Engineering Applications of Artificial Intelligence 40 (2015) 1–10

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

# Occupancy grid based graph-SLAM using the distance transform, SURF features and SGD

Arturo Gil *, Miguel Juliá, Óscar Reinoso

*Universidad Miguel Hernández de Elche, Elche, Alicante, Spain*

ARTICLE INFO

*Article history:*
Received 18 March 2014
Received in revised form
23 October 2014
Accepted 17 December 2014
Available online 22 January 2015

*Keywords:*
Occupancy grids
Distance transform
Graph-SLAM
Mobile robot navigation architecture

ABSTRACT

In this paper, we present a SLAM approach that builds global occupancy-grid maps using laser range data. The method consists of two basic algorithms: a process of finding correspondences and alignments between local sub-maps and a high level optimization algorithm that aligns and builds a global map. The main novelty of the paper is the use of a visual description of the local sub-maps. We propose to use visual features to easy the search of correspondences between different sub-maps. The association of features between different maps gives us transformations between the different key maps. Afterwards, a graph is built using the reference frames as the vertexes and the transformation between key-maps are the edges. Stochastic Gradient Descent (SGD) is next employed to compute a global map. The results show the validity of the proposed algorithm in terms of precision and robustness.



## 1. Introduction

Mobile robots possess the capability of moving around the environment while carrying out a task. These machines are becoming more and more frequent in industrial, military and surveillance applications. Thus, research in this area focuses on the achievement of a true autonomous robot capable of performing high-level tasks without supervision. Navigating inside a given scenario usually requires a precise map. In consequence, the problem of Simultaneous Localization and Mapping (SLAM) receives significant attention. Solving the SLAM problem implies the skill of incrementally building the map of the environment while, simultaneously, using this map to compute the robot's absolute location. This is considered as a hard problem, since any error included in the estimation of the location and orientation of the robot induces an error in the estimation of the map, and inversely an error in the map will produce an error when computing the localization of the robot with respect to it. SLAM approaches differ mainly in the kind of sensors used to extract information from the environment, such as laser range finders (e.g. Grisetti et al., 2007a; Hähnel et al., 2003; Biber et al., 2004; Eustice et al., 2005; Triebel and Burgard, 2005). Other researchers have used cameras (Gil et al., 2010b,c), omnidirectional vision sensors (Valiente et al., 2014) and others to obtain usable information to build the map. SLAM algorithms typically differ also in the underlying algorithm used to estimate the map. Classic algorithms are based on the Extended Kalman Filter (EKF) (Dissanayake et al., 2001), a particle Filter approximation (Montemerlo et al., 2002) or Stochastic Gradient Descent (Grisetti et al., 2007b).

A common pitfall in SLAM algorithms is scalability. In general, SLAM algorithms should be capable of computing extensive areas, either using a single or multiple robots. For example, assume that an algorithm is capable of estimating a $1 \times 1$ m occupancy-grid map and expends 1 s in this task. As the area and the quantity of information from the environment grow, the time needed to compute the map generally increases in an exponential manner. Typically, EKF-based approaches to SLAM (Dissanayake et al., 2001) suffer from poor scalability and have a limited applicability to large maps, since their update stage has a quadratic dependence with the number of features in the map. Also, particle filter based SLAM algorithms have problems when used with multiple robots (Gil et al., 2010c), since they depend on sampling the robots' pose. The number of particles required grows as the number of dimensions increases. Currently, researchers are focusing their efforts on Graph-based SLAM algorithms, which present better scalability performance and are a more compact approach to SLAM (Grisetti et al., 2009).

An important concept in any SLAM algorithm is that of loop closing. In general, the mobile robot must be capable of establishing relations between the different areas explored until the moment, for example, whether the area that the robot is traversing now has been visited before or not. Graph-SLAM techniques consider the computation of the map based on a set of restrictions. These restrictions include odometry readings that introduce noisy distances between the nodes of the graph and relations between nodes obtained from observations, e.g. the distance and orientation between node $i$ and node $j$ in the graph. When occupancy-grid maps are used, a typical

* Corresponding author.
*E-mail address:* arturo.gil@umh.es (A. Gil).

http://dx.doi.org/10.1016/j.engappai.2014.12.010
0952-1976/

observation between node $i$ and node $j$ can be derived by aligning the local occupancy maps associated with both nodes. Aligning two local sub-maps is equivalent to finding a translation and rotation between both maps.

Several authors have applied graph-SLAM techniques in combination with computer vision techniques as a way to find correspondences between different observations of the environment. Later, a graph of views of the environment can be created from the correspondences (Konolige et al., 2009; Cummins and Newman, 2008). In these approaches, each node in the graph corresponds to an image acquired by the robot at different poses in space. However, little research has been carried out so far in the usage of graph-based approaches with laser data. In this paper we concentrate on this problem and present several new ideas and techniques. Our approach exploits the concept of the manifold representation (Howard et al., 2006). A manifold consists of a set of patches from a two dimensional space embedded in a higher dimensional space. The use of this representation allows an easy way of retro-traversing the space and it also permits to delay indefinitely the closure of loops while having always a consistent map for navigation.

