Robotics and Autonomous Systems 60 (2012) 296–308

Contents lists available at SciVerse ScienceDirect

Robotics and Autonomous Systems

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

Linear-time robot localization and pose tracking using matching signatures

Asim Kar *

Division of Remote Handling and Robotics, BARC, India

ARTICLE INFO

Article history:
Received 25 March 2010
Received in revised form
8 August 2011
Accepted 18 November 2011
Available online 26 November 2011

Keywords:
Global localization
Data association
SLAM
Multi-hypothesis tracking
Signature matching
Feature-based navigation

ABSTRACT

A feature-based method for global localization of mobile robot using a concept of matching signatures is presented. A group of geometric features, their geometric constraints invariant to frame transform, and location dependent constraints, together are utilized in defining signature of a feature. Plausible global poses are found out by matching signatures of observed features with signatures of global map features. The concept of matching signatures is so developed that the proposed method provides a very efficient solution for global localization. Worst-case complexity of the method for estimating and verifying global poses is linear with the size of global reference map. It will also be shown that with the approach of random sampling the proposed algorithm becomes linear with both the size of global map and number of observed features. In order to avoid pose ambiguity, simultaneous tracking of multiple pose hypotheses staying within the same framework of the proposed method is also addressed. Results obtained from simulation as well as from real world experiment demonstrate the performance and effectiveness of the method.

1. Introduction

One of the most studied problems in mobile robotics is SLAM (Simultaneous Localization and Map Building) [1–7]. Localization and data association are two integral parts of SLAM. Mobile robot localization is broadly divided into two categories: position tracking and global localization. In position tracking [7–9], the initial estimate of the robot pose is known and periodical correction in its pose is done by searching correspondences of sensor data in a small region in the given map. During continuous tracking, pose uncertainty is usually small, search space is limited around the pose estimate and many position tracking methods proposed have been proven to be precise and efficient. But, they fail if the pose error grows large, the initial robot location is unknown, or the robot is lost, which is known as the *kidnapped robot problem* [10]. On the other hand global localization [10–13] finds a robot pose from scratch but is a more complex problem and is found to be very expensive in terms of both computation time and memory requirements. Moreover, it may suffer from the *perceptual aliasing* problem, i.e., if the environment contains many regions of similar shape, it finds multiple solutions and selecting the correct one out of many similar choices is difficult. In order to overcome such ambiguity, a global localization system is expected to have the capacity of tracking distinct multiple pose hypotheses in parallel.

Global localization with the capacity of multi-hypothesis tracking is studied with the approaches of multi-modal Gaussian distributions [14–16], ML (Markov localization) [10], and MCL (Monte Carlo localization) [11]. ML represents robot's belief by a posterior probability density over all pose states distributed across a 3-D grid. Though ML is very robust and fail safe, it is extremely slow and unfit for real-time application. A tradeoff between pose accuracy (grid resolution) and computational efficiency as well as various heuristics are to be adopted to run it in reasonable time. MCL overcomes disadvantages of ML by representing robot's belief by a fewer number of weighted samples. However, care must be taken in selecting the sample size as fewer samples might prematurely collapse to a wrong belief. Moreover, naive implementation of MCL cannot handle the kidnapped robot problem. Several extensions have been suggested to overcome such shortcomings (see [17] for details). On the other hand MCL is multi-modal, fast, and can cope with sensor uncertainty easily. It is probably one of the most popular and widely used multi-hypothesis localization algorithms. Other approaches like [18] propose hybrid methods to combine advantages of extended Kalman filters and POMDP. Reverse Monte Carlo Localization [19] aims to combine ML and MCL to take advantage of both and overcome their disadvantages.

Another approach is *search in correspondence space* [20–24] where geometric features such as points, lines, arcs, etc., are extracted from high-dimensional sensor data and are matched against stored map features. A key advantage of this approach is that it enables us to maintain exactly as many pose hypotheses as necessary and as few as possible. The most employed technique to search correspondence space is the interpretation tree search approach [25] and the worst case complexity of interpretation tree search is exponential, i.e., $O((m+1)^n)$ including the star node, where n is the number of observed features and m is the number of map features. However, in reality only a small portion

* Tel.: +91 2225592202.
E-mail address: asim@barc.gov.in.

0921-8890/$ – see front matter
doi:10.1016/j.robot.2011.11.010

of the tree is subject to the interest of search concerning current sensory observation. Different location dependent and location independent geometric constraints, individual compatibility and joint compatibility [20–22,26] are utilized in order to prune the tree to reduce the search space. While a physics-based sonar model is used in [24] to derive geometric constraints, the global interpretation tree is divided into two sub-trees based on horizontal and vertical edges in [23]. The worst-case search in case of spurious measurements is still exponential. To overcome this, [27] introduces a concept of *locality* and achieves $O(m(c+1)^n)$ worst-case complexity, which is linear with the size of the global reference map and exponential with the size of *locality* $c(\leq m)$. With consideration of grid sampling, individual feature-to-feature correspondence and a strategy of voting, worst-case complexity linear both in number of map features (m) and in number of observed features (n) is achieved in [28] at the cost of pose accuracy (pose accuracy depends on grid resolution) and true positive solutions. Another technique used to represent the correspondence space is by MCG (Maximum Common Graph) in which nodes are unary compatible pairings and undirected edges are binary compatible pairings [5]. Finding the largest set of unary–binary compatible pairings is equivalent to searching for a maximum clique in the correspondence graph. Searching for a maximum clique in a graph is NP complete in general. Given the correspondence graph is very sparse, many efficient implementations can be found in the literature (see [29,30] for example). CCDA (Combined Constraint Data Association) proposed in [5] first constructs a unary–binary compatible MCG and then searches for a maximum clique in the common graph. Computational complexity in constructing MCG reported in [5] is quadratic with the size of the map. Average case computational complexity of maximum clique search is quadratic with the node size, if prior pose estimation is available. Otherwise computational complexity without prior pose estimation is not well defined.

