Robotics and Autonomous Systems 57 (2009) 1042–1047

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Effect of global position information in unknown world exploration — A case study using the Teleworkbench

Safaa Amin [a,*], Andry Tanoto [a], Ulf Witkowski [a], Ulrich Rückert [a], M. Saied Abdel-Wahab [b]

[a] System and Circuit Technology, Heinz Nixdorf Institute, University of Paderborn, Fuerstenallee 11, D-33102 Paderborn, Germany

[b] Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt

ARTICLE INFO

Article history:
Available online 26 July 2009

Keywords:
Mini-robots
Environment exploration
Robot navigation
Teleworkbench

ABSTRACT

This paper presents empirical results of the effect of the global position information on the performance of the modified local navigation algorithm (MLNA) for unknown world exploration. The results show that global position information enables the algorithm to maintain 100% success rate irrespective of initial robot position, movement speed, and environment complexity. Most mobile robot systems accrue an odometry error while moving, and hence need to use external sensors to recalibrate their position on an ongoing basis. We deal with position calibration to compensate the odometry error using the global position information provided by the Teleworkbench, which is a teleoperated platform and test bed for managing experiments using mini-robots. In this paper we demonstrate how we incorporate the global position information during and after the experiments.

1. Introduction

One of the requirements for successful deployment of mobile robots is the ability to autonomously navigate through the environment. This, in turn, suggests the need for mobile robot systems that can explore an environment and automatically build a map of where the robot can go and where objects are located [1–4]. Most mobile robot systems suffer an odometry error while moving, which makes pure dead reckoning unreliable. Determining the odometry error of a mobile robot is very important, both in order to reduce the error itself and to know the accuracy of the state configuration estimated by using wheel encoder data. The fundamental idea of odometry is the integration of incremental motion information over time, which leads inevitably to the accumulation of errors. In particular, the accumulation of orientation errors will cause large position errors that grow proportionally with the distance travelled by the robot. Nonetheless, most researchers agree that odometry is an important part of a robot navigation system and that navigation tasks can be simplified if odometric accuracy can be improved. The odometry error contains both systematic and non-systematic components. Both components depend on the interaction of the robot with the environment in which the robot moves. In particular, the latter is highly dependent on the environment.

In the case of an ideal robot with an ideal range scanning sensor and no odometry error, Lumelsky [5] was one of the first to develop provably correct exploration strategies, one of which is based on circumnavigating successive objects. This approach, like several others, assumes that there are no dead reckoning errors and that sensors return perfect data. Regarding the systematic errors in differential-drive mobile robots, there are two dominant error sources: unequal wheel diameters and uncertainty of the effective wheel base. In the work by Borenstein and Feng [6], a calibration technique called, "UMBmark" was developed to calibrate the systematic errors of a two-wheeled robot. This method has also been used by other authors in [7]. Goel et al. [8] used another calibration technique to compensate for systematic errors. They referred to the differential-drive mobile robot Pioneer AT. They measured the actual velocities of the wheels and the velocity measurements from the encoders when the robot was placed on a box and the wheels rotated freely in the air. In this way, they found a relationship between the velocity returned by the encoders and the actual velocity measured by using a precise tachometer. Moreover, they measured the effective axle length due to skid steering which differs from that given by the specifications for the robot. Finally, Roy and Thrun [9] suggested an algorithm that uses the robot's sensors to automatically calibrate the robot as it operates. Kelly [10] presented the general solution for linearized systematic error propagation for any trajectory and any error model.

* Corresponding author. Fax: 49 5251606351.
E-mail addresses: safaa.amin@hni.upb.de (S. Amin), tanoto@hni.upb.de (A. Tanoto), witkowski@hni.upb.de (U. Witkowski), rueckert@hni.upb.de (U. Rückert), mswahab@gmail.com (M.S. Abdel-Wahab).

0921-8890/$ – see front matter
doi:10.1016/j.robot.2009.07.019

In this paper we deal with position calibration and odometry error by using the position information provided by the Teleworkbench (TWB) [11]. The TWB uses several webcams to track the robots on the field. From the captured video frames, it extracts the robot positions and orientations relative to the field. As shown in our previous work [4,12], the position and orientation information were used only for visualizing the behaviour of the robots during the experiments. In this paper, the position and orientation information will be used as well for compensating the odometry errors in order to improve the success rate of our modified local navigation algorithm.

The paper is organized as follows. Section 2 presents our algorithm for unknown environment exploration. The method for robot tracking is introduced in Section 3. Section 4 presents a comparison result between the experiment with and without the global position information using the mini-robot Khepera II on the Teleworkbench. Finally, Section 5 concludes the paper.

2. The Modified Local Navigation Algorithm (MLNA)

In order to explore an unknown environment, grid maps can used to represent the environments. The concept of grid maps is to employ a grid of equally spaced cells and to store in each cell its state. The Modified Local Navigation Algorithm (MLNA) described in [12] uses grid maps to enable a robot to explore the environment.

