Robotics and Autonomous Systems 63 (2015) 206–213

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Route learning and reproduction in a tour-guide robot☆

V. Alvarez-Santos[a,*], A. Canedo-Rodriguez[a], R. Iglesias[a], X.M. Pardo[a], C.V. Regueiro[b], M. Fernandez-Delgado[b]

[a] CITIUS, Centro Singular de Investigacion en Tecnoloxias da Informacion, University of Santiago de Compostela, E-15782, Spain
[b] Department of Electronic and Systems, University of A Coruña, E-15071, Spain

HIGHLIGHTS

- We present our tour-guide robot which is able to learn routes from humans.
- We detail the route recording and reproduction processes of our robot.
- We introduce a novel multi-sensorial algorithm for robot localization.
- We describe several demonstrations that we have carried out with our robot.

ARTICLE INFO

Article history:
Available online 7 August 2014

Keywords:
Tour-guide robot
Route learning
Robot localization
Route teaching

ABSTRACT

Traditionally, route information is introduced in tour-guide robots by experts in robotics. In the tour-guide robot that we are developing, we allow the robot to learn new routes while following an instructor. In this paper we describe the route recording process that takes place while following a human, as well as, how those routes are later reproduced.

A key element of both route recording and reproduction is a robust multi-sensorial localization algorithm that we have designed, which is able to combine various sources of information to obtain an estimate of the robot's pose. In this work we detail how the algorithm works, and how we use it to record routes. Moreover, we describe how our robot reproduces routes, including path planning within route points, and dynamic obstacle avoidance for safe navigation. Finally, we show through several trajectories how the robot was able to learn and reproduce different routes.

1. Introduction

The route information that a tour-guide robot needs is usually introduced by a robotics expert in the laboratory. Contrary to that, we propose a tour-guide robot that allows anyone to teach routes to it, by letting the robot follow them. After a brief route learning stage, our robot will be able to reproduce that route on demand.

Therefore, to record and reproduce a route, we have provided our robot with the ability to estimate its position in the environment where it operates. This is not an easy task because indoor environments contain symmetries, objects that can change its position with time and temporary occlusions. In this sense, humans are a challenge themselves: this kind of robots operate in environments where many humans can significantly change the robot's perception of the environment. Because of this, a robust localization is of key importance for us: if the robot does not record the correct path while it follows the instructor, the instructor will need to repeat the teaching process. For this reason, a key element of both our route recording and reproduction is a robust multi-sensorial localization algorithm that we have designed, which is able to combine various sources of information to obtain an accurate estimate of the robot's pose.

In previous works we have described two important elements of our proposal: how our robot is able to detect and track humans [1], and also how it is able to interact with humans [2]. In this work, we describe the remaining parts: (a) the route recording process that takes place in our robot when it is following a route instructor, and (b) the route reproduction process that takes place

☆ This work was funded by the research projects TIN2009-07737, TIN2012-32262, the Grant BES-2010-040813 FPI-MICINN, and by the Galician Government (Consolidation of Competitive Research Groups, Xunta de Galicia ref. 2010/6).

* Corresponding author. Tel.: +34 618142983.
E-mail addresses: victoralvarezsantos@gmail.com, victor.alvarez@usc.es (V. Alvarez-Santos), adrian.canedo@usc.es (A. Canedo-Rodriguez), roberto.iglesias@usc.es (R. Iglesias), xose.pardo@usc.es (X.M. Pardo), cvazquez@udc.es (C.V. Regueiro), manuel.fernandez.delgado@usc.es (M. Fernandez-Delgado).

http://dx.doi.org/10.1016/j.robot.2014.07.013
0921-8890/

when a user demands a route. More specifically, we describe our multi-sensorial localization algorithm, its advantages over single-sensor location algorithms, and how we use it to record routes. Moreover, we detail how our robot reproduces routes, including path planning within route points, and dynamic obstacle avoidance for safe navigation. Finally, we show through several trajectories how our robot was able to learn and reproduce different routes.

2. Related work

In the late nineties, the first well-known tour-guide robots (*Rhino* [3] and *Minerva* [4]) had no online route learning abilities. In fact, route information was introduced in the robot by an expert. This has been a common element in the tour-guide robots that were developed afterwards. Examples of those include: *RoboX*, [5], which was a tour-guide robot designed for long time operation in a public exposition where the routes were introduced by experts before the exposition, and Robotinho [6], which was one of the first humanoid tour-guide robots. Its routes were also manually introduced in a pre-operational stage of the robot. Even though that there have been improvements in route management, the insertion of route information in the robot has remained as part of a heavy pre-operational stage carried by an expert. The only case that we have found that might introduce certain route learning abilities is a recent tour-guide robot [7] that states that new routes can be created on the fly. Unfortunately no details about the process are given.

Contrary to the route recording problem, mobile robot localization in indoor environments has been deeply studied in the past years. The researchers behind the first tour-guide robots Rhino and Minerva laid the foundations of one of the most popular approaches nowadays: the Monte Carlo Localization (MCL) algorithm [8]. It consisted in estimating the robot's pose by integrating sensorial information in a particle filter. Laser rangefinders are the most widely used sensors for mobile robot localization. Other sensors, such as sonars [9], and cameras [10] have been successfully used as well. In the next years, the popularity of particle filters for robot localization kept growing with improved versions of the original MCL algorithm.

3. Multi-sensor algorithm for mobile robot localization

Generally, localization algorithms use only one sensor. This may be a problem, specially when: (a) the sensor data is highly noisy, (b) the sensor fails to provide data, (c) different areas look alike to the sensor, etc. These problems get worse in crowded environments, because there are people moving around which produce occlusions or changes in the environment like the moving of the furniture. In order to solve these issues, we have proposed [11,12] a localization algorithm that combines the evidence supplied by several sensors.

