Robotics and Autonomous Systems 60 (2012) 1078–1092

Contents lists available at SciVerse ScienceDirect

Robotics and Autonomous Systems

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

Poly line map extraction in sensor-based mobile robot navigation using a consecutive clustering algorithm

Mohamad Ali Movafaghpour, Ellips Masehian *

Faculty of Engineering, Tarbiat Modares University, Tehran, 14115-143, Iran

ARTICLE INFO

Article history:
Received 30 January 2010
Received in revised form
31 January 2012
Accepted 6 February 2012
Available online 18 February 2012

Keywords:
Mobile robot mapping
Line extraction
Sequential clustering
Rank Order Clustering

ABSTRACT

In this paper a new technique is presented for online mapping of unknown indoor environments using laser range data scans performed by a mobile robot. The developed algorithm hierarchically utilizes clustering methods to convert data points into point-clusters and eventually to line-segments. In addition to using the K-means algorithm to form appropriate point-clusters, the Rank Order Clustering (ROC) technique is used for the first time in mapping, where no preset number of clusters is required for recognizing line clusters. To do this, a set of five fuzzy membership functions are designed for calculating the Similarity Index Matrix (SIM) of line-segments, after which line-segments lying in each cluster are merged to form the final perceived lines in the constructed map. The map-building process is performed dynamically: it incrementally adds new lines to the previously calculated map-lines and merges them with the overall map. Various simulations exhibit favorable results for a mobile robot navigating in indoor environments, both with static and dynamic obstacles.

1. Introduction

The ability to explore unknown environments is essential to any autonomous mobile robot. When no representation of the surrounding is available, mobile robots face three fundamental questions, which are "Where am I?", "Where am I going?" and "How can I get there?" [1]. A system that can help a robot to model its circumstance and find its way through a workspace correctly is called an exploration system. The workspaces in which mobile robots navigate are usually unpredictable and dynamic, and cannot be completely expressed by a map in advance. For a robot to successfully function in a real world it must dynamically react to the changes in its surroundings.

Robotic exploration is composed of two fundamental abilities: map building, and path planning. A 'map' in this context denotes any one-to-one transformation of the world onto an internal representation. In navigating an unknown environment, a complete *a priori* map of the environment is missing and so must be built incrementally. This is done basically through sensor-based guidance of the robot while it discovers the environment and reaches its goal without colliding with obstacles. In [2] a 3D range scanning is implemented for map-building of 2D cluttered environments. Some works such as [3,4] benefit from interdependency of the localization and map building and try to solve them simultaneously. In this approach, which is called 'Simultaneous Localization and Mapping' (SLAM), a mobile robot can build a map of an environment and simultaneously use this map to deduce its location. Both the trajectory of the platform and the location of all landmarks are estimated online without the need for any prior knowledge of location [5]. In order for the robot to complete its perception of the environment, the scan data provided by range sensors must be matched and fused to produce a correct and reliable map of the environment.

There are two approaches commonly employed for integrating sensor readings into a perceived map: Point-based Matching and Feature-based Matching. In the first approach, the raw measurement points are directly used to build a map, as in [6] in which the problem of self-localization in an unknown, not necessarily polygonal, environment is addressed. Their method approximates the alignment of two consecutive scans and then iteratively improves the alignment by defining and minimizing a distance metric between sensor readings of the scans.

In feature-based matching, the raw scans are first transformed into geometric features such as lines, circles, arcs, etc. The extracted features are then used for matching in the next step. This approach has been studied and employed intensively in recent researches on feature extraction, feature tracking, and mapping. Being more concise, they require much less memory than point-based algorithms, while still providing rich and accurate information. Algorithms based on parameterized geometric features are therefore more efficient and stable than point-based algorithms when dealing with noisy data [7]. Among many geometric primitives, line-segments are the simplest ones.

* Corresponding author. Tel.: +98 21 82884939.
E-mail addresses: masehian@modares.ac.ir, ellips_masehian@hotmail.com (E. Masehian).

0921-8890/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2012.02.001

Since it is easy to describe most indoor environments using line-segments, poly-line mapping seems to produce promising results for indoor exploration.

Many algorithms have been proposed for mobile robotic mapping using line features extracted from 2D range data. For example, [7] provides a good survey and comparison on line extraction algorithms for indoor mobile robots, including the Split-and-Merge, Incremental, Hough-transform, Line regression, Random Sample Consensus (RANSAC), and Expectation–Maximization (EM) algorithms. In [8] a 2D scan segmentation method is developed for line extraction based on regression which does not treat single range readings in real-time, but processes a level of whole scans. The extracted information is then used for map building in a reasoning step. The authors of [9] introduce a weighted matching algorithm to estimate the robot's planar displacement by matching 2D range scans. They also compare their developed algorithm with the unweighted least-square scan matching algorithm presented in [6] and conclude that their algorithm is more robust to errors in initial estimates. The same authors in [10] suggest a line extraction algorithm for line-based mapping using weighted line fitting, and use a chi-square test to check whether two line-segments are similar enough to be merged. The best estimation of the line pairs to be merged is eventually determined based on a maximum likelihood approach.

