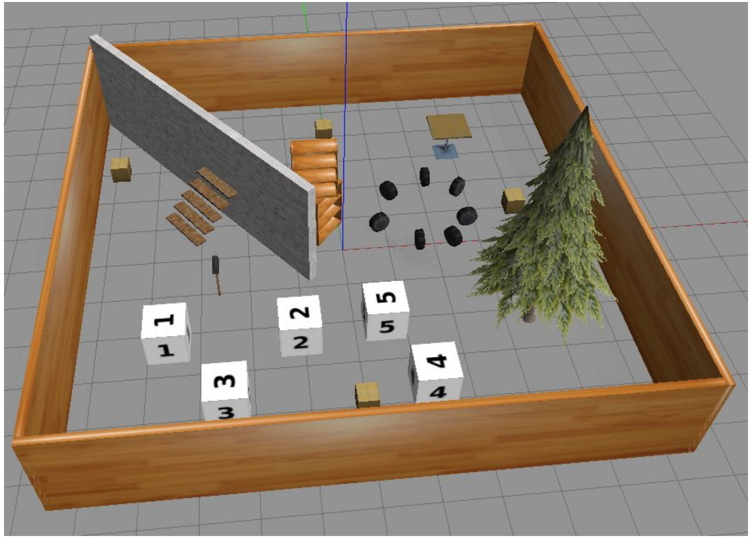
Map 2 - Slight modification from Map 1

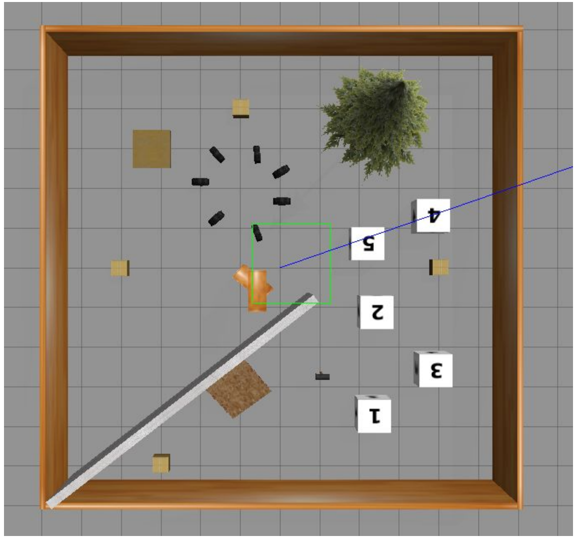




Experiments

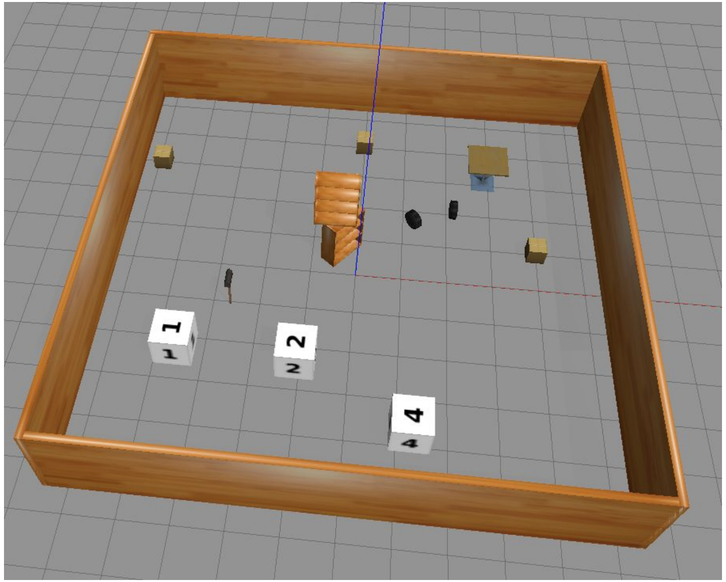
Map 3 - Complex map with stairs, tree, etc.

