Applied Mechanics and Materials Vol. 151 (2012) pp 493-497 *Online: 2012-01-24*
© (2012) Trans Tech Publications, Switzerland
doi:10.4028/www.scientific.net/AMM.151.493

EXTENDING RRT FOR ROBOT MOTION PLANNING WITH SLAM

Hai Zhu Pan [1, a], Jin Xue Zhang [2,b]

[1] Department of Computer ,Qiqihar University, China

[2] Department of Electronic Engineering,Huaihai Institute of Technology,China

[a]Pan_haizhu@163.com , [b]greyzjx@gmail.com

Keywords: Motion planning,control,RRT,SLAM,MobileSim

Abstract. In this paper,the motion planning problem for mobile robot is addressed. Motion planning (MP) has diversified over the past few decades to include many different approaches such as cell decomposition, road maps, potential fields, and genetic algorithms. Often the goal of motion planning is not just obstacle avoidance but optimization of certain parameters as well. A motion planning algorithms based on Rapidly-exploring random Tree(RRT) is present in the paper. Then the RRT algorithm has been extended which combines the SLAM algorithm.The Extend-RRT-SLAM has been simulated in MobileSim.Simulation results show Extend-RRT-SLAM to be very effective for robot motion planning.

Introduction

Motion planning for robots is a fundamental problem that received more attention in recent years[1][2],and many efforts have been conducted in robotics research for solving the fundamental problem of motion planning. The planning of motion and its realization was analysed by lots of authors (Arkin, 1998; Berenstain and Koren, 1989; Latombe,1991). Motion planning problem has its origins in robotics, where the ultimate goal is true autonomous robots. Motion planning for robots is basically the problem of given an initial and a final configuration of the robot in its workspace, find a path, starting at the initial configuration and terminating at the goal configuration, while avoiding collisions with the obstacles.

Motion planning is often decomposed into path planning and trajectory planning.Path planning is to generate a collision free path in an environment with obstacles and optimize it with respect to some criterion.Trajectory planning is to schedule the movement of a mobile robot along the planned path.Several approaches have been proposed to address the problem of motion planning of a mobile robot.

This paper is structured as follows: Section 2 gives a brief description of motion planning .Section 3 describes Rapidly-exploring Random Tree (RRT) algorithm and Extend-RRT-SLAM algorithm is proposed.Section 4 provides the simulation results. In Section 5 some summaries are given.

Motion planning

Motion planning emerged as a crucial and productive research area in robotics in the late 1960’s [1]. Motion planning, considering robot control or motion and sensing uncertainties for mobile robots, has been studied for decades. Motion planning is the process of selecting a motion and the associated set of input forces and torques from the set of all possible motions while ensuring all the constraints are satisfied. This process can be viewed as a set of computations typically performed offline that provide sub goals or set points for robot control. The computations and the resulting motion plans are based on a suitable model of the robot and its environment.

Motion planning algorithms can be classified into explicit motion planning and implicit motion planning [3]. Explicit motion planning approaches are by far the most common place approaches wherein the motion planning task is decomposed into three steps: path planning-trajectory planning robot control, as depicted in Fig.1.

Fig. 1: The traditional, explicit motion planning decomposed the motion planning task into three separate tasks.

Definition (Motion Planning).

The objective of motion planning is to compute a trajectory that enables the robot to accomplish the assigned task while satisfying constraints such as collision avoidance, and velocity and acceleration bounds along the trajectory. Fig.2 provides an illustration. A formal definition follows.

A motion-planning problem is a tuple $P = (S, Valid, s_{init}, Goal)$, where S is a state space consisting of a finite set of variables that completely describe the state of the system and $Valid : S \rightarrow \{\mathrm{T}, \perp\}$ is a state-constraint function, which satisfies the state constraints. The motion planning goal is specified as desired constraints that a goal state should satisfy. Such constraints could include a desired position or orientation. In motion planning with dynamics, it is also common to require that a robot's velocity remains within certain bounds.

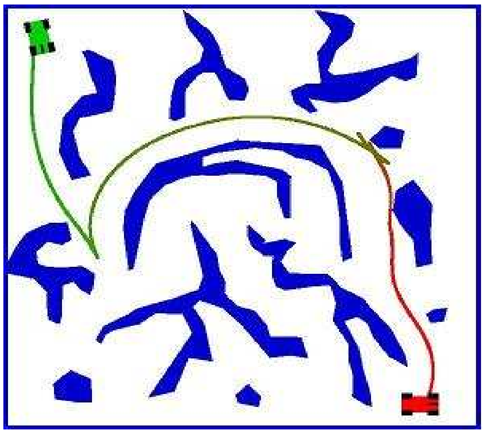

Fig. 2: In this motion-planning problem the objective is to find a trajectory that allows the robotic car to reach the goal position starting while avoiding collisions with the obstacles.

