Information Sciences 329 (2016) 105–124

Contents lists available at ScienceDirect

Information Sciences

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

Generating automatic road network definition files for unstructured areas using a multiclass support vector machine

Néstor Morales*, Jonay Toledo, Leopoldo Acosta

Departamento de Ingeniería Informática, Universidad de La Laguna, Spain

ARTICLE INFO

Article history:
Received 8 July 2014
Revised 6 August 2015
Accepted 14 September 2015
Available online 25 September 2015

Keywords:
Multiclass support vector machines
Trajectory generation
Road network definition file
Autonomous vehicles
Unstructured areas

ABSTRACT

In this paper, an innovative methodology for the generation of a Road Network Definition File (RNDF) using only an obstacle map as input is presented. This RNDF, which relies on a Multiclass Support Vector Machine(MSVM)-based trajectory generation method, will be used by an autonomous vehicle for transporting people in closed, unstructured areas for which no previous information is available, such as residential areas or industrial parks. The advantages of using this technique are the generation of a safe and smooth trajectory graph (making the trip more comfortable for riders by having trajectories pass as far away as possible from obstacles). Moreover, although there exist other previous Support Vector Machine (SVM) path planning methods, this is the first to use a MSVM. The advantages of doing so are that by obtaining a decision boundary for each object in the scene, all possible trajectories are computed and joined to form a graph. This is done through a combination of a Nearest-Neighbor Graph (NNG) and a Relative Neighborhood Graph (RNG). The method was tested with real data and in real conditions, yielding good results. At the end of the paper, results for two kinds of studies are presented. The first set of tests is intended to determine the best parameter values for the proposed methodology. In the second set of evaluations, the approach is compared with other *state-of-the-art* SVM-based methods, as well as with a classical approach, demonstrating that the method outperforms them in some aspects. Furthermore, the source code of the method is available for testing, as are some videos in which the output of the method is shown, including a comparison with previous methods.

1. Introduction

Research on autonomous vehicles has been a topic of study for quite some time. Proof of this is the extensive related literature, particularly since the start of the DARPA Challenges [3,4]. One of the main topics of interest in the field of Unmanned Autonomous Vehicles (UAV) is related to planning and control. Some examples are the works in [9,46,49].

The method described in this paper is intended to be used in our testing platform, an autonomous robotic prototype called Verdino[1]. This vehicle is a standard golf cart (a TXT-2 from EZ-GO[2]), which has been electronically and mechanically modified so that it can be controlled by an on-board computer. This vehicle, shown in Fig. 1, comes standard with six 6 *V* batteries, a speed controller, a 36 *Vcc* electric motor, mechanical brakes and steering, and has a maximum speed of between 19 and 23 Km/h.

* Corresponding author. Tel.: +34 922 318 287; fax: +34 922 318 288.
E-mail addresses: nestor@isaatc.ull.es (N. Morales), jonay@isaatc.ull.es (J. Toledo), leo@isaatc.ull.es (L. Acosta).

[1] http://verdino.webs.ull.es.

[2] http://www.ezgo.com.

http://dx.doi.org/10.1016/j.ins.2015.09.019
0020-0255/

Fig. 1. Verdino prototype.

For localization, the vehicle is equipped with an odometry system attached to each wheel, which allows for relative position estimates. This information is combined with the information provided by an Inertial Measurement Unit (IMU) (an *Xsens Mti*[3]) consisting of 3 accelerometers, 3 gyroscopes and 3 magnetometers, which is merged in real time in order to yield the three-dimensional orientation of the vehicle; and a centimeter-level precision DGPS (a *JAVAD GNSS Triumph-1*[4]), accurate horizontally to within 1 cm and vertically to around 1.5 cm using Differential GPS (DGPS) at a frequency of 5 Hz. There are also several Light-Detection And Ranging (LIDARs) units, which are used for SLAM. All this information is combined using the method in [34] to ensure the vehicle is properly localized.

These LIDARs units are also used for obstacle detection, which is needed for the method described in this paper. One of the advantages of these sensors is that they are fast and precise. The vehicle is equipped with 5 LIDARs. Two of them are located at the same plane in the front corners of the vehicle, at a height of 50 cm. Each of these covers an angle of 270°. Another one is located in the front of the vehicle, at a height of 20 cm, tilted slightly upward in order to detect small obstacles or non-traversable areas. Another sensor is on top of the vehicle, also at the front, tilted downward to cover the blind spots of the other sensors. Finally, the last LIDAR is situated at the back of the vehicle and is used for maneuvering in reverse. The sensors used are the *SICK LMS 100* and *SICK LMS 111* models[5], allowing a maximum detection distance of 20 m, a precision of 10–35 mm, and a maximum angular resolution of 0.25° at a maximum rate of 50 Hz.

High-level control of the vehicle is accomplished by an on-board computer equipped with an *i7-3770K* processor, 16 Gb of *RAM DDR-3* memory, *SSD* storage and an NVIDIA GeForce GT 640. Low-level control relies on a set of electronic devices developed by the authors that receive commands from the computer and transform them into a signal that is understandable by the actuators described above.

In Verdino, two different planning levels are considered: global planning (on which this paper is focused) and local planning. This latter level uses an approach based on the Frenét space. The idea is that, at each iteration, the orientation information and distance to the global plan are used to generate a set of local tentative paths in the Frenét space. These are transformed again to the Euclidean space and costs are assigned based on various factors, such as the path's safety or curvature. One of these paths is selected and used to compute the steering and speed commands. This local planner, however, is beyond the scope of this paper. For more information, please see the description included in [9].

As concerns the global planner, a fast way to compute paths is needed. It is desirable to have a graph of the existing paths on the road, but the prototype will work in unstructured areas for which only a map obtained with noisy data is available. This map is computed using the method in [19]. Thus, a way to compute a safe and smooth Road Network Definition File (RNDF) that could

[3] http://www.xsens.com.
[4] http://www.javad.com.
[5] http://www.sick.com.