Robotics and Autonomous Systems 68 (2015) 1–11

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments

Ahmed Hussain Qureshi *, Yasar Ayaz

Robotics And Intelligent Systems Engineering (RISE) Lab, Department of Robotics and Artificial Intelligence, School of Mechanical And Manufacturing Engineering (SMME), National University of Sciences And Technology (NUST), H-12 Campus, Islamabad, 44000, Pakistan

HIGHLIGHTS

- Intelligent sample insertion.
- Non-greedy trees connection.
- Improves the convergence rate.

ARTICLE INFO

Article history:
Received 8 September 2014
Received in revised form
6 January 2015
Accepted 16 February 2015
Available online 23 February 2015

Keywords:
Motion planning
Sampling-based algorithms
RRT
RRT*
Optimal path planning
Bidirectional trees

ABSTRACT

The sampling-based motion planning algorithm known as Rapidly-exploring Random Trees (RRT) has gained the attention of many researchers due to their computational efficiency and effectiveness. Recently, a variant of RRT called RRT* has been proposed that ensures asymptotic optimality. Subsequently its bidirectional version has also been introduced in the literature known as Bidirectional-RRT* (B-RRT*). We introduce a new variant called Intelligent Bidirectional-RRT* (IB-RRT*) which is an improved variant of the optimal RRT* and bidirectional version of RRT* (B-RRT*) algorithms and is specially designed for complex cluttered environments. IB-RRT* utilizes the bidirectional trees approach and introduces intelligent sample insertion heuristic for fast convergence to the optimal path solution using uniform sampling heuristics. The proposed algorithm is evaluated theoretically and experimental results are presented that compares IB-RRT* with RRT* and B-RRT*. Moreover, experimental results demonstrate the superior efficiency of IB-RRT* in comparison with RRT* and B-RRT in complex cluttered environments.

1. Introduction

Motion planning is a well-known problem in robotics [1]. It can be defined as the process of finding a collision-free path for a robot from its initial to goal point while avoiding collisions with any static obstacles or other agents present in its environment. Although motion planning is not the only fundamental problem of robotics, perhaps it has gained popularity among researchers due to widespread applications such as in robotics [2], assembly maintenance [3], computer animation [4], computer-aided surgery [5], manufacturing [6], and many other aspects of daily life.

The journey of finding solution to motion planning problems started with complete planning algorithms that comprised of deterministic path planning approach. Complete motion planning algorithms [7,8] are those algorithms that converges to a path solution, if one exists, in finite time. These algorithms are proven to be computationally inefficient [9] in most of the practical motion planning problems [10]. Resolution complete algorithms were then introduced that require fine tuning of resolution parameters for providing the motion planning solution, if one exists, in a finite time period. Artificial Potential Fields (APF) [11] is a well-known resolution complete algorithm. However, APF suffers from the problem of local minima [12] and does not perform well in the environment with narrow passages. Hence, the search for an efficient solution to the problem continued and the idea of exact roadmaps was introduced in the literature which relies on the discretization of the given search space. This discretization of search space makes the algorithm computationally expensive for higher dimensional spaces, that is why the application of such algorithms like Cell Decomposition methods [13,14], Delaunay Triangulations [15] and Dynamic Graph Search methods [16,17] are limited to low dimensional spaces only [18]. Moreover the algorithms that combine the set of allowed motions with the graph search methods thus generating state lattices, such as in [19–21], also suffered from the undesirable effects of discretization.

* Corresponding author.
E-mail address: ahqureshi@smme.nust.edu.pk (A.H. Qureshi).

