Paper:

A Universal Autonomous Robot Navigation Method

Annamária R. Várkonyi-Kóczy

Dept. of Measurement and Information Systems, Budapest University of Technology
and Economics Integrated Intelligent Systems Japanese-Hungarian Laboratory
Magyar tudósok körútja 2., H-1117 Budapest, Hungary
[Received September 7, 2007; accepted November 7, 2007]

Recently, autonomous navigation has become an important research topic. There are a lot of applications where the need for autonomous robots is obvious, either because the real or virtual human presence is impossible, dangerous, or expensive, or the tasks to be solved are against the human nature. In most of the applications where robots are to be used, the conditions/environment change along the time that results in an ever-increasing need for universal methods, which are general enough to be used at a wide range of problems. In this paper, a universal, hybrid navigation method is proposed, which is able to work in cases of known, partially known, dynamically changing, or unknown environments. The model consists of two parts which are able to co-operate or to work alone. The modules combine two techniques that deal with a priori information and sensory data separately, thus blends the intelligence and optimality of global navigation methods with the reactivity and low complexity of local ones. The first, global navigation module, based on a priori information, chooses intermediary goals for the local navigation module, for which the so called A* algorithm is used. The second part, carrying out the (local) navigation relying on sensory data, applies a fuzzy-neural representation of an improved potential field based guiding navigation tool. Vision based obstacle detection is implemented by difference detection based on a combination of RGB and HSV representations of the pixels.

Keywords: mobile/indoor robots, vision-based obstacle detection, autonomous/local/grobal navigation, potential field based guiding, A^* algorithm

1. Introduction

Nowadays the topic of autonomous navigation has become popular among researchers all over the world. This is because autonomous navigation can provide a solution to several problems which cannot be solved without an autonomous robot. Most of such applications can be found in fields where the main problem is that human presence is impossible or dangerous in a location where manipulations need to be carried out. For example, the exploration of the surface of Mars was performed by NASA's Sojourner, the autonomous robot carried by Pathfinder (1997), which provided the scientific world with the invaluable piece of information that water had been present on Mars (see details at http://mars.jpl.nasa.gov/MPF/). Without using an autonomous robot, this discovery wouldn't have been made. In Japan, an autonomous robot helicopter has been developed [1], which can help us to put helicopters to various uses where the navigation task is less complex than usual, thus avoid the cost of paying helicopter pilots. Another application presented by Wettergreen et al. [2] is an underwater autonomous robot that collects samples from coral reefs. Human presence would again be much less practical for this task.

Naturally, the use of an autonomous robot can also be practical under less extreme circumstances. There are several simple tasks that could be performed by robots but are today done by humans, for example going round periodically in a building at night and checking whether something is moving or making a noise somewhere; collecting litter in rooms; carrying things from one part of a building to another, etc. Such indoor applications involve a similar navigation task: the environment is well-known and structured (the floor is even and horizontal, walls are vertical and usually perpendicular, rooms are rectangle-shaped and connected by similar doorways), but often dynamic situations arise: humans or other robots appear in the neighbourhood of the robot, which means moving obstacles from the point of view of the navigation.

We can also consider very similar outdoor applications. Just to mention one example, when the task of the robot is to move around scenes/objects and based on the visual information or photos made by the robot to build the 3D model of the scene/object. Accept some non-primary differences (which concern more the setting up and the obstacle detection of the robot and not the navigation) the above situations are very similar. Due to this similarity, we will not consider the specific purpose of the robot, instead we will deal with autonomous (indoor) navigation in general throughout this article.

The navigation system of an autonomous robot is the component which takes care of the movement of the robot, deciding at each moment how to move on. By the word "autonomous," we mean that under normal circumstances, the robot does not communicate with any other

