Real-time Robot Path Planning Using Experience Learning From Common Obstacle Patterns

(Extended Abstract)

Olimpiya Saha
University of Nebraska at Omaha
6001 Dodge St
Omaha, NE 68182
osaha@unomaha.edu

Prithviraj Dasgupta
University of Nebraska at Omaha
6001 Dodge St
Omaha, NE 68182
pdasgupta@unomaha.edu

ABSTRACT

In this paper we investigate the problem of online robot path planning in an environment. Our main hypothesis in this paper is that the path planning times for a robot can be significantly reduced if it can refer to previous maneuvers it used to avoid collisions with common obstacles during earlier missions, and adapt that information to avoid obstacles during its current navigation. To verify this hypothesis, we propose an online path planning algorithm called LearnerRRT. Our algorithm utilizes a pattern matching technique called Sample Consensus Initial Alignment (SAC-IA) in combination with an experience based learning technique to adapt to the current scenario. We have conducted several experiments in simulations to verify the performance of LearnerRRT and compared it with a sampling-based planner Informed RRT*. Our results show that LearnerRRT performs much better than Informed RRT* in terms of planning time and total time to solve a given navigation task. When navigation times and distances traveled are explicitly compared, LearnerRRT takes slightly more navigation time and distance than Informed RRT*.

Categories and Subject Descriptors

Computing methodologies, Machine Learning, Artificial Intelligence

General Terms

Algorithms, Performance, Reliability

Keywords

Experience learning, Real-time robot path planning, Obstacle feature matching

1. INTRODUCTION

Autonomous navigation is one of the fundamental problems in robotics used in several real life applications such as unmanned search and rescue, autonomous exploration and surveillance, and domestic applications such as automated waste cleanup or vacuum cleaning. We consider the navigation problem for a robot in an unstructured environment where the location and geometry of obstacles are initially unknown or known only coarsely. To navigate in such an environment, the robot has to find a collision-free path in real-time, by determining and dynamically updating a set of way-points that connect the robot's initial position to its goal position. While there are several state-of-the-art path planners available for robot path planning [1], these planners usually replan the path to the goal from scratch every time the robot encounters an obstacle that obstructs its path to the goal - an operation that can consume considerable time (order of minutes or even hours), if the environment is complex, with many obstacles. Excessively expended path planning time also reduces the robot's energy (battery) to perform its operations, and aggravates the overall performance of the robot's mission. To address this problem, in this paper, we propose an algorithm LearnerRRT based on the insight that, although obstacles could be geometrically dissimilar in nature, yet there exists some generic features that are common across most obstacles. If a robot can be trained to navigate around obstacles with some basic geometric patterns, it can adapt and synthesize these movements to navigate around more complex obstacles without having to learn the motion around those obstacles from scratch. In this manner, navigation can be learned incrementally by a robot, without having to be trained independently for each new environment it is placed in. The features of the obstacles perceived by the robot's sensors and its movements or actions to avoid the obstacles are stored in summarized format within a repository maintained inside the robot. Our results illustrate that the state-of-the-art sampling-based planner Informed RRT* takes significantly more planning time and total time when compared to the time taken by our algorithm to solve the same navigation tasks. With respect to navigation times and distances traveled LearnerRRT on average takes marginally more navigation time and distance to solve a given task. However, we believe this difference is trivial when compared to the significant improvements observed in planning time and total time taken by the robot to achieve its navigation task in real-time.

2. SOLUTION APPROACH

Consider a wheeled robot situated within a bounded environment having obstacles. The objective of the robot in navigation tasks is to find a sequence of actions that guarantees a collision free navigation path connecting the start location to the goal location. In other words, no action along the navigation path should take the robot to a configuration that is in collision with an obstacle.The LearnerRRT algorithm first creates a library of obstacle patterns and robot actions and reuses the learned actions to navigate around obstacles. The proposed algorithm proceeds in two steps which are described below.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

2.1 Library Creation

The robot is first trained to find a collision-free path for navigating around obstacles that have different but well-defined geometries. Each navigation task used for training is called a source task. We assume that the environments in which the robot will perform navigation tasks later on will have obstacles with significant similarities in their boundary patterns with respect to the source tasks, although the orientation and dimensions of individual obstacles in the later environments might vary. We consider four well-defined obstacle geometry patterns as source tasks - cave, column or passage, corner and block. To construct the action library, L, for a source task corresponding to an obstacle with label lab, the robot is initially placed in front of the obstacle in such a way that the robot's range sensor can perceive the obstacle's entire boundary facing the robot. A set of goal locations, G_{lab}, corresponding to positions where the robot will have avoided the obstacle are specified. The robot internally constructs the obstacle boundary from the range data as a set of 2D coordinates $LS_{lab} = \{(\hat{x}, \hat{y})\}$ and uses the Informed RRT* path planner [2] to plan a path to each of the goal locations. The path returned by Informed RRT* along with LS_{lab} are stored in the library.

