Robotics and Autonomous Systems 58 (2010) 425–434

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

A local obstacle avoidance method for mobile robots in partially known environment

Chaoxia Shi [a,*], Yanqing Wang [b], Jingyu Yang [a]

[a] School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, 210094, China

[b] School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, 150080, China

ARTICLE INFO

Article history:
Received 22 December 2005
Received in revised form
20 February 2010
Accepted 23 February 2010
Available online 26 February 2010

Keywords:
Obstacle avoidance
Autonomous navigation
Mobile robot

ABSTRACT

Local obstacle avoidance is a principle capability for mobile robots in unknown or partially known environment. A series of velocity space methods including the curvature velocity method (CVM), the lane curvature method (LCM) and the beam curvature method (BCM) formulate the local obstacle avoidance problem as one of constrained optimization in the velocity space by taking the physical constraints of the environment and the dynamics of the vehicle into account. We present a new local obstacle avoidance approach that combines the prediction model of collision with the improved BCM. Not only does this method inherit the quickness of BCM and the safety of LCM, but also the proposed prediction based BCM (PBCM) can be used to avoid moving obstacles in dynamic environments.

1. Introduction

Local obstacle avoidance is a fundamental problem in autonomous navigation for mobile robot. Most of the navigation approaches in partially known environment combines a global navigation method to find a feasible free path leading to the goal and a local navigation method to follow the path avoiding obstacles. Several well-known local obstacle avoidance methods work by a direction for the robot to head in, in Cartesian space or configuration space [1,2]. For example, the Artificial Potential Fields methods (APF) [3,4] were first proposed by Khatib [5]. The main idea is to generate attraction and repulsion forces, within the working environment of the robot, to guide it to the goal. The goal point has an attractive influence on the robot and each obstacle tends to push away the robot. Potential field methods provide an elegant solution to the path-finding problem. Since the path is the result of the interaction of appropriate force fields, the path-finding problem becomes a search for optimum field configurations instead of the direct construction of an optimum path. APF uses vector sums of repulsive and attractive virtual force to compute a desired robot heading. The velocity of robot is proportional to the magnitude of the potential vector. The Vector Field Histogram method (VFH) [6,7] uses a two-dimensional Cartesian histogram grid as a world model, which is updated continuously with range data sampled by on-board range sensors. The VFH method subsequently employs a two-stage data-reduction process in order to compute the desired control commands for the vehicle. In the first stage the histogram grid is reduced to a one-dimensional polar histogram that is constructed around the robot's momentary location. Each sector in the polar histogram contains a value represents the polar obstacle density in that direction. In the second stage, the algorithm selects the most suitable sector from among all polar histogram sectors with a low polar obstacle density, and the steering of the robot is aligned with that direction. Robot's velocity chosen after the direction has been selected is proportional to the distance obstacle ahead. While this method produces smoother travel and can handle both narrow and wide openings, it does not account for the fact that when the robot turns they typically move along arcs, rather than in straight lines. Further more, it is still not adequate to deal with vehicle dynamics, which can cause problems in cluttered environments.

Many approaches have been addressed to deal with local obstacle avoidance problem in dynamic environment. Those works were classified into two categories in terms of the knowledge of the movement of obstacles [8]. In the first category, movements of obstacles are completely unknown to the robot [9]. The collision-free near-optimal paths for mobile robots were generated purely depending on the information acquired by the sensor on board. In the second category, movements of obstacles are completely known. In recent studies some methods are presented for optimal motion planning where a start to goal trajectory is computed at discrete time intervals by searching a tree of feasible avoidance maneuvers [10]. Literature [11] investigated behavior of a single mobile robot which is navigated by an "iterated forecast and planning" scheme in an environment where multiple obstacles are

* Corresponding author.
E-mail address: stonexia@sohu.com (C. Shi).

0921-8890/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2010.02.005

moving around. This navigation scheme searches a feasible path for a robot in (x, y, t) space by a heuristic method. The movement of each obstacle is then forecasted under the assumption that it moves with a piecewise constant velocity.

Velocity space approaches choose angular velocity along with linear velocity, and can take vehicle dynamics into account. The Curvature Velocity Method [12] formulates the local obstacle avoidance problem as one of constrained optimization in the velocity space of the robot. Constraints that result from the robot's physical limitations and the environment are placed on the velocity space of the robot. The robot chooses velocity commands that satisfy all the constraints and maximize an objective function that trade off speed, safety and goal-directedness. The method has been proved to be efficient and real-time in partially known environments indoors. However, CVM converts Cartesian space to configuration space through expanding the radii of the obstacles by the radius of the robot. Subsequently, it calculates the tangent curvature by the geometric center of the obstacle. Obviously, it is not easier to achieve space conversion than calculate the tangent curvature by the obstacle's center without prior knowledge of obstacles in partially known or unknown environments. Though CVM produces reliable, smooth and speedy navigation in office environments, it often fails to guide the robot into an open corridor towards the goal direction and it sometimes lets a robot head towards an obstacle until the robot gets near the obstacle. These problems derived from the fact that CVM chooses commands based on the collision-free length of the arcs assumed to be robot's trajectories. It is easy to ignore the case that the most appropriate path perhaps exists in short distance.