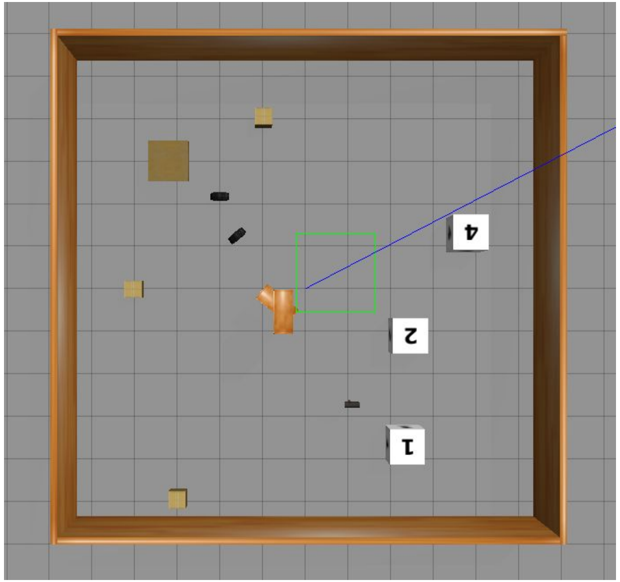




Experiments

Map 4 - Removing some large elements from Map 3

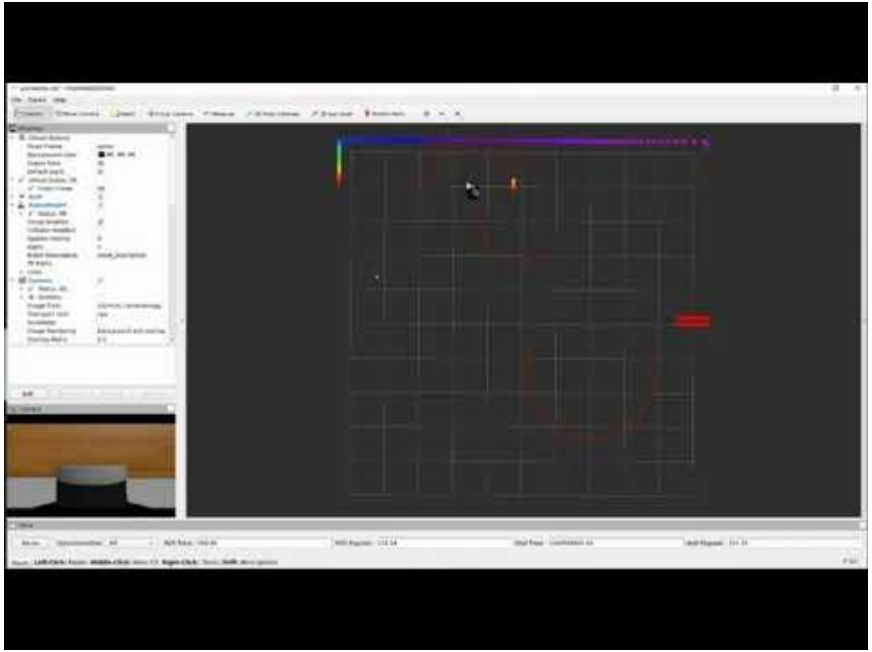




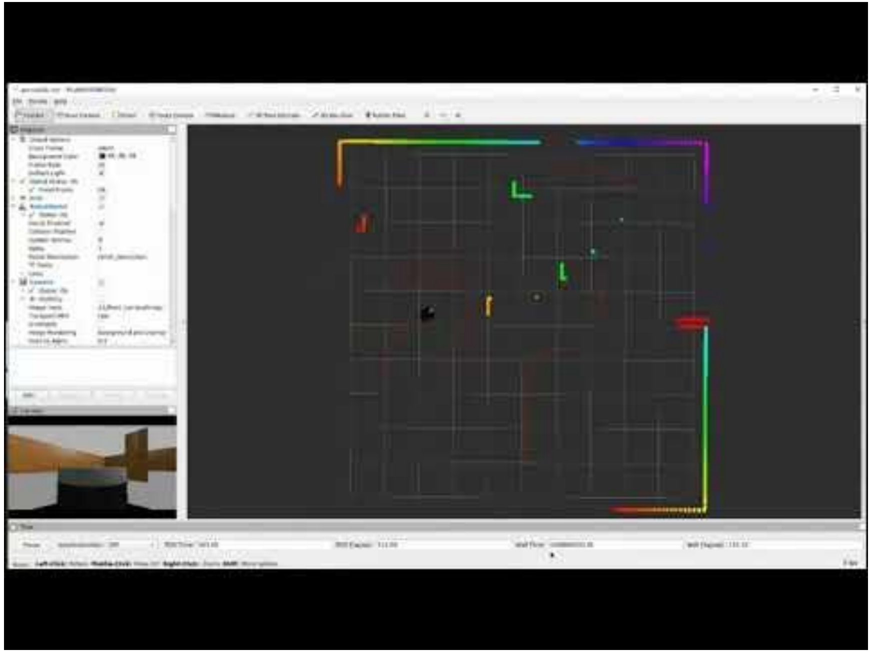
Experiments