This paper presents an occupancy grid-based Simultaneous Localization and Mapping (SLAM) algorithm. The approach shows nice scalability properties and can be applied to the mapping of large areas. The main contribution of this paper consists in a novel technique that uses local occupancy sub-maps in combination with visual features to find robust correspondences between different locations in the map. In particular, we propose to use SURF features (Bay et al., 2008) applied to the distance transform (Felzenszwalb and Huttenlocher, 2012) of the local maps. SURF features have been used extensively to extract significant point from planar (Gil et al., 2010a) or panoramic images (Valiente et al., 2014). However, they have never been used on occupancy grid images. In addition, in this study we propose the use of the distance transform prior to the extraction of significant points. We have observed that this image transformation allows us to obtain better results. The distance transform of an image gives to each pixel a value that corresponds with the distance to the closest obstacle in the map. In our representation, obstacles are represented in the occupancy grid as cells that possess a zero value. As a result, after applying the distance transform, we obtain a figure that emphasizes the structure of the environment. As the results will show, SURF features extracted from these distance-transformed maps are very robust. In addition, the number of features is significantly lower than applying the SURF feature detector directly to the occupancy grid, since the distance transform tends to remove unnecessary details and emphasizes the structure of the local areas.

In our scheme we use the manifold representation proposed previously (Howard et al., 2006) but extending and developing it with a different local map representation and a feature based alignment of the sub-maps.

Finally, the SLAM approach is completed by building a map using the graph created from the observations. Each observation is computed as the alignment of two different local occupancy grid maps. Finally, this graph is optimized using stochastic gradient descent (SGD) (Grisetti et al., 2009) and all the occupancy grids are fused together to create a global occupancy grid map.

The rest of the paper is organized as follows. First, Section 2 presents a global view of our proposed architecture. Next, Section 3 defines the different local occupancy-grid maps used in the approach. Following, Section 4 details the graph-SLAM approach, explaining the description, alignment and graph optimization of the local maps, as well as the loop closure and map fusion procedures. Then, Section 5 presents several experiments that were carried out to test our approach. Finally, Section 6 states our conclusions and introduces our future work.

## 2. Architecture

Fig. 1 shows a global layout of the architecture we have implemented using the ROS framework (Quigley et al., 2009). The architecture was designed in order to allow a robot team to explore and create a map of an unknown environment. The different processes that cooperate to solve the problem are indicated with boxes. Note that some processes include high-level navigation tasks that are solved with the map built until that moment. In this paper, we focus on the SLAM part of this architecture consisting of the local maps generation in the *low level planner* as well as the *graph-SLAM* that is the main contribution of this paper. The design of the other modules of this architecture, mainly the *high level planner* and *reactive navigation control*, follows the hybrid exploration model detailed in Juliá et al. (2010). Refer to Juliá et al. (2010) for a full description of these modules.

In Fig. 1 the gray dashed box is used here to indicate that the *simple local mapper*, the *local navigation mapper* and the *reactive navigation control* processes are integrated in one single node for low level planning. These three processes work together at a fast rate since they are implementing a real-time reactive control.

As it can be seen, the *low level planner* takes the scans of the laser and the localization from a scan matcher (Censi, 2008) and builds local maps that are used for SLAM and navigation. It includes a *reactive navigation control* module that generates a set of speed commands in

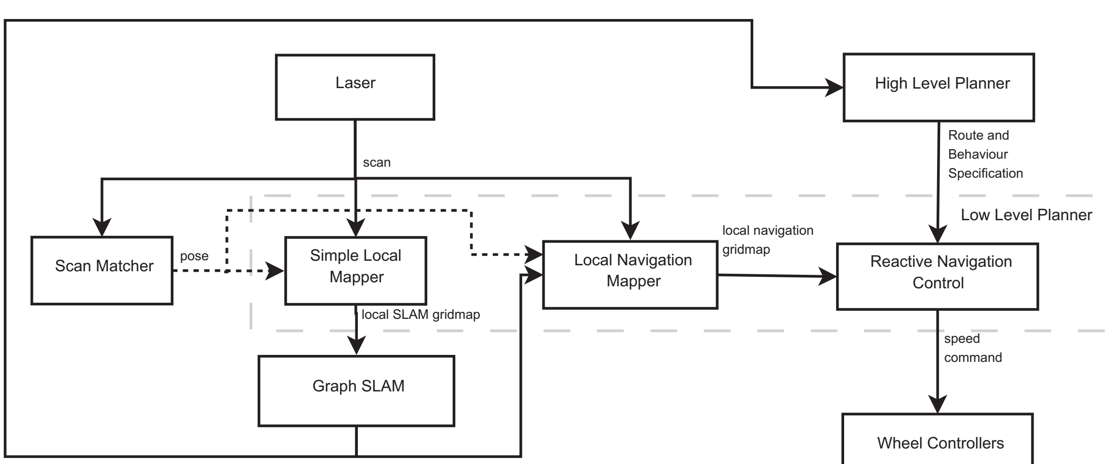


**Fig. 1.** Architecture of the mapping system.