http://dx.doi.org/10.1016/j.robot.2015.02.007
0921-8890/

Hence to solve the higher dimensional planning problems, the sampling-based algorithms were introduced [18]; the main advantage of sampling-based algorithms as compared to other state-of-the-art algorithms is avoidance of explicit construction of obstacle configuration space. These algorithms ensure *probabilistic completeness* which implies that as the number of iterations increases to infinity, the probability of finding a solution, if one exists, approaches one. The sampling-based algorithms have proven to be computationally efficient [9] solution to motion planning problems. Arguably, the most well-known sampling-based algorithms include Probabilistic Road Maps (PRM) [22,23] and Rapidly exploring Random Trees (RRT) [24]. However, PRMs tend to be inefficient when obstacle geometry is not known beforehand [10]. Therefore, in order to derive efficient solutions for motion planning in the practical world, the Rapidly-exploring Random Trees (RRT) algorithms [24] have been extensively explored. Various algorithms enhancing original RRT algorithm have been proposed [25–27,10]. These algorithms present a solution regardless of whether specific geometry of the obstacles is known beforehand or not. One of the most remarkable variant of RRT algorithm is RRT*, an algorithm which guarantees eventual convergence to an optimal path solution [10], unlike the original RRT algorithm. Just like the RRT algorithm, RRT* is able to generate an initial path towards the goal very quickly. It then continues to refine this initial path in successive iterations, eventually returning an optimal or near optimal path towards the goal as the number of iterations approach infinity [28]. This additional guarantee of optimality makes the RRT* algorithm very useful for real -time applications [29]. However, some major constraints still exist in this RRT variant which are presented in this paper. For example:

(i) its slow convergence rate in achieving the optimal solution;
(ii) its significantly large memory requirements due to the large number of iterations utilized to calculate the optimal path; and
(iii) its rejection of samples which may not be directly connectable with the existing nodes in the tree, but may lie closer to the goal region and hence could aid the algorithm in determining an optimal path much faster.

Various heuristics have been introduced, such as [30–33], which perform guided search of the given space instead of pure uniform search (as by RRT and RRT*). Although these biased sampling heuristics make the original RRT* algorithm fast but there is a drawback of computational overload caused by biased sampling. This computational overload limits their application to a limited number of fields [34]. Moreover another disadvantage of deterministic sampling heuristics is that they may interfere with the algorithm characteristics. For example assume a simple case of using goal-biased sampling [31] with bidirectional RRT that alternatively grows two trees. The use of this biased sampling will cause the two trees to always remain in one half of the search space, which is quite undesirable. Hence, to cover the whole search space, a separate sample generator is required for both trees which will cost a significant computational load. Hence there is a need of some better approach that enhances the convergence rate of RRT* for achieving the optimal path solution without affecting the randomization of its sampling heuristic. More recent proposition is the bidirectional version of RRT* known as B-RRT* [35]. B-RRT* presented in [35] is a simple bidirectional implementation of RRT*. B-RRT* uses a slight variation of greedy RRT-Connect heuristic [27] for the connection of two trees. Two directional trees employing greedy connect heuristic for the connection of trees does not ensure *asymptotic optimality* [35]. The B-RRT* uses slight variation of greedy heuristic i.e., the tree under process first searches for the neighbor vertices before making an attempt to connect the trees using RRT-Connect heuristic [27]. This hybrid greedy connection heuristic of B-RRT* slows down its ability to converge to the optimal solution and also makes it computationally expensive. More detailed discussion is provided in the analysis section. This paper introduces a bidirectional variation to the RRT* algorithm, with unique sample insertion and tree connection heuristics that allows fast convergence to the optimal path solution. The proposed Intelligent Bidirectional-RRT* (IB-RRT*) algorithm has been tested for its robustness in both 2-D and 3-D environments and has also been compared with other state-of-art algorithms such as Bidirectional-RRT* [35] and RRT* itself [28]. The rest of the paper is organized as follows. Section 2 addresses the problem definition, Section 3 explains the RRT* algorithm while Section 4 describes the B-RRT* motion planning algorithm in detail. Section 5 presents the proposed Intelligent Bidirectional-RRT* (IB-RRT*). Section 6 presents analysis of the three algorithms under investigation in terms of probabilistic completeness, asymptotic optimality, convergence to the optimal solution and computational complexity. Section 7 provides experimental evidence in support of theoretical results presented in the previous section, whereas Section 8 concludes the paper, also suggesting some future areas of research in this particular domain.