Algorithm Description

Rapidly-exploring Random Tree (RRT)

The Rapidly-exploring Random Tree (RRT) [4] is a well known randomized algorithm to explore large state space in a relatively short time. The pseudocode of the algorithm is given in Algorithm 1.RRT algorithms grow a tree of dynamically feasible trajectories by sampling numerous points randomly.

RRT expands a tree in C_{free} which generates tree nodes. The RRT algorithm attempts to find a continuous path through C_{free} from the initial state of the system q_{init} to some state $q_{goal} \in G$, a set of valid goal states.

The Rapidly-exploring Random Tree (RRT) algorithm was originally developed for robot motion planning in high dimensional configuration spaces. RRT has been applied to plan motions for different system setups, such as rigid body, manipulator system of many degrees of freedom [5],nonholonomic systems,and systems with dynamic contrains[6].As pointed out in[2] ,the distance metric applied plays a crucial role in RRT.

We use $[x \quad y \quad \theta]$ to denote a mobile robot base pose at position x and y, with orientation of θ. Denote m as the environment model. The control algorithm use a kinematic model of the mobile robot is as following:

$\dot{x} = v\cos\theta$, $\dot{y} = v\sin\theta$, $\dot{\theta} = \omega$

Where in the equation v is longitudinal velocity, ω is angular velocity.

Algorithm 1:Classic RRT

Input: q_{init} - initial configuration.
d_{goal} -a distance threshold.
q_{goal} -goal configuration.
d_{step} -a user defined fixed step.
Output:tree-a random tree.

begin
  while !TimeUp() **do**
    q_r = RandomConfig()
    q_c = NearestNeighbor(q_r ,tree)
    q_{new} = ExtendTo(q_c , q_r , d_{step})
    q_{new} = Extend-Classic-RRT(tree, q_{new} , q_c)
    if $q_{new} \neq$ NULL **then**
      if $\rho(q_{new}, q_{goal}) \leq d_{goal}$ **then**
        if Extend-Classic-RRT(tree, q_{goal} , q_{new}) **then**
          break;
end

In this paper Extend-RRT-SLAM algorithm is proposed in which we extend RRT algorithm and combine RRT with particle based SLAM algorithm. The pseudocode of the algorithm is given in Algorithm 2.In Algorithm 2 a RRT tree node,and update the collision probability of the path corresponding to the new node based on the outcome of the simulation.

Algorithm 2:Extend-RRT-SLAM

Input: n_c - a tree node.
$< q_r , u_r >$ - a node state the tree will extend from n_c towards to.
Tree – a random tree
m^* - a map.
d_{step} - a user defined fixed step length

begin
  $< \varsigma_{n_{new}} , \mathrm{T}_{n_{new}} , P_{free}^{n_{new}} >=$
    SimLocal(ς_{n_c} , T_{n_c} , m^* , $\pi_{[q_{n_s}, q_{n_{new}}]}$ $P_{free}^{n_c}$,);
  if $\varsigma_{n_{new}} \neq$ NULL and $P_{free}^{n_{new}} > P_{thresh}$ **then**
    u_{new} = UncertaintyScalar($\varsigma_{n_{new}}$);
    if $u_{new} > B_u$ **then**
      return;
    $n_{new} := \{ q_{new}, u_{new}, \varsigma_{n_{new}} , \mathrm{T}_{n_{new}} , P_{free}^{n_{new}} \}$;
    add n_{new} to tree as son of n_c;
    return n_{new}
  else
    return NULL;
end

Simulation results

In this section some simulation results were conducted.The algorithms proposed were tested in the MobileSim software.MobileSim,a mobile robot simulator,is a modified version of the Player/Stage Simulator.MobileSim is developed basded on the Stage library which is offered by MobileRobots[7]. We first manually set a sequence of intermediate goal positions, moving the robot along these positions while executing SLAM to explore the real environment. Fig.3 shows the partially constructed map.

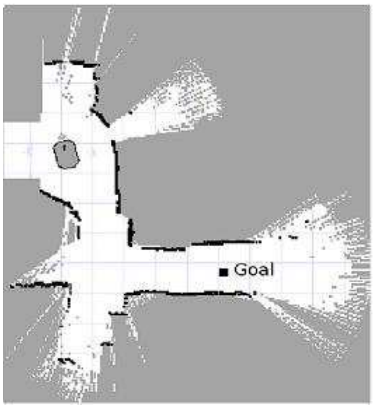

Fig. 3: A constructed map

Fig.4 and Fig.5 show the paths returned by Extend-RRT-SLAM and classic RRT (with collision probability of 0.0 and 0.21), respectively. This result clearly shows that path returned by the Extend-RRT-SLAM gives the robot more clearance (especially in the horizontal corridor area), and therefore is safer compared to the path returned by the classic RRT.

