Robotics and Autonomous Systems 58 (2010) 720–726

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Clustering and line detection in laser range measurements

Carlos Fernández, Vidal Moreno *, Belen Curto, J. Andres Vicente

University of Salamanca, Department Computers and Automation, Pl. Merced s/n. 37008, Salamanca, Spain

ARTICLE INFO

Article history:
Received 23 March 2009
Received in revised form
23 October 2009
Accepted 29 October 2009
Available online 14 November 2009

Keywords:
Feature extraction
Robot mapping

ABSTRACT

This article presents two algorithms that extract information from laser range data. They are designed to work sequentially. The first method (DCC) separates the data into clusters by means of a convolution operation, using a high-pass filter. The second one (REHOLT) performs line detection in each of the clusters previously discovered. The reliability of the algorithms devised is tested on the experimental data collected both indoors and outdoors. When compared with other methods found in the literature, the ones proposed here prove to achieve higher performance.

1. Introduction

One of the most challenging difficulties ever faced by the mobile robotics community is the navigation problem. It chiefly consists of deciding how to get from point A to point B, avoiding all possible collisions [1]. Many research efforts have been channeled into solving this problem, and as a result there are specific approaches that successfully solve the problem for certain types of scenarios. However, if an unrestricted environment is taken into account, a global solution is still to be found. It is highly probable that a navigation method designed to work on a flat floor within an indoor environment will not perform well if the robot is moved to a rough terrain inside a forest, for instance. This is due to the fact that robots perceive the world by means of sensors, and the usefulness of every sensor varies depending on the place where it is employed. Consequently, the navigation problem is undoubtedly still interesting nowadays.

According to some researchers ([2,3]), the navigation problem can be subdivided into at least three other tasks, namely: path planning, localization and map building. The first task can be solved without excessive difficulty as long as the map and localization are accurate. Therefore, the actual problems are how to determine the pose of a robot and how to build a map of its environment. It is well known that using proprioceptive sensors for localization is only a partial solution, since it is not possible to bound the error in the estimation. Therefore, robots are commonly equipped with exteroceptive sensors so that the error can be kept small. This type of sensor also becomes essential when it comes to build maps, because it is the only way a robot can get information about its surroundings.

Localization and map building algorithms rely mainly on two factors: the type of environment and the sensors employed. Throughout this work, we will focus on a laser scanner — model SICK LMS 221 — operating both in indoor environments and in structured outdoor environments. Lasers have several advantages with respect to other sensors: they are more accurate and have a higher sampling rate than sonars, and range measurements are easier to interpret than images from cameras, for example. Laser data are usually a set of points $\mathcal{R} = \{(\rho_i, \varphi_i)\,|i = 1, \ldots, n\}$, where (ρ_i, φ_i) are the polar coordinates of the i-th scan point, where the location of the sensor is the origin of this polar coordinate system. The laser, using a fixed angular resolution $\varphi_i - \varphi_{i-1} = \Delta\varphi$, acquires the points in an ordered sequence, starting at direction $\varphi_{min} = \varphi_1$ and stopping at $\varphi_{max} = \varphi_n$. It is important to note that when laser data are acquired with the robot moving, they might be deformed, depending on the robot speed and the scanning time [4].

Regarding the processing of laser data, there are two main approaches: metrical or topological. Nevertheless, hybrid metrical/topological approaches also exist, such as the *Atlas* framework [5]. Metrical techniques work with raw data, using point-based matching techniques [6] or particle filters [7], for example. A major drawback of these methods is that the storage requirements grow disproportionately for large environments, and performance hitches arise. On the other hand, topological approaches extract features from raw data, such as corners and line segments. Those features can be subsequently used as building blocks for localization or map building. This is largely beneficial since simple parametric models are used to describe large amounts of data, and it is possible to maintain the scalability even if the dataset grows enormously. In man-made environments (both indoors and outdoors), planar surfaces are very common and they are typically modeled by line segments. Some examples include corridor or room walls, doors, tables, building facades, etc.

* Corresponding author.
E-mail address: vmoreno@abedul.usal.es (V. Moreno).

0921-8890/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2009.10.008

However, prior to perform line extraction, it is advisable to detect breakpoints, which are discontinuities that take place during the scanning process [4]. Typical breakpoints are the result of detecting open doors along a wall, or perceiving objects that occlude distant surfaces. Breakpoint detection is useful for clustering the points of a laser scan, and additionally it prevents line extraction algorithms from connecting distant neighbor clusters.

In what follows, we present two novel algorithms intended to be applied sequentially to laser range data: DCC and REHOLT. The first one is a clustering method based on the convolution operation. It creates clusters that are to be passed to the line extractor, REHOLT, which combines the advantages of two classical algorithms: *Line Tracking* and the *Hough Transform*.

In Section 2, we discuss some existing clustering and line extraction methods. Sections 3 and 4 describe the algorithms proposed in this article. After that, an experimental comparison can be found in Section 5, followed by our conclusions (Section 6).

2. State-of-the-art

As stated before, in this article we will consider two different groups of algorithms applied to laser data: clustering and line extraction. In this section, we discuss some of the most popular methods found in the literature.