In this work we propose a simple, easy to implement, and very efficient feature-based method for multi-hypothesis localization using a concept of matching signatures. We consider a number of features and their geometric constraints together in order to interpret n measurements (throughout this paper we interchangeably use the terms measurement, local feature, and observed feature to refer to geometric features extracted from sensor data) as n observed signatures and m map features as m reference signatures. We generate all plausible global poses by matching observed signatures with signatures of map features and show that worst-case computation in order to generate all plausible global robot poses and verifying them is $O(n^3mc^2)$. Here $c \leq m$ but does not refer to size of a local region in the map that is generally determined depending on either sensor maximum range or independent local maps [31] or simultaneous visibility of features [27]. We include the entire global map and make use of location independent binary constraints to determine c with very little computation cost. Details are presented in Section 4. We will also show that with approach of random sampling as proposed in [27] our algorithm becomes linear with both the size of the global map (m) and the size of the measurements (n). Section 2 provides a definition of signature and Section 3 deals with the concept of matching signatures and in turn with the problems of data association, hypothesis (hypothesis always refers to pose hypothesis in this paper) generation and verification. The global pose generation algorithm, the computational complexity of it and the multi-hypothesis tracking algorithm are described in Section 4. Run time efficiency of this method is also compared with the JCBB-based (Joint Compatibility Branch and Bound) relocation algorithm proposed in [27], using simulated and DLR data sets [32]. Comparison results and results of multi-hypothesis tracking with a real robot are discussed in Section 5, followed by conclusions in Section 6.

2. Definition of a signature

Let $\mathcal{F} = \{f_i\}_{i=1}^n$ be a set of geometric features in a common reference frame. To each feature f_i in $\mathcal{F}$ we associate a signature s_i. A signature s_i is a tuple that consists in a set of unary constraints $\mathcal{U}_i$, a set of binary constraints $\mathcal{B}_i$, and a set of location-dependent constraints $\mathcal{R}_i(\mathbf{x})$,

$$s_i(\mathbf{x}) = \{\mathcal{U}_i, \mathcal{B}_i, \mathcal{R}_i(\mathbf{x})\}. \quad (1)$$

Unary constraint applies to intrinsic properties of a feature. Examples are feature type, color, texture or dimension such as length or width. Let u_i encodes unary constraint of feature f_i, and suppose f_i has a total nu number of unary constraints, then

$$\mathcal{U}_i = \{u_j\}_{j=1}^{nu}. \quad (2)$$

Unary constraints are defined for line segments (segment length) and circular features (radius). They are undefined for (x, y)-point features. In case of multiple feature types, the type itself can also be considered a unary constraint.

A binary constraint always applies to a pair of features $f_i - f_j$. Examples include measures such as relative distance or relative angle. Let b_{ij} denotes a binary constraint (distance of f_j from f_i, angle of f_j with respect to f_i, etc.) between features f_i and f_j, then there are n possible binary constraints of feature f_i in $\mathcal{F}$

$$\mathcal{B}_i = \{b_{ij}\}_{j=1}^n. \quad (3)$$

Both, unary and binary constraints are location independent, that is, they are invariant to the frame transform of the features in $\mathcal{F}$. In contrast, a location-dependent constraint is a function of a pose $\mathbf{x} = (x, y, \theta)^T$. Let $\mathbf{x}$ define a reference frame in which all features in $\mathcal{F}$ are expressed. Then, the location dependent constraint of all features in $\mathcal{F}$ is defined as

$$\mathcal{R}_i(\mathbf{x}) = \{r_j(\mathbf{x})\}_{j=1}^n \quad (4)$$

where $r_j(\mathbf{x})$ denotes location of the feature f_j in frame $\mathbf{x}$ and is expressed depending upon its type. For an example of a point feature, it can be the (x, y) location in the frame. For a line feature, it can be expressed by (d, α), where d is the normal distance of the line from the origin of frame $\mathbf{x}$ and α measures angle of that normal line with respect to the positive x axis of frame $\mathbf{x}$.

Though unary constraint, binary constraint and location dependent constraint validate compatible associations between measurements and map features, they do not confirm whether a measurement is at least partially within the region occupied by its target map feature. This is especially relevant for finite-dimension features such as line segments, arcs, etc. Such a problem is overcome by *extension constraint*, details of which can be found in [22,33]. An extension constraint, though, is not included in the definition of signature but is used explicitly during signature-based data association processes.

3. Signature-based data association, hypothesis generation and verification

This section deals with the process of matching two signatures and in turn with the problems of data association, hypothesis generation and hypothesis verification. The proposed signature matching method involves two steps: (a) generation of hypothesis using location independent unary and binary constraints, and (b) verification of hypothesis that includes searching of compatible supporting set of pairings using both location dependent and location independent constraints. The process of matching two signatures is illustrated in Fig. 1. To describe this process let us assume that a signature $s_i(\mathbf{o}) = \{\mathcal{U}_i, \mathcal{B}_i = \{b_{ij}\}_{j=1}^n, \mathcal{R}_i(\mathbf{o}) = \{r_j(\mathbf{o})\}_{j=1}^n\}$ of an observed feature e_i ('**o**' refers to local frame attached with the robot body) is to be compared with a signature $s_k(\mathbf{r}) = \{\mathcal{U}_k, \mathcal{B}_k = \{b_{kl}\}_{l=1}^m, \mathcal{R}_k(\mathbf{r}) = \{r_l(\mathbf{r})\}_{l=1}^m\}$ of a map feature f_k