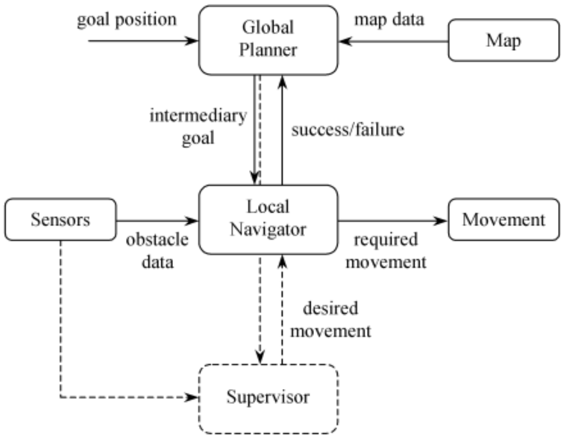

Fig. 1. Block scheme of the hybrid navigation system. Under normal circumstances, the supervisor block (below, dashed line) is not present, but it can be used in a possible teaching phase.

system. Apart from the goal position and the robot's current position and orientation, the navigation system can use two different types of information: a priori, pre-supplied data describing the environment in which the navigation takes place, and sensory data that carry information on the momentary state of the immediate surroundings of the robot, thus also the static and dynamic (moving) obstacles.

To perform the navigation task suitably, the navigation system must be capable to use both types of information mentioned above. However, if we take a look at previous works on autonomous navigation, we find mainly solutions which use principally only one of these. Such navigation algorithms are called global if they take their decisions using mainly a priori information, and local if they rely mostly on actual sensor data. Both types of methods can be appropriate under a variety of different circumstances, and have their specific advantages and disadvantages; however, for the case of navigation indoors, neither a global nor a local method provides a solution that is intelligent enough.

This paper proposes a new method for navigation indoors, which permits to use both a priori information and momentary sensor data in an intelligent way, thus combine the reactive dynamic behaviour and low computational complexity of local methods with the intelligence and optimality of global ones. This is reached by introducing a hybrid navigation system consisting of two modules, one of which uses the a priori information and determines the main steps of the optimal route towards the goal, whereas the other carries out the navigation itself using a local approach (see **Fig. 1**). The specific method used in this part of the system is the potential function based local navigation algorithm, to which ways of reducing complexity, increasing flexibility, and a possibility to teach the system using a fuzzy representation have been suggested. We would also like to emphasise the fact that the introduced method can be used not only in environments that are known and dynamic but in a more general way, since any of the two components can be discarded if special circumstances permit it, i.e. the environment is unknown or we know that nothing changes compared to the pre-stored map.

The paper is organised as follows. Section 2 describes global and local navigation methods, their advantages and disadvantages. In Section 3, the details of the proposed local navigation method are explained, whereas Section 4 deals with the global planning part of the navigation system. Section 5 contains results and comparisons of the simulations and Section 6 gives the conclusion.

2. Navigation Methods

A navigation method is classified as global if it takes its decision using mainly a priori information available on the environment. The use of global algorithms generally leads to a more intelligent navigation behaviour than that of local ones, as global methods have the ability to take all the details of the navigation environment into consideration, if they are known a priori. Some global navigation algorithms - mainly trajectory planning methods that are based on previously existing optimisation algorithms - are able to find an optimal trajectory according to various optimality criteria [3, 4]. The method suggested in [3], for example, follows the two objectives of passing as far as possible from every obstacle, and keeping the trajectory as short as possible in length (how much these objectives take priority over each other can be set using an internal constant). However, global algorithms have their drawbacks; the most important among these is probably their inability to deal properly with dynamic situations. On the appearance of a moving obstacle, the trajectory may have to be modified, i.e. recalculated, which in turn might take significantly more time than the movement of the obstacle itself. As a result, the robot will either slow down or even stop during the time it perceives the moving obstacle, or use a lot more resources than needed. It is also a disadvantage of global methods that to be able to use them, the navigation environment must be known precisely in detail. If an obstacle gets moved into an area that is marked free, it can lead to the same recalculation problem as a moving obstacle. Finally, a robot with a global navigation algorithm can of course not navigate in regions outside its a priori known map.

All these disadvantages are due to the fact that global algorithms calculate the entire trajectory at the beginning of the navigation task, and whenever a new situation arises. This may result in a large quantity of resources being used up needlessly, or an inappropriate navigation behaviour.

Local algorithms take their navigation decisions according to current sensor information, thus they respond well to dynamic challenges, and may be used in an unknown environment as well [5, 6]. A local navigation algorithm is usually safe, it is able to avoid all the obstacles that appear during the navigation, while it continually approaches the goal. As a local algorithm calculates only