Dynamic Recognition of Obstacles for Optimal Robot Navigation

Miguel Fernandes
http://www.di.ubi.pt
Luís A. Alexandre
http://www.di.ubi.pt/~lfbaa

Dep. Informática
Universidade da Beira Interior
6201-001 Covilhã, Portugal
Instituto de Telecomunicações, Torre Norte, Piso 10
Av. Rovisco Pais, 1
1049-001 Lisboa, Portugal

Abstract

Navigation is a well established field with robust algorithms that can work out-of-the-box in systems like ROS. Nonetheless, there are situations in which the current navigation approaches are laking in terms of optimality. Examples arise when too much "safe space" is assigned around a given object that can completely prevent a robot from using a given path and forces the use of an alternative path that can be much longer. In this paper we propose the dynamic adaptation of robot navigation strategies depending on the type of obstacles that are met during navigation. We do this in real time using a convolutional neural network for obstacle recognition and a path planning parameter adjustment depending on the obstacle category. We present experiments illustrating the difference in paths that can be obtained by using the proposed approach versus standard approaches implemented in ROS.

1 Introduction

Robot navigation is an important and active research area since it is one of the fundamental tasks for a mobile robot. In this paper we propose a method that dynamically adapts the cost mapping parameters to the type of obstacle that is present in the robot's path in order to aid the path planner in choosing the best path to take. The objects are recognized using a Convolutional Neural Network (CNN).

In related work regarding navigation of robotic systems, Xin *et al.* [5] proposes a visual navigation system in order to plan a smoother path for the robot to navigate,while taking into account the dimensions of the robot, and being successful in dealing with a dynamic environment.

Courbon *et al.* [1] improves the robustness of localization and the path-following in visual memory-based navigation frameworks with the concepts of short-term and long-term memories.

Menlingui *et al.* [2] develop a new navigation approach by combining Artificial Potential Fields and Interval Type-2 Fuzzy Logic Systems in a omnidrive mobile robot that presents smooth paths that are fast to calculate.

So, although some work has been done in dynamic adjustment of navigation parameters, it has focused on different goals than the ones we are pursuing in this work. Namely, the above works focused improving the navigation by using smoother paths, improved localization and speed in path planning. We are concerned with allowing the robot to be able to choose paths that could be considered as blocked by obstacles and hence allow for eventually using shorter paths than would otherwise be possible.

2 Proposed Method

2.1 Obstacle recognition

We are interested in determining if the obstacles belong to one of two categories: mobile or static objects.

For this, we start by classifying the obstacles in the scene into the 1000 classes of ImageNet. All the animals, transports, and moving objects are mapped into the mobile category. All the others are placed into the static object category. The classification is done on every frame, thus if an object classified as static and then in a subsequent frame is classified as mobile, the algorithm will adjust to the most recent category. To achieve object recognition in real time we take advantage of a previously trained CNN. We use the Extraction model from [4], which is a CNN trained for the ImageNet dataset. It has top-1 validation accuracy of 72.5%.

2.2 Navigation Adjustment

After obtaining the category of the obstacle, the method changes the parameters of the path planner in order to adapt to the obstacle category. The idea is that some objects are "safer" than others. For instance, objects that can move, such as people or animals, require a larger "safety" distance than static objects, like tables or walls.

Navigation in robotic systems is split in two parts, the Path-Planning and the Cost-mapping. A major component of ROS (Robot Operating System) [3] Navigation Stack is the movebase package. It is composed by two planers, two costmaps and a recovery node. One planner and costmap are local, as a dynamic window around the robot, and the other planer and costmap are related to the global map. The cost maps are filled with information from layers, such as obstacle information as lethal points, or an inflationer that inflates points around lethal points in order for the robot to have a safety distance from obstacles.

Our method acts by:

- first recognizing the object closest to the robot in its planned path;
- mapping this object to one of the two categories: static or mobile;
- adjusting the inflation parameters to allow for the robot to pass closer to static than to mobile obstacles.

3 Experiments

In this section we present two experiments that illustrate the benefits of the proposed approach. The first experiment takes place in a virtual environment (Gazebo simulator) and the second on a real environment. Both use a Turtlebot 2 robot equiped with a Kinect camera. We contrast the application of our method to the use of the standard path planner in ROS.

3.1 Experiment 1

In this experiment, the category of the objects is predetermined (no real-time object categorization). We have created a small maze where the robot is placed in the lower left corner and is instructed to navigate to the upper right corner. There are two paths available for this experiment. The shortest path has two coke cans that serve as a static obstacle that, nonetheless, allows the robot to pass. With the regular cost-mapping static inflation method, the robot takes the longer path due to the fact that the inflation ratios affects the coke cans in a way that the cost of using the longer path is smaller than to pass very close to the obstacle. With our layer, the path-planer makes the robot use the shortest path although it passes very close to the static obstacles.

3.2 Experiment 2

In this experiment we have a setup in our lab that contains two possible paths from point A to B. The first one is shorter but forces the robot to pass very close to an obstacle (under a tripod) – see Fig. 3.

The standard setup for navigation in the ROS stack does not allow the robot to use this path because the safety distance that is used by the planner forces the robot to consider the path as blocked (Fig. 4).