In [11] a standard 3D perception sensor is used to improve a 2D model of an outdoor environment: the measured 3D point cloud is segmented to extract landmark points for landmark matching. Some well-recognized landmarks are then projected onto a horizontal plane to obtain a leveled range scan and to reduce computation. In [12] an algorithm is developed for feature detection in semi-structured outdoor environments which can be used for extraction of edge and circle features (e.g. planar surfaces, tree trunks or tree-like objects). In this approach, the collected data are first divided into groups through the Extended Kalman Filter (EKF), and then a fitted feature is optimized in each group.

The authors of [13] present an extended version of the Split-and-Merge method, where in each single scan, an ordered sequence of points is selected for line fitting and potential breakpoints are labeled for feature endpoint detection. They use two threshold-based line extraction and one Split-and-Merge kernels to finalize line detection.

In the study of [14] the information is obtained by ultrasonic sensors, and the workspace is first decomposed into square cells, and an Enhanced Adaptive Fuzzy Clustering (EAFC) method together with noise clustering is applied to each cell to identify at most two line-segments. Next, these line-segments are tried to be merged considering their eigenvalues as an indicator of their parallelism. The work has some drawbacks however: it needs prior setting of cell sizes, unrealistically assumes that in each cell only one or two line-segments can be clustered, and requires locating of some landmarks in different regions of the environment before navigation. All these limitations reduce the level of autonomy and adaptability of the robot.

In a recent work by Fernandez et al. [15], a method called Distance-based Convolution Clustering (DCC) is developed in which the robot's scanned points are grouped into some clusters using a convolution operation. The parameters of this method can be tuned independently of the size of the workspace. Also, another method is proposed in the same work in which map-lines are detected and appended to their proper cluster of points using a combination of Hough Transform and line tracking algorithms. A potential shortcoming of this algorithm is that it greatly relies on detecting breakpoints, which are the regions where the Euclidean distances between two consecutive scan points increase drastically. In fact, for considerably obtuse obstacle corners or highly-dense scan point clouds, the algorithm may face difficulties in defining proper clusters (which are the bases for creating map-lines).

A glance at the existing research in the field of poly-line mapping reveals that despite their wide application, most algorithms experience problems caused by predetermined thresholds such as: minimum number of points per line or curve segment, minimum physical length of a segment, maximum distance of a point to its representative line, or distance between two consecutive scan points to be clustered in the same cluster. These thresholds are to be tuned for a group of similar environments or specific sensor errors.

This paper presents an exploration system for exploring an unknown environment, and is comprised of two modules: mapping and path planning. The mapping module is a novel algorithm called Consecutive Clustering Algorithm (CCA), and the path planning module utilizes the Tangent Bug planner [16] as the local planner, embedded in a greedy global planner. The global planner sequentially prescribes new goal locations for the local planner in order to maximize the information gain in a greedy manner. The advantage of this method is its ease of implementation while the execution time is kept low. This is comparable with the information-based exploration strategy described by Amigoni and Caglioti [17] in which the next best observation point is calculated after gathering a small set of laser scans of the environment.

Some new feature extraction methods such as Zhao and Chen [18] try to work independent of prior knowledge of the environment. Focusing on this capability, the CCA is robust enough to function satisfactorily in presence of noisy data. Unlike most clustering algorithms employed in map-building tasks, an advantage of the CCA is that it does not require presetting parameters like number of clusters, length of line-segments, maximum distance in clusters when clustering and merging line segments, or even minimum number of points per segment. Such a capability is due to implementation of a clustering method called Rank Order Clustering (ROC), which was originally developed for Group Technology in [19] and is applied for the first time in the map-building discipline. In contrast to classic clustering algorithms (such as K-means) which focus on 'individual features', the ROC uses 'pair-wise similarity' of objects which leads to its advantage over other clustering algorithms and makes it capable of recognizing hidden patterns with no peripheral knowledge about the correct number of clusters. Also, many parameters (excluding the main three input parameters discussed in Section 4.2) like sensor thresholds, similarity criteria, fuzzy membership coefficients, and cluster pruning factors are not set before the robot's navigation, but tuned dynamically and adaptively relative to the features of the already sensed data. For securing a robust functionality for the CCA, a statistical-based analysis is proposed for tuning the algorithm's three input parameters to their optimal values.

The paper is organized as follows: Section 2 describes the Consecutive Clustering Algorithm in detail, Section 3 provides experimental and simulation results, and Section 4 presents discussions on the algorithm's time complexity and parameter tuning. Finally, conclusions and directions for future research are presented in Section 5.

2. The Consecutive Clustering Algorithm (CCA)

The Consecutive Clustering Algorithm proposed in this paper can be classified as a learning-by-clustering algorithm. The general steps of the algorithm are as follows:

Step 0. *Path planning*: A goal point is set for the robot by the global planner in the most unexplored area of the environment. The robot follows the current goal point using the Tangent Bug technique [16].