The solution that we propose is based on the Augmented Monte Carlo Localization Algorithm (AMCL), and unlike other solutions [13] (e.g. Kalman Filters), it can properly handle: (a) sensors with non-Gaussian noise, (b) multi-modal estimations of the pose of the robot, (c) multi-modal sensor models, (d) non-synchronized sensors, and (e) sensors with different data rates that can even stop providing data (e.g. sensor failures).

The goal of our solution is to estimate, at any time t, the robot's pose $\vec{s_t}$ using: (1) perceptual information $\vec{Z_t}$ (or the set of sensor measurements), and (2) control data u_t (the robot movement as provided by odometry encoders). The robot's pose corresponds to $\vec{s_t} = (x_t, y_t, \theta_t)$, where x_t and y_t are the position coordinates, and θ_t is the orientation one.

Essentially, in order to accomplish our goal, we have to previously compute the likelihood $bel(\vec{s_t})$ of every possible pose of our robot, using u_t and $\vec{Z_t}$ (we call this process *pose probability estimation*, Section 3.1). Then, we will perform a *pose estimation* process that is able to obtain the robot's pose $\vec{s_t}$ using these likelihoods (Section 3.2).

3.1. Pose probability estimation

Following a Bayesian filtering approach [14], the likelihood assigned to each robot pose $bel(\vec{s_t})$ will be the posterior probability over the robot state space conditioned on the control data u_t and the sensor measurements $\vec{Z_t}$:

$$bel(\vec{s_t}) = p(\vec{s_t}|\vec{Z_t}, u_t, \vec{Z_{t-1}}, u_t, \ldots, \vec{Z_0}, u_0). \quad (1)$$

Assuming that the current state $\vec{s_t}$ suffices to explain all the previous states, measurements and control data (Markov assumption), we can estimate $bel(\vec{s_t})$ recursively [14]:

$$bel(\vec{s_t}) \propto \left[\int p(\vec{s_t}|\vec{s_{t-1}}, u_t) bel(\vec{s_{t-1}}) d\vec{s_{t-1}}\right] p(Z_t|\vec{s_t}). \quad (2)$$

In this equation, the term $\int p(\vec{s_t}|\vec{s_{t-1}}, u_t) bel(\vec{s_{t-1}}) d\vec{s_{t-1}}$ is in charge of inferring the new $bel(\vec{s_t})$ from $bel(\vec{s_{t-1}})$ and u_t. On the other hand, the term $p(\vec{Z_t}|\vec{s_t})$ corresponds to the update process in charge of sensor fusion, where $\vec{Z_t} = \{z_1, z_2, \ldots, z_{n_t}\}$ is the set of all sensor measurements at time t (n_t is the number of sensors modalities available at time t). Therefore, $p(\vec{Z_t}|\vec{s_t}) = p(z_1, z_2, \ldots, z_{n_t}|\vec{s_t})$ represents the probability that, at time t, the system receives the sensor measurements $\{z_1, z_2, \ldots, z_{n_t}\}$ conditioned on state $\vec{s_t}$. This joint probability function may be very hard to estimate in practice, especially if our sensors provide information at different data rates. For this reason, we assume that the sensor measurements are conditionally independent given the state of the robot; therefore:

$$bel(\vec{s_t}) \propto \left[\int p(\vec{s_t}|\vec{s_{t-1}}, u_t) bel(\vec{s_{t-1}}) d\vec{s_{t-1}}\right] \prod_{k=1}^{n_t} p(z_t^k|\vec{s_t}). \quad (3)$$

In order to be able to apply Eq. (3), we must know: (1) the initial belief distribution $bel(s_0)$ (it can be chosen randomly), (2) the *motion model* of the robot $p(\vec{s_t}|\vec{s_{t-1}}, u_t)$, and (3) the *measurement model* $p(z_t^k|\vec{s_t})$ of each sensor k. The motion model represents the probability of transition from state $\vec{s_{t-1}}$ to state $\vec{s_t}$, provided u_t. This model depends on the odometry of the robot, but it is common to assume that it follows a multivariate normal distribution [14]:

$$p(\vec{s_t}|\vec{s_{t-1}}, u_t) \sim \mathcal{N}(f_{mov}(\vec{s_{t-1}}, u_t), \Sigma_s) \quad (4)$$

where f_{mov} is a function that models the movement of the robot, and Σ_s represents the noise of the model. On the other hand, the measurement model $p(z_t^k|\vec{s_t})$ depends on the nature of each specific sensor. This will be described in detail in Section 4.

3.1.1. Estimation with a particle filter

Eq. (3) can be approximated very efficiently using a particle filter [14]. In this regard, $bel(\vec{s_t})$ can be approximated with a set of M random weighted samples or particles:

$$bel(\vec{s_t}) \approx \vec{P_t} = \{\vec{p_t^1}, \ldots, \vec{p_t^M}\} = \{\{s_t^1, \omega_t^1\}, \ldots, \{s_t^M, \omega_t^M\}\} \quad (5)$$

where each particle $\vec{p_t^i}$ consists, at time t, of a possible robot state s_t^i and a weight ω_t^i (likelihood) assigned to it. We estimate Eq. (3) using the Augmented Monte Carlo Localization algorithm (AMCL) with low variance resampling [14]. Essentially, this algorithm proceeds as follows:

Initially, all particles are distributed randomly over the state space and are assigned equal weights ($\frac{1}{M}$). Then, the algorithm