Our method classifies the type of obstacle in the static category and hence assigns it a low probability of motion so allows for a closer approximation of the robot to the obstacle and makes the shorter path usable (Fig. 5). In this experiment, the category of objects is recognized in real-time using a CNN. The obstacle is recognized as a tripod when the robot is about 1 meter away from it. The recognition code runs on the GPU (Titan X) and takes around 10 ms to recognize the object in each frame,

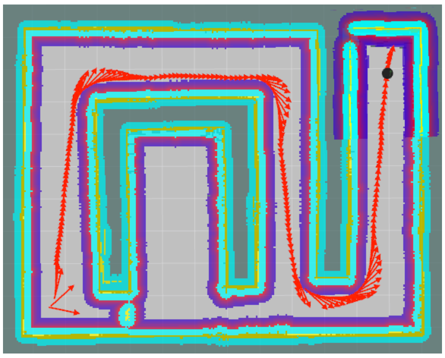

Figure 1: Experiment 1. Map showing the path used by the robot when working with the standard ROS configuration: when faced with the static obstacles in the shortest path, the robot had to choose the longest path.

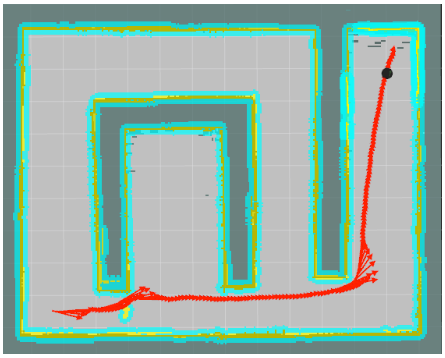

Figure 2: Experiment 1. Map showing the path used by the robot when working with our proposed approach: by recognizing that the obstacle is static it could adjust its parameters to allow for passing closer to the obstacle, making the shorter path available for navigation and saving time.

allowing for real-time recognition. The same code running on CPU takes 5s per frame. We took advantage also of the highly optimized CuDNN library for deep learning.

4 Conclusions

Navigation in unstructured environments is challenging and in this paper we improve the navigation stack of ROS by allowing the recognition of the type of obstacle present in the scene and automatically adjusting the navigation parameters to allow a reduction or increase in the safety space that is considered around obstacles.

We show that this allows a robot to use paths that were previously considered as blocked and can thus improve its navigation performance. We compare in two experiments the results of our method against the standard ROS navigation stack.

Future work will be done to use more object categories and have a finner adjustment of the path planner parameters.

Figure 3: Experiment 2 setup showing the lab, the robot on the starting position and the path with obstacles (two computer boxes and a tripod). The photo was taken standing on the end position.

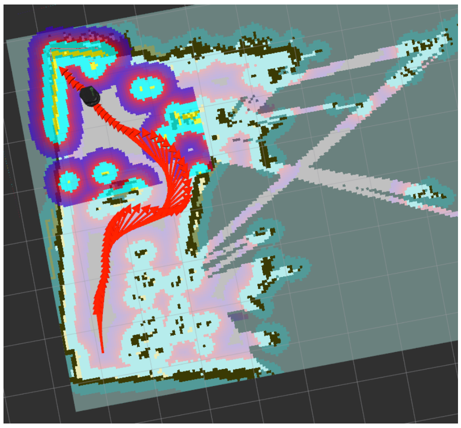

Figure 4: Experiment 2. With the standard ROS approach, the robot has to take the longer path when faced with the tripod obstacle.

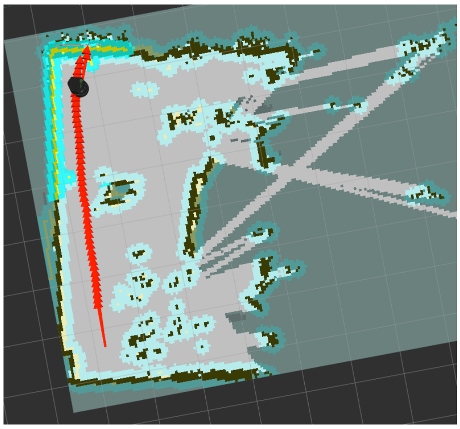

Figure 5: Experiment 2. With our proposal, the robot recognizes the tripod as a static obstacle and adjusts the safety margin around the object, enabling the use of the shortest path.

5 Acknowledgments

This work was supported by National Funding from the FCT - Fundação para a Ciência e a Tecnologia, through the UID/EEA/50008/2013 project.

The GTX Titan X used in this research was donated by the NVIDIA Corporation.

References

[1] J. Courbon, H. Korrapati, and Y. Mezouar. Adaptive visual memory for mobile robot navigation in dynamic environment. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 873–878, June 2012. doi: 10.1109/IVS.2012.6232166.

[2] A. Melingui, T. Chettibi, R. Merzouki, and J. B. Mbede. Adaptive navigation of an omni-drive autonomous mobile robot in unstructured dynamic environments. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1924–1929, Dec 2013. doi: 10.1109/ROBIO.2013.6739750.

[3] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully B. Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[4] Joseph Redmon. Darknet: Open source neural networks in C. `http://pjreddie.com/darknet/`, 2013–2016.

[5] J. Xin, X. L. Jiao, Y. Yang, and D. Liu. Visual navigation for mobile robot with kinect camera in dynamic environment. In *2016 35th Chinese Control Conference (CCC)*, pages 4757–4764, July 2016. doi: 10.1109/ChiCC.2016.7554091.