Applied Soft Computing 50 (2017) 260–279

Contents lists available at ScienceDirect

Applied Soft Computing

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

Multi-objective evolutionary approach to prevent premature convergence in Monte Carlo localization

Chiang-Heng Chien, Wei-Yen Wang, Chen-Chien Hsu *

Department of Electrical Engineering, National Taiwan Normal University, 162 He-Ping East Rd., Sec. 1, Taipei 10610, Taiwan

ARTICLE INFO

Article history:
Received 21 April 2016
Accepted 14 November 2016
Available online 21 November 2016

Keywords:
Monte carlo localization
Premature convergence
Global localization
Multi-objective particle swarm optimization
Mobile robots

ABSTRACT

In this paper, we propose a global localization algorithm for mobile robots based on Monte Carlo localization (MCL), which employs multi-objective particle swarm optimization (MOPSO) incorporating a novel archiving strategy, to deal with the premature convergence problem in global localization in highly symmetrical environments. Under three proposed rules, premature convergence occurring during the localization can be easily detected so that the proposed MOPSO is introduced to obtain a uniformly distributed Pareto front based on two objective functions respectively representing weights and distribution of particles in MCL. On the basis of the derived Pareto front, MCL is able to resample particles with balanced weights as well as diverse distribution of the population. As a consequence, the proposed approach provides better diversity for particles to explore the environment, while simultaneously maintaining good convergence to achieve a successful global localization. Simulations have confirmed that the proposed approach can significantly improve global localization performance in terms of success rate and computational time in highly symmetrical environments.

1. Introduction

Reliably offering a dynamic state estimation according to noisy sensor measurements is a critical topic for a mobile robot to autonomously localize itself when it comes to navigation. As far as localization is concerned, global localization is an utmost matter to be considