Robotics and Autonomous Systems 86 (2016) 13–28

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Heuristic approaches in robot path planning: A survey

Thi Thoa Mac [a,b,*], Cosmin Copot [a], Duc Trung Tran [b], Robin De Keyser [a]

[a] Department of Electrical Energy, Systems and Automation, Ghent University, Technologiepark 914, 9052 Zwijnaarde, Belgium
[b] School of Mechanical Engineering, Hanoi University of Science and Technology, Dai Co Viet street 1, Hanoi, Viet Nam

HIGHLIGHTS

- Autonomous navigation of a robot is a promising research domain due to its extensive applications.
- This survey concentrates on heuristic-based algorithms in robot path planning which are comprised of neural network, fuzzy logic, nature inspired algorithms and hybrid algorithms.
- The strengths and drawbacks of each algorithm are discussed and future outline is provided.

ARTICLE INFO

Article history:
Received 17 August 2015
Accepted 5 August 2016
Available online 26 August 2016

Keywords:
Autonomous navigation
Robot path planning
Heuristic methods
Neural network
Fuzzy logic
Nature— inspired algorithms
Potential field method

ABSTRACT

Autonomous navigation of a robot is a promising research domain due to its extensive applications. The navigation consists of four essential requirements known as perception, localization, cognition and path planning, and motion control in which path planning is the most important and interesting part. The proposed path planning techniques are classified into two main categories: classical methods and heuristic methods. The classical methods consist of cell decomposition, potential field method, subgoal network and road map. The approaches are simple; however, they commonly consume expensive computation and may possibly fail when the robot confronts with uncertainty. This survey concentrates on heuristic-based algorithms in robot path planning which are comprised of neural network, fuzzy logic, nature-inspired algorithms and hybrid algorithms. In addition, potential field method is also considered due to the good results. The strengths and drawbacks of each algorithm are discussed and future outline is provided.

1. Introduction

Autonomous navigation is one of the most important requirements of an intelligent vehicle. Robot navigation is a designed process toward a target position while avoiding obstacles. There are four basic components of this process as shown in Fig. 1 [1]: (i) *perception*, the robot uses its sensors to extract meaningful information; (ii) *localization*, the robot determines its location in the working space; (iii) *cognition and path planning*, the robot decides how to steer to achieve its goal; (iv) *motion control*, the robot regulates its motion to accomplish the desired trajectory. Path planning of a robot can be considered as a sequence of translation and rotation from starting position to the destination while avoiding obstacles in its working environment. There are two suggested techniques covering all approaches in robot path planning: (i) *global path planning* or off-line path planning and (ii) *local path planning* or on-line path planning [2,3]. A global path planner usually generates a low-resolution high-level path based on a known environmental map or its current and past perceptive information of the environment. The method is valuable of producing an optimized path; however, it is inadequate reacting to unknown or dynamic obstacles. On the other hand, local path planning algorithm does not need *a priori* information of the environment. It usually gives a high-resolution low-level path only over a fragment of global path based on information from on-board sensors. It works effectively in dynamic environments. The method is inefficient when the target is long distance away or the environment is cluttered. Normally, the combination of both methods is advised to enhance their advantages and eliminates some of their weaknesses [4–6]. The robot path planning problem can be divided into classical methods and heuristic methods [7,8] as shown in Fig. 2.

The most important classical methods consist of cell decomposition method (CD), potential field method (PFM), subgoal method

* Corresponding author at: Thi Thoa Mac, Department of Electrical Energy, Systems and Automation, Ghent University, Technologiepark 914, 9052 Zwijnaarde, Belgium.
E-mail address: thoa.macthi@ugent.be (T.T. Mac).

http://dx.doi.org/10.1016/j.robot.2016.08.001
0921-8890/

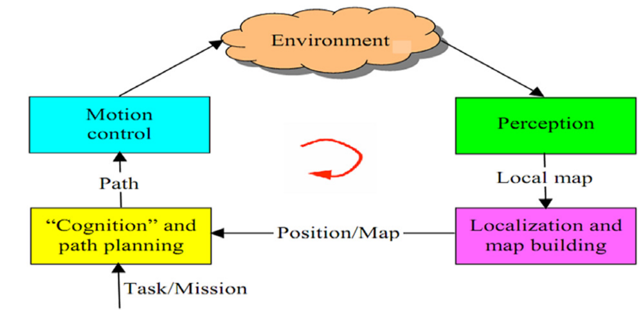

Fig. 1. Robot navigation structure [1].

(SG) and sampling-based methods. In Cell Decomposition method, the free space of the robot's configuration is divided into small regions called cells. The goal is to provide a collision-free path to reach the target. The applications of robot path planning based on this approach can be found in [9,10]. In potential field method, the obstacles and the goal are assigned repulsive and attractive forces, respectively, so that the robot is able to move toward the target while pushing away from obstacles [11]. A new formula of repelling potential is performed in the interest of reducing oscillations and avoiding conflicts when obstacles locate near a target [12]. Instead of single robot, the method is also extended successfully to navigate multi robots to perform complex tasks [13,14]. D.H. Kim proposed a new framework to escape from a local minimum location of robot path based on PFM [15,16]. To solve path planning problem in dynamics environments, modifications on classical PFM are introduced in [17,18]. Subgoal method uses a list of reachable configurations from the starting position to the goal position while avoiding all obstacles. The subgoal method applications for robot navigation are presented in [19–21].