2. Problem definition

Let the given state space be denoted by a set $X \subset \mathbb{R}^n$, where n represents the dimension of the given space i.e., $n \in \mathbb{N}, n \geq 2$. The configuration space is further classified into obstacle and obstacle-free regions denoted by $X_{\text{obs}} \subset X$ and $X_{\text{free}} = X \setminus X_{\text{obs}}$, respectively. $X_{\text{goal}} \subset X_{\text{free}}$ is the goal region. Let $T_{\text{a}} = (V_{\text{a}}, E_{\text{a}}) \subset X_{\text{free}}$ and $T_{\text{b}} = (V_{\text{b}}, E_{\text{b}}) \subset X_{\text{free}}$ represent two growing random trees, where V denotes the nodes and E denotes the edges connecting these nodes. $x^{\text{a}}_{\text{init}} \in X_{\text{free}}$ and $x^{\text{b}}_{\text{init}} \in X_{\text{goal}}$ represent the starting states for T_{a} and T_{b}. The function $\mu()$ computes the Lebesgue measure of any given state space e.g. $\mu(X)$ denotes the Lebesgue measure of the whole state space X. It is also called the n-dimensional volume of any given configuration. This paper only considers Euclidean space and positive Euclidean distance between any two states e.g., $x_1 \in X$ and $x_2 \in X$ is denoted by $\text{d}(x_1, x_2)$. The closed ball region of radius $r \in \mathbb{R}, r > 0$ centered at x is denoted as $\mathfrak{B}_{x,r} := \{x_2 \in X : d(x, x_2) \leq r\}$, where $x \in X$ can be any given configuration state. Let the path connecting any two states $x_1 \in X_{\text{free}}$ and $x_2 \in X_{\text{free}}$ be denoted by $\sigma : [0, s']$, such that $\sigma(0) = x_1$ and $\sigma(s') = x_2$, whereas s' is the positive scalar length of the path. The set of all collision-free paths σ is denoted as $\sum_{\text{free}}$. Given any random state $x \in X_{\text{free}}$, the path function connecting initial state x_{init} and random state x is denoted as $\sigma'_{\text{a}}[0, s_{\text{a}}] \subset X_{\text{free}} | \{\sigma'_{\text{a}}(0) = x_{\text{init}}$ and $\sigma'_{\text{a}}(s_{\text{a}}) = x\}$, while the path function connecting random state x and goal region X_{goal} is denoted as $\sigma'_{\text{b}}[0, s_{\text{b}}] \subset X_{\text{free}} | \{\sigma'_{\text{b}}(0) = x$ and $\sigma'_{\text{b}}(s_{\text{b}}) \in X_{\text{goal}}\}$. The complete, end-to-end path function i.e., the path function from root to the goal is denoted by $\sigma'_{\text{f}}(s) = \sigma'_{\text{a}} | \sigma'_{\text{b}} : [0, s] \in X$, where s represents the scalar length of the end-to-end path. The expression $\sigma'_{\text{a}} | \sigma'_{\text{b}} \in X$ describes the concatenation of the two path functions, σ'_{a} and σ'_{b}. The path function σ_{f} is the end-to-end feasible path in obstacle-free configuration space, i.e., $\sigma_{\text{f}} \in X_{\text{free}}$. The set of all end-to-end collision-free paths is denoted as $\sum_{\text{f}}$ i.e., $\sigma_{\text{f}} \in \sum_{\text{f}}$. The cost function $c(\cdot)$ computes the cost in terms of Euclidean distance.

The following motion planning problems will be considered in the proposed algorithm:

Problem 1 (*Feasible Path Solution*)**.** Find a path $\sigma_{\text{f}} : [0, s]$, if one exists, in obstacle-free space $X_{\text{free}} \subset X$ such that $\sigma_{\text{f}}(0) = x_{\text{init}} \in X_{\text{free}}$ and $\sigma_{\text{f}}(s) \in X_{\text{goal}}$. If no such path exists, report failure.

Problem 2 (*Optimal Path Solution*)**.** Find an optimal path $\sigma^*_{\text{f}} : [0, s]$ connecting x_{init} and X_{goal} in obstacle-free space $X_{\text{free}} \subset X$, such that the cost of the path σ^*_{f} is minimum, i.e., $c(\sigma^*_{\text{f}}) = \{\min_{\sigma_{\text{s}}} c(\sigma_{\text{f}}) : \sigma_{\text{f}} \in \sum_{\text{f}}\}$.