The MLNA works as follows. The robot starts the exploration from any position in the environment. It can move between patches in all directions (east, west, north, south, and diagonal). When the robot visits a certain patch, it is considered to be explored. The next step is computed by calculating the costs of reaching each free patch around the robot. The cost function C for a free patch P is given as

$$C(P) = N(P) \quad (1)$$

where $N(P)$ is the number of free neighbouring patches around patch P. After evaluating the cost for all free patches around the robot, the robot moves to the patch with the lowest cost. A patch is considered to have the lowest cost when it has the lowest number of neighbours and therefore it is most unlikely to be reached again in the future. A cell can be in one of the following states:

- *Unexplored:* The robot has not been in the cell yet. A cell with state 0 is detected by Khepera sensors as an unexplored free cell. It is called a frontier cell.
- *Explored:* The cell has been traversed at least once by the robot, but it might be necessary to go through it again in order to reach unexplored regions.
- *Wall:* The cell cannot be traversed by the robot because it is blocked by an obstacle or a wall.

This algorithm is applied as long as there is at least one free neighbouring cell around the robot. In the case that all the neighbouring cells are explored or occupied by an object, the algorithm will determine a set of frontier cells in the map. A frontier cell represents the boundary between the explored area and the unknown area. Once frontier cells have been detected within a particular evidence grid, the robot attempts to navigate to the nearest, accessible, unvisited frontier cell. The path planner uses a depth-first search on the grid, starting at the robot's current cell. It then attempts to calculate the shortest, obstacle-free path to the target cell. Fig. 1 illustrates the flowchart of the overall exploration process. As in this figure, the algorithm terminates the exploration when there is no unexplored cell in the grid map.

3. Robot global position information

The robot gets its global position information from the robot tracking system. This system works by processing the video frames

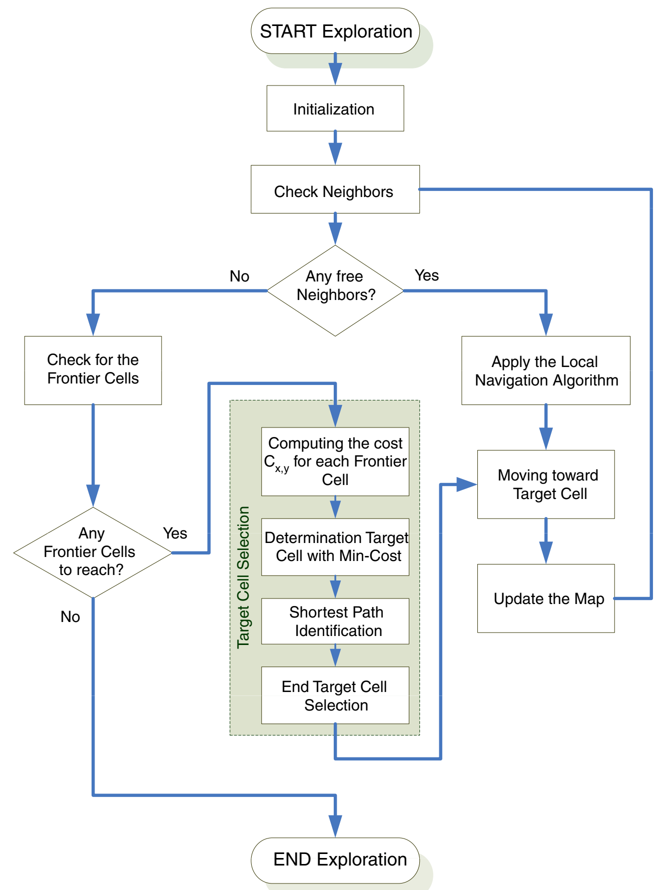

Fig. 1. Exploration process flowchart.

captured by a webcam located above the field. To enable robot identification, a colour mark is placed on top of the robot.

3.1. Teleworkbench

The Teleworkbench is a teleoperated platform and test bed for managing experiments using mini-robots. The system is accessible locally or remotely via the Internet. Through the web-based user interface, local or remote users can set up and execute experiments. Via Bluetooth modules, robots can exchange messages to each other or to the Teleworkbench server wirelessly. During experiments, the video server processes video frames from the webcams to provide the position and orientation (herein after *POS*) of the robots. In parallel, the video data will be stored locally and streamed as live video via the Internet.

For experiment analysis purpose, a graphical analysis tool based on the MPEG-4 video standard has been developed [13]. This tool provides the visualization of the internal information as well as the behaviour of the robots. With this tool, the recorded video of the experiment can be overlaid with some computer-generated objects representing important information, e.g. robot path, sensor data, and exchanged messages. Moreover, users can interactively control the appearance of those objects during runtime.

3.2. Khepera identifier

In order to provide the *POS* of the robot during experiments, each robot is equipped with a specific colour mark (see Fig. 2). Moreover, this colour mark is used as well for robot identification.