Planning schemes based on sampling-based motion planning (SBP) algorithms have received considerable attention due to their capability in complex and/or time critical real world planning problems. Arguably, the most persuasive SBPs to date include probabilistic road-map (PRM) and rapidly-exploring random trees (RRT) [22]. Although the idea of connecting points sampled randomly is fundamental in both approaches, these two methods are different in the manner that they construct a graph connecting the points [23]. A comprehensive survey of work in SBP is given in [24]. The PRM algorithm has been recorded to implement well in high-dimensional state spaces. The PRM is created by curves or straight lines that enable the robot to go anywhere in its free space. The two well-known road-map methods, namely, visibility graph (VG) and Voronoi diagram (VD) have achieved very good results with dramatically different types of roads. A visibility graph is a graph that comes as close as possible to obstacles. As a result, the shortest path is found by applying this method; however, the path touches obstacles at the vertices or edges and thus is dangerous for the robot. Contrary, Voronoi diagram creates a road that tends to maximize the distance between the robot and the obstacles. Therefore, the solution paths based on Voronoi diagram are not optimal with respect to path length. The advantage of this method is that only a limited number of sensors is used in the robot navigation task. Path planning of a robot swarm using road-map technique is proposed in [25–27]. Several improvements are proposed by [28,29]. RRT has received a considerable amount of attention, because of its computational efficiency and effectiveness and its ability to find a feasible motion plan relatively quickly, even in high-dimensional space [30,31]. In [32,33], the navigation approach consists of four separate modules: localization, path planning, path execution and obstacle avoidance; obstacle avoidance is proposed for autonomous urban service mobile robots. To avoid obstacles, the authors combine a local planner with a slightly modified dynamic window method. The local planner is implemented using RRT. RRT explores a robot working space by incrementally building a tree, creating new branches by generating points randomly and linking them to the closest point for which an obstacle-free path is obtained. A problem in RRT is that it produces a path with many branches in the workspace by using the randomized technique. To overcome this problem, a novel path planning approach for a mobile robot in dynamic and cluttered environments with kinodynamic constraints is presented in [34]. The algorithm called Heuristic Arrival Time Field biased Random Tree (HeAT-RT) that takes advantage of the high-exploration ability of a randomized tree is combined with an arrival time field and heuristics to achieve the path optimality, safety, and applicability to the real robot. Instead of choosing a random point from the entire workspace like the basic RRT algorithm, they select a random point using the bias from the arrival time field so that the tree grows in a favorable direction toward the target. The kinodynamic RRT*, an incremental sampling-based approach for asymptotically optimal motion planning for robots with linear dynamics is introduced in [35].

The ability of SBP to provide valid paths for constrained high dimensional problems is advantageous. Despite the hit-or-miss sampling approach being the core of the SBP's effective strategy, it leads to the inclusion of many redundant maneuvers in the obtained path. In [36], a modification of the termination condition is proposed in a way such that the SBP keeps running to iteratively converge the path cost. The solution convergence remains an unanswered problem; until it is proven that given infinity runtime RRT will not achieve an optimal path [37]. Recently, a family of optimal SBP, RRT*, PRM* and RRG* are introduced to guarantee asymptotic optimality. Despite their effectiveness, they provide no theoretical guarantees for reaching an optimal solution [24].

Many efforts have been made to apply classical approaches onto real-time motion planning [38–42]. Incremental algorithms to update distance maps, Voronoi diagrams, and configuration-space collision maps are presented in [38]. The representations are initialized by using a given grid map or point cloud. For efficient on-line applications, only update cells that are affected by changes in the environment are updated. Therefore, these algorithms can be used in real-world scenarios with unexpected or moving obstacles. Another practical approach to solve the limitations of the road-map based mobile robot path planner in a home environment is introduced in [39]. The proposed

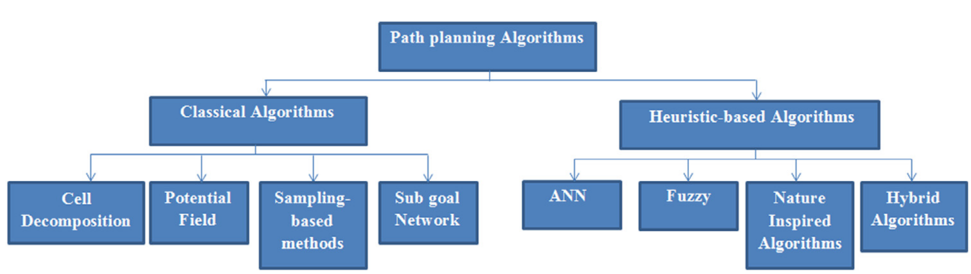

Fig. 2. The classification of robot path planning algorithms.