Data Driven Strategies for Active Monocular SLAM using Inverse Reinforcement Learning

(Extended Abstract)

Vignesh Prasad
Robotics Research Centre
International Institute of Information Technology
Hyderabad, India

Rishabh Jangir
Department of Physics
Indian Institute of Technology Guwahati
Guwahati, India

Ravindran Balaraman
Department of Computer Science
Indian Institute of Technology Madras
Madras, India

K. Madhava Krishna
Robotics Research Centre
International Institute of Information Technology
Hyderabad, India

ABSTRACT

Learning a complex task like robot maneuver while preventing Monocular SLAM failure is challenging for both robots and humans. We devise a computational model for representing and inferring strategies for this task, formulated as a Markov Decision Process (MDP). We show how the reward function can be learned using Inverse Reinforcement Learning. The resulting framework allows us to understand how chosen parameters affect the quality of Monocular SLAM. A significant improvement in performance as compared to other state-of-the-art methods is also shown.

Keywords

Active Monocular SLAM; Inverse Reinforcement Learning

1. INTRODUCTION

Active Simultaneous Localization and Mapping (Active SLAM), deals with the generation of controls for a robot moving in an unknown environment while simultaneously mapping the environment and localizing itself. Most works in this area ([1, 2, 3, 4, 5, 6, 7]) assume the availability of dense range data or depth maps. Monocular SLAM methods on the other hand provide sparse maps and are susceptible to errors in pose estimates due to insufficient visual tracking or motion induced errors.

Literature that talks about Active Monocular SLAM is sparse. There have been works demonstrating Autonomous Navigation for Micro Aerial Vehicles (MAVs) with Monocular SLAM [8, 9]. We approach the problem for non-holonomic robots, which is more constrained than using MAVs. Recent work shows the use of Reinforcement Learning to do so [10].

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

The above mentioned methods are hand crafted and may not accurately capture the importance of the parameters used.Hence, we formulate an Inverse Reinforcement Learning (IRL) model, that learns behavior which performs even more favorably than the above mentioned works. The main focus here is not to introduce a new IRL method, rather to apply existing methods to solve a challenging problem.

2. INVERSE REINFORCEMENT LEARNING

The problem of learning a reward function for an MDP led to the emergence of IRL methods [11, 12, 13, 14, 15] under the umbrella of Learning from demonstration frameworks. The algorithm that we follow for performing IRL can be found in detail in [12].

Let $\boldsymbol{\phi} : S \times A \times S \rightarrow [0,1]^n$ be a parametrization of state-action pairs. We assume that the reward function is a weighted combination of these parameters given by

$$R(s,a,s') = \boldsymbol{\omega}^T \boldsymbol{\phi}(s,a,s') \tag{1}$$

Given a policy π, its feature expectation $\boldsymbol{\mu}(\pi)$, can be expressed as

$$\boldsymbol{\mu}(\pi) = \sum_{t=0}^{\infty} \gamma^t \boldsymbol{\phi}(s_t, a_t, s_{t+1}) \tag{2}$$

Given the feature expectation of an expert agent $\boldsymbol{\mu}(\pi_E)$, IRL tries to find weights that resemble the reward function the expert demonstrator is trying to maximize.

3. REWARD FUNCTION PARAMETERS

Failure in Monocular SLAM systems usually occurs when we enter areas of low feature density. Large rotations without adequate translation also add to the deterioration of pose estimates. When performing Monocular SLAM, multiple sequences of subsequent forward and backward motions are executed to give differing viewpoints from which similar parts of the scene can be viewed, thereby improving the quality of the map and consequently, the pose estimate.