Ko and Simmons [13] realized these limitations and developed the lane curvature method This method combines CVM with a new directional method called the lane method, which divides the environment into lanes, and then chooses the best lane to follow to optimize travel along a desired heading. A local heading is then calculated for entering and following the best lane, and CVM uses this heading to determine the optimal linear and angular velocities. The lane method chooses a direction to a wide and collision-free opening since it decides heading direction based on the collision-free distance and the width of the lane. While VFH method chooses a direction to the opening with wide collision-free angular range rather than an opening with big width. So VFH often forces a robot into a narrow opening near the robot because even a narrow opening can offer wide collision-free angular range if the opening is close to the robot. In this respect, the lane method can provide safer heading commands to CVM than VFH. Nevertheless, because the lane calculation modifies the sensor vision of the environment, the robot may not see clear space with enough distance free of collision caused by the lack of radial projected vision. In practical experiments, the sharp turns that caused by delays in finding clear spaces have been observed.

Fernandez and Sanz [14] combine a new directional method, called beam method (BM), to improve the performance of CVM as well as LCM. The beam method calculates the best heading that will be delivered to CVM to obtain the optimal linear and angular velocities. The resulting combined technique is called the beam curvature method (BCM). BM obtains a divergent radial projection model of the environment based on the sensors' common position. The projection model is defined by a set of sectors or radial beams. This model is simplified and then a set of possible candidate beams is determined. After that, the best beam is calculated by maximizing an objective function. The local heading target is calculated by imposing security constraints around the best beam. Experiments in different scenarios indicate that BCM can not only find the opening faster, but also produce smoother trajectory than CVM and LCM. However, BCM assumes that there are no openings behind the obstacles, which may leads to the false selection of optimal opening when appropriate path lies in between the two obstacles especially in clustered environment. Furthermore, BCM performs well in clustered environment with static obstacles, but may fail to avoid a moving obstacle in dynamic environments. It is because BCM itself doesn't take the corresponding velocity of the obstacles in robot's local reference frame into account. Therefore, we used a prediction model to help BCM realize local obstacle avoidance in dynamic environment.

The remainder of this paper is organized as follows. A method that converts Cartesian space to configuration space is introduced in Section 2. The improved BCM has been given in Section 3. Section 4 describes the prediction model for BCM in detail. Some experimental results of autonomous navigation are summarized in Section 5. Finally, conclusions are given in Section 6.

2. Conversion from Cartesian space to configuration space

Although various methods about obstacle avoidance have been thoroughly considered in the literature, it is evident that not all the developed methods are able to give an appropriate answer to all possible situations. In most partially known environment, the global pose of the robot in world reference frame and the detailed shape of obstacles are unknown, the robot has to detect and localize the obstacles by sensors on board. But the navigation system can supply the robot with the angle error between its current heading direction and the desired heading direction. For example, a mobile vehicle in outer space to explore the unknown planet can get its position and the target's position in world reference frame by the Global Position System. However, the detailed information of obstacles located on the way to the target is unknown. Another good case in point is that a mobile robot can achieve the task according to the map constructed by global vision systems located on the floor in office environments. Nevertheless, the robot may fail to complete the task without the capability of handling the emergency such as to avoid new added chairs, desks or unexpected moving human beings. So, in most cases, the geometry centers of obstacles in partially known or unknown environment are a puzzle to the robot even though it has seen them. Therefore, the fact that both CVM calculates tangent curvature and BCM forms the beam in terms of the coordinates of obstacle center limits their practical application when the prior knowledge of the obstacle is unknown. Expanding each sensing point by the radius of the robot is a feasible resolution for space conversion, as is shown in Fig. 1(a), but the robot need to consider every circles centered by the sensing points to calculate the tangent curvatures for each obstacle.

The conversion from Cartesian space to configuration space employs two-stage data-reduction techniques as follows:

(1) Searching for the candidate tangent points of the obstacle. Provided point C and C_1 are the nearest points from the obstacle I and II to the robot, we can denote the two obstacles shown in Fig. 1(a) as two line segments ACB and $A_1C_1B_1$ respectively. Choose point A, T_1 on line segment AO and point B, T_2 on line segment BO as the left and right candidate tangent points under the condition that $|T_1O| = |T_2O| = |CO|$, where point O is the origin point of the robot's local reference frame.

(2) Calculating the left and right tangent curvatures of the obstacle. Denote the candidate tangent points by their polar coordinates. As is shown in Fig. 1(b), we calculate the curvature centers of the left and right tangent arcs with Eqs. (1) by expanding all the left tangent points (A, T_1) and right tangent points (B, T_2) by the radius of the robot, where

O_1: curvature center of the left tangent arc that is tangent with circle T_1 (or A) and passes through the point O;

O_2: curvature center of the right tangent arc that is tangent with circle B (or T_2) and passes through the point O;