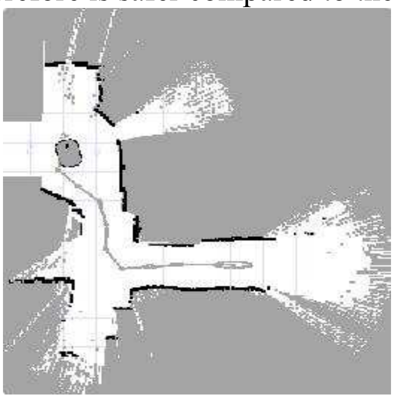

Fig. 4: A solution path returned by Extend-RRT-SLAM

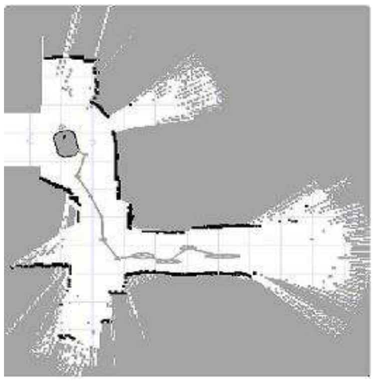

Fig. 5: A solution path returned by the classic RRT

Finally, the robot successfully executed the path planned by Extend-RRT-SLAM and reached the goal. The trajectory of the robot is plotted in Fig.6.

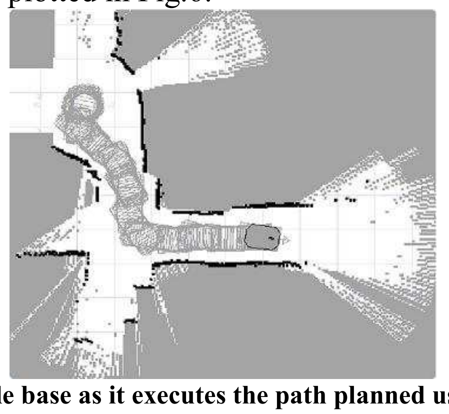

Fig. 6: Trace of the mobile base as it executes the path planned using Extend-RRT-SLAM

A different scenario, which indicates the effectiveness of our Extend-RRT-SLAM during the exploration is shown in Fig.7, where the robot needs to go across the open free area with a collision probability. The collision probabilities for the paths in Fig.7(a) and (b) are 0.0 and 0.82,respectively.

(a) A solution path returned by Extend-RRT-SLAM (b) A solution path returned by the classic RRT

Fig. 7: Comparing solution paths returned by RRT-SLAM and the classic RRT in the simulated environment.

Fig.7 shows the advantage of applying Extend-RRT-SLAM compared with the classic RRT without considering the localization uncertainty. For the path returned by the classic RRT we also simulate the localization uncertainty and calculate the expected collision probability.
Fig.7 (a) and (b) show Extend-RRT-SLAM gives a more robust path than the classic RRT.

Summary

In this paper, we present robot motion planning problem and propose an algorithm to solve mobile robot motion planning problem which is based on RRT and SLAM. In the proposed algorithm(Extend-RRT-SLAM) the simulated SLAM into each RRT extension step, where each tree node is a configuration. Some simulation results have been implemented in MobileSim platform.The effectiveness of the proposed algorithm was demonstrated by the simulation results.

Acknowledgement

The research is supported by the Program for Young Teachers Scientific Research in Qiqihar University,China No. 2010k-M16.

References

[1] J.C. Latombe. Robot Motion Planning. Kluger Academic Pub. (1991)
[2] S. M. LaValle. Planning Algorithms. [Online], 1999-2004. Available at http://msl.cs.uiuc.edu/planning/
[3] V. Kumar, M. Zefran, and J. Ostrowski, "Motion planning and control of robots," in Handbook of industrial robotics, J. Wiley and Sons.(1997)
[4] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In Proceedings Workshop on the Algorithmic Foundations of Robotics (WAFR'00), (2000).
[5] J. J. Kuffner Jr. and S. M. LaValle. RRT-connect: An efficient approach to singlequery path planning. In Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2000), pages 995–1001, (2000).
[6] P. Cheng. Reducing RRT metric sensitivity for motion planning with differential constraints. Master's thesis, Computing Science, Iowa State University, (2002).
[7] B. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003), pages 317–323, June (2003).
[8] N. Roy and S. Thrun. Coastal navigation with mobile robots. Advances in Neural Processing Systems, 12:1043–1049, (2000).
[9] M. Kalisiak and M. V. de Panne. RRT-blossom: RRT with a local flood-fill behavior. In Proceedings of IEEE International Conference on Robotics and Automation (ICRA06), pages 1237–1242, April (2006).

New Trends in Mechatronics and Materials Engineering
10.4028/www.scientific.net/AMM.151

Extending RRT for Robot Motion Planning with SLAM
10.4028/www.scientific.net/AMM.151.493