2.1. Clustering algorithms

The main purpose of clustering methods, when applied to laser data, is to test whether a discontinuity exists between two consecutive scan points. In such a case, the dataset will be split into two clusters. The most common approach found in the literature matches the pseudocode found in Algorithm 1, where $D(p_i, p_{i+1})$ indicates the euclidean distance between two consecutive points. The main difficulty here lies in how to specify a criterion for deciding the threshold value D_{max}. If a fixed value was chosen, the results would not be very promising. For instance, when the laser scans a long wall, the separation between points becomes bigger and bigger, and it may happen that $D(p_j, p_{j+1}) > D_{max}$ for every $j \geq k$. This would imply that from a certain point k, every point would be considered a cluster on its own, and that is why it is more common to set D_{max} adaptively.

Algorithm 1 Simple Clustering Algorithm

```
P ← {p_1, ..., p_n}                    ▷ Laser Measurements
for i ← 1, ..., n − 1 do
    if D(p_i, p_{i+1}) > D_max then
        p_{i+1} ∈ new cluster
    end if
end for
```

For example, Dietmayer et al. [8] define the value of the threshold distance according to a method developed for basic object classification intended to distinguish vehicles, persons and small objects. The value relies on two constants: one is a function of the angular resolution of the scanner ($\Delta\varphi$), while the other must be manually set and allows an adjustment to noise. Santos et al. [9] propose a method based on the previous one. They argue that Dietmayer's technique is highly dependent with respect to the distance between the laser and the objects detected. In order to correct that they devise a new formula, adding a new parameter which must be manually tuned. Finally, Borges et al. [4] suggest a method called *Adaptive Breakpoint Detector* in which they specify D_{max} as a function of the noise in the measurement and a virtual line determined by an angle λ. Such a virtual line passes through point p_{n-1} and makes an angle λ with respect to the scanning direction φ_{n-1}. Thus, the technique aims to extrapolate the worst acceptable point p_n.

The main disadvantage of these methods, as will be stated later on, is that the experimental results show that their parameters must be tuned depending on the size of the environment. For example, they do not perform equally well if they are tuned for a small room and then they are tested on a large hall.

2.2. Line extraction algorithms

Line segments are the simplest geometric primitives that can be used to describe man-made environments. The challenge of line extraction procedures is to fit lines to sets of points with the best possible accuracy. If the reader is interested in a more comprehensive analysis of the subject, please refer to [10].

Probably, the most straightforward example of a line extractor is the *Successive Edge Following* (SEF) algorithm [11]. Briefly, it considers that a new line begins when the distance between two consecutive scan points exceeds a certain threshold. It matches closely much the pseudocode of the Algorithm 1, but applied to line detection. Another algorithm that it is remarkable not only for being uncomplicated but also for its reduced time complexity is the *Line Tracking* (LT) algorithm [4], also known as *Incremental*. Basically, it starts off by building a line model which passes through the first and second scan points, successively adding a new scan point if the line criterion is validated. Otherwise, the line is terminated and a new one is started, repeating the algorithm until the end of the dataset is reached.

Unlike the two previous methods, the *Iterative End Point Fit* algorithm (IEPF) [12] is recursive. It begins by constructing a line using the first and last scan points. Next, it finds the most distant point (p_k) from the line, and if it is far enough, two subsets are created taking p_k as the splitting point. This procedure is repeated recursively for all the subsets until the validation criterion fails. The Split & Merge algorithm [13] is twofold. Its first phase (split) is similar to the IEPF method. Nevertheless, it differs from it in that it adds another phase (merge) in which collinear segments are fused together if the angle between them is sufficiently small.

In contrast to the previously discussed methods, the *RANdom SAmple Consensus* (RANSAC) algorithm [14] makes use of a probabilistic approach, and it robustly fits models in the presence of data outliers. Firstly, it constructs a line R using two randomly chosen points from the initial point set. Secondly, a consensus set is created, formed by line inliers, and if it is big enough, the line R is readjusted to the points included in the consensus set. Otherwise, the algorithm is repeated until a proper consensus set is found or the maximum number of loop iterations is reached.

A main disadvantage of SEF, LT, IEPF and S&M is that they fit lines to a set of points using fast but non-robust methods, such as *least squares* for example, which is known to have problems with outliers [14]. On the other hand, RANSAC is a robust method, but its processing time and results are not always the same, because it is a nondeterministic algorithm.

3. Distance-based Convolution Clustering

In this section, we propose a new clustering technique based on convolution sums, named Distance-based Convolution Clustering (DCC). The method basically analyzes a laser scan in search of breakpoints, and once a breakpoint is detected, a new cluster is created.

The algorithm works as follows: let $P = \{p_1, \ldots, p_n\}$ be a scan point set, and let $D = \{d_1, \ldots, d_{n-1}\}$ be a vector containing the set of euclidian distances between each pair of consecutive points (p_j, p_{j+1}). Then, the breakpoint set is computed by means of the convolution operation in its discrete form, which is given by (1). There, σ is the standard deviation of the laser scanner, whereas $k[-\frac{m-1}{2}, \ldots, \frac{m-1}{2}]$ is a high-pass convolution kernel of size m, where m is necessarily an odd number.