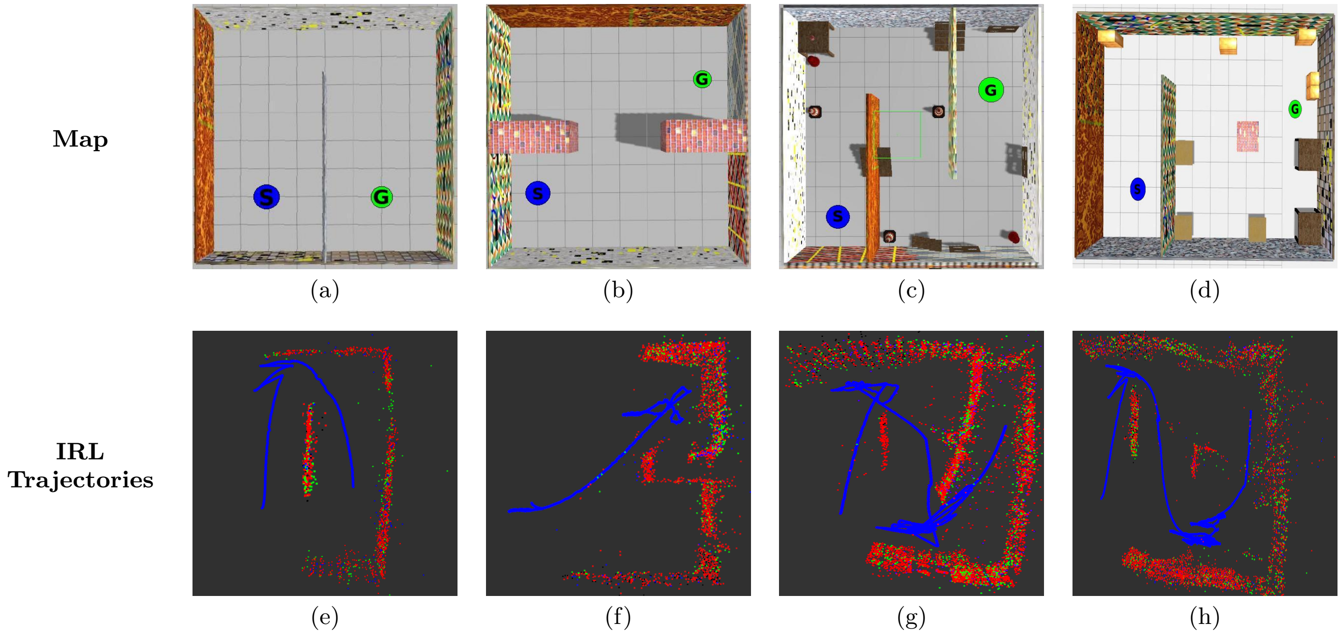

Figure 1: Results of Goal based trajectory navigation. The first row shows the maps with start and goal locations shown as blue and green circles, marked as "S" and "G" respectively. The second row shows the Trajectory and Map estimates using IRL for navigation.

Keeping in mind the above points, the parameters that we have considered are the direction of motion, angle change $\Delta\theta$ and the common features seen between subsequent views, denoted as ΔFOV. One additional feature that we have considered is the SLAM failure itself after an action is executed, which is obtained as a feedback from the SLAM.

4. EXPERIMENTATION AND RESULTS

Gazebo [16] is a framework that accurately simulates robots and dynamic environments. Experiments were performed in simulated environmentson a Turtlebot using a Microsoft Kinect for the RGB camera input. We use **PTAM (Parallel Tracking and Mapping)** [17] for the Monocular SLAM framework. The Q-values are learnt offline between every IRL iteration using Q-learning [18, 19] and are interpolated with Stochastic Gradient Descent Regression, implemented using scikit-learn [20]. We use 5th order Bernstein curves for trajectory planning [21]. Experiments were carried out on a laptop with Intel Core i7-5500U 2.40GHz CPU running Ubuntu 14.04 using Robot Operating System (ROS) [22] for controlling the robot and performing SLAM. The IRL algorithm terminates after around 6-7 iterations on an average. The weights obtained, shown in table 1, are quite intuitive and capture the way the parameters affect Monocular SLAM.