Demo

An episode from training period

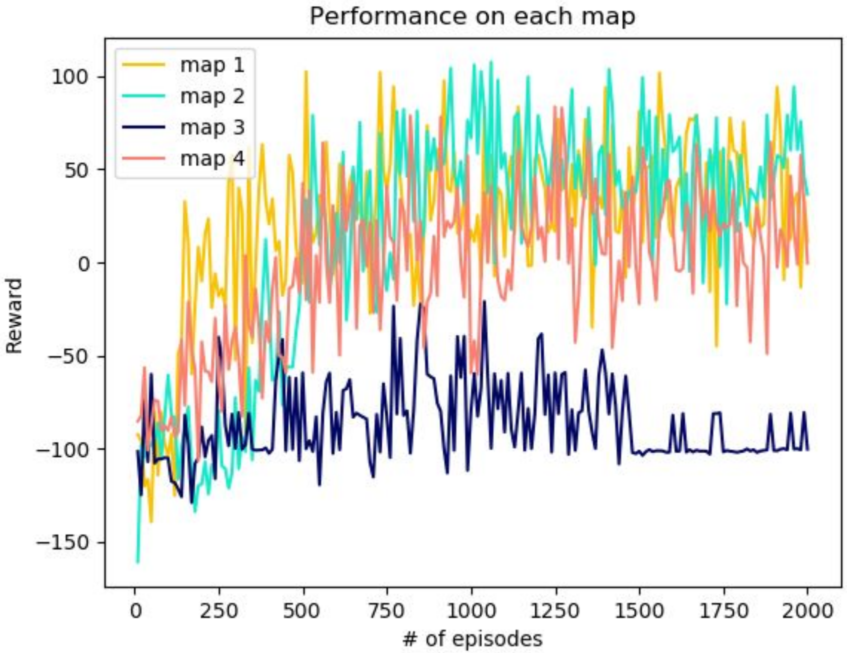


An episode from testing period



Experiments

Results



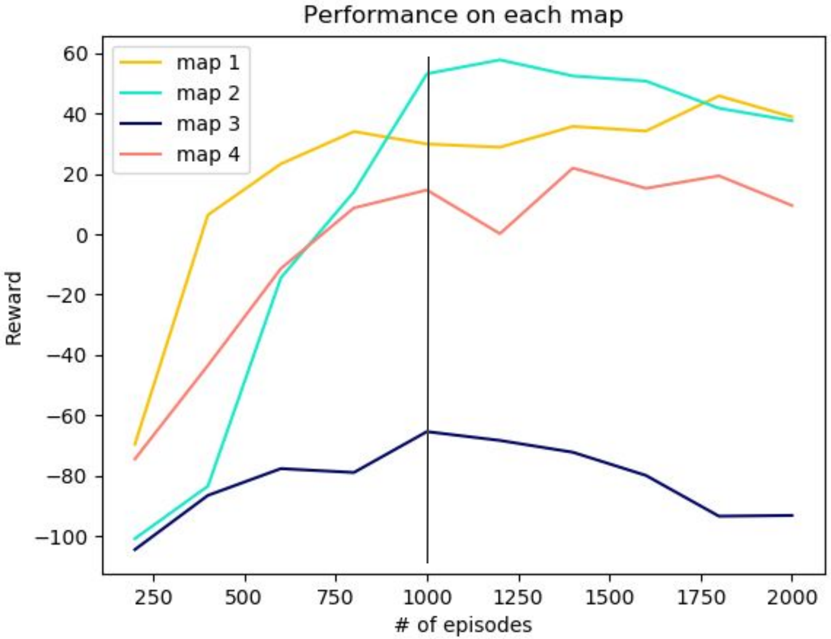
Average reward:

map 1 > map 2 > map 4 >> map 3

The model performs better on the maps with less small obstacles

Experiments

Results



The authors claim that it took about 8 hours to train 800 episodes, while it took us about 19 hours.

For 10x10 meter maps, we think that 1,000 episodes could be a good choice

Summary

- The navigation system for the robot without prior knowledge of surroundings can be established based on the IDLE evaluation for POI selection and DRL algorithm for the decision of the robot motion.

- The evaluation of the POI selection consists of three components, the function for giving a score based on the DRL training environment, the calculation of Euclidean distance between the current position and gaol, score for each candidate of POI.

- The experiment results show the reward will significantly go to the territory of negative values in a harsh environment, e.g., a space constrained by massive obstacles.

- The second experiments also suggest a possible number of episodes that can be the optimal time threshold for this algorithm.

Conclusion

- We found out this algorithm takes more time than we expected and described in the paper, especially when there is less free space for the robot to find the path toward the goal.
- We also see the reward value tends to stay low reward value when the robot encounters a challenging situation for exploration.
- One possible factor causing this result may be the design of the reward policy.
- We may give momentum value to the reward values among the POI when the system happens to be in a stuck-like environment.
- The one with the most significant reward value in an environment where all POI has negative reward values can receive an extra positive reward so that we can increase the chance and reduce the time consumption for the robot to find the way out.
- Suppose the memory and the hardware of the system are allowed. In that case, we may determine several POI as the following candidates and grant them more significant rewards to deal with this situation.

$$r(s_t, a_t) = \begin{cases} r_g & \text{if } D_t < \eta_D \\ r_c & \text{if collision} \\ v - |\omega| & \text{otherwise,} \end{cases}$$

$$r_{t-i} = r(s_{t-i}, a_{t-i}) + \frac{r_g}{i}, \qquad \forall i = \{1, 2, 3, ..., n\}$$

References

[1] R. Cimurs, I. H. Suh, and J. H. Lee, “Goal-driven autonomous exploration through deep reinforcement learning,” IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 730–737, 2022.

[2] A. B.-H. Khalid Yousif and R. Hoseinnezhad, “An overview to visual odometry and visual slam: Applications to mobile robotics.” Intelligent Industrial Systems, 2015.

[3] A. N. Hartmut Surmann and J. Hertzberg, “An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments,” 2003, pp. 45(3–4):181–198.

[4] M. Sugiyama, “Statistical reinforcement learning: modern machine learning approaches.” CRC Press, 2015.

[5] D. A. et al., “Policy-gradient algorithms for partially observable markov decision processes.” The Australian National University, 2003.

[6] X.-J. L. F. X. Y. Y. Q. W. Chao Yu, Zuxin Liu and Q. Fei, “Ds- slam: A semantic visual slam towards dynamic environments,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, p. 1168–1174.

[7] G. L. W. X. Guolai Jiang, Lei Yin and Y. Ou, “Fft-based scan-matching for slam applications with low-cost laser range finders,” 2019, p. 9(1):41.

[8] P. G. Marek Pierzchała and R. Astrup, “Mapping forests using an unmanned ground vehicle with 3d lidar and graphslam,” 2018, p. 145:217–225.

[9] P.-T. W. Shao-Hung Chan and L.-C. Fu, “Robust 2d indoor localiza- tion through laser slam and visual slam fusion.” IEEE, 2018, p. 1263–1268.

[10] M. Filipenko and I. Afanasyev, “Comparison of various slam systems for mobile robot in an indoor environment.” IEEE, 2018, p. 400–407

THE END

Questions?