2.2 Obstacle Avoidance Using Learned Navigation Actions

After learning actions for avoiding common obstacle patterns, the robot is given a navigation task. Note that the new navigation task can be given in a different environment than the one in which the action library was created. The general scheme that the robot uses after constructing its action library is to reuse actions from its action library, after suitable adaptations, when it encounters an obstacle while navigating towards the goal. When the robot encounters an obstacle obs it first records the proximity data from the obstacle, as a set of coordinates in the 2D plane $LS_{obs} = \{(\hat{x}, \hat{y})\}$. The robot first preprocesses the data to scale it approximately to match LS_{lab}. It then uses a state-of-the-art algorithm for aligning object features called Sample Consensus Initial Alignment(SAC-IA) algorithm [3], to match LS_{obs} with the obstacle proximity data for the different obstacles recorded in the action library L. The algorithm returns the corresponding transformation between the proximity data and the library data. The extent of match between LS_{obs} and LS_{lab} calculated by SAC-IA is measured by analyzing their Jaccard Index JI, which reflects the extent of overlap between two geometric shapes. Once the best obstacle match in the action library lab_{match} has been determined, the robot retrieves the set of paths from the library, applies the transformation to each of them, followed by applying the same scaling factor calculated during preprocessing. It then selects the adapted path which minimizes the distance to the goal. Finally, the robot does two post-processing steps to correct and optimize the selected path.

3. EXPERIMENTAL SETUP AND RESULTS

We have verified the performance of the LearnerRRT algorithm using simulated Corobot robots on the Webots version 6.3.2 simulator and compared the performance with Informed RRT*. For comparing the performance of our algorithm with Informed RRT*, we have primarily used three main measures- the planning time to predict the path to be undertaken by the robot, the total time that the robot requires in order to traverse the entire arena and lastly we also compared the total distance that the robot navigated in each of the cases in order to reach the goal. Different test cases have been created by selecting different start-goal pairs in the simulated environments. We have selected the start-goal pairs for our test cases in

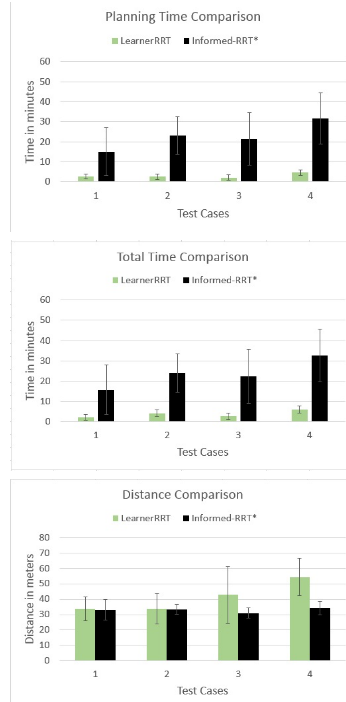

Figure 1: Planning time, total time and distances traveled for the test cases

such a manner that the direct path connecting the start to the goal contains maximum number of obstacles in it. This means that the robot had to replan its path multiple times while navigating from the start to the goal. In order to make the test cases representative, we have selected start and goal locations from different regions of the environments. Figure 1 shows the comparative planning time, total time and distance traveled by the robot for solving the test cases by following our LearnerRRT algorithm and Informed RRT*. From our experimental results it can be observed that the planning time and total time taken by LearnerRRT is much lesser when compared to Informed RRT*. When compared with the mean of the planning times and total times for all the test cases across all environments, LearnerRRT was found to beat Informed RRT* approximately by a factor of 10 for planning time and a factor of 7 for total time.

REFERENCES

[1] H. Choset, W. Burgard, S. Hutchinson, G. Kantor, L. E. Kavraki, K. Lynch, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press, June 2005.

[2] J. Gammell, S. Srinivasa, and T. Barfoot. Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2997–3004, Sept 2014.

[3] R. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3212–3217, May 2009.