Table 1: Weights obtained from the IRL algorithm

Features	Backward	Forward	$\Delta\theta$	ΔFOV	SLAM Failure
Weights	0.0801	-0.1831	-0.4698	0.3127	-0.8009

To verify the usefulness of the learnt weights, we use two different criteria. The first is the average number of steps executed till PTAM failure, the results of which are shown in table 2. The percentages refer to the exploitation ratio.

Table 2: Average no. of steps executed till PTAM failure

RL 60%	RL 80%	RL 95%	IRL 95%
80	95	112	174

The second is navigation between start and goal locations in various maps. During navigation, we continuously check if the subsequent part of the trajectory would lead to a SLAM failure, by thresholding the Q-value of an action and performing recovery actions in case a failure is detected. Table 3 summarizes the results of our goal based navigation experiments which can be seen qualitatively in Fig. 1.

Table 3: Results for Goal Based Trajectory Planning

Map	Planner Type	Runs	Success	Failures	Success %
1	RL	10	9	1	90
1	IRL	10	10	0	100
2	RL	10	8	2	80
2	IRL	10	9	1	90
3	RL	10	8	2	80
3	IRL	10	7	3	70
4	RL	10	6	4	60
4	IRL	10	8	2	80

5. CONCLUSION

Automating Monocular SLAM has been a significantly challenging problem to solve as failures are common even if the camera is carefully moved or teleoperated by an expert. This paper proposes a novel data driven strategy for learning handcrafted expert behavior. The proposed strategy learns such expert intuited policies and outperforms the expert through enhanced SLAM longevity and goal reaching behavior on a variety of maps.

REFERENCES

[1] Cindy Leung, Shoudong Huang, and Gamini Dissanayake. Active slam using model predictive control and attractor based exploration. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5026–5031. IEEE, 2006.

[2] Cindy Leung, Shoudong Huang, Ngai Kwok, and Gamini Dissanayake. Planning under uncertainty using model predictive control for information gathering. *Robotics and Autonomous Systems*, 54(11):898–910, 2006.

[3] Thomas Kollar and Nicholas Roy. Efficient optimization of information-theoretic exploration in slam. In *AAAI*, volume 8, pages 1369–1375, 2008.

[4] Vadim Indelman, Luca Carlone, and Frank Dellaert. Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *The International Journal of Robotics Research*, 34(7):849–882, 2015.

[5] Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Ken Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar. Information-theoretic planning with trajectory optimization for dense 3d mapping. In *Robotics: Science and Systems*, 2015.

[6] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. isam: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

[7] Gabriele Costante, Christian Forster, Jeffrey Delmerico, Paolo Valigi, and Davide Scaramuzza. Perception-aware path planning. *arXiv preprint arXiv:1605.04151*, 2016.

[8] Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011.

[9] Christian Mostegel, Andreas Wendel, and Horst Bischof. Active monocular localization: Towards autonomous monocular exploration for multirotor mavs. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3848–3855. IEEE, 2014.

[10] Vignesh Prasad, Saurabh Singh, Nahas Pareekutty, Balaraman Ravindran, and Madhava Krishna. Slam-safe planner: Preventing monocular slam failure using reinforcement learning. *arXiv preprint arXiv:1607.07558*, 2016.

[11] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.

[12] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[13] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736. ACM, 2006.

[14] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. *Urbana*, 51(61801):1–4, 2007.

[15] Yang Gao, Jan Peters, Antonios Tsourdos, Shao Zhifei, and Er Meng Joo. A survey of inverse reinforcement learning techniques. *International Journal of Intelligent Computing and Cybernetics*, 5(3):293–311, 2012.

[16] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE, 2004.

[17] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[18] Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards*. PhD thesis, University of Cambridge England, 1989.

[19] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[20] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[21] Bharath Gopalakrishnan, Arun Kumar Singh, and K Madhava Krishna. Time scaled collision cone based trajectory optimization approach for reactive planning in dynamic environments. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 4169–4176. IEEE, 2014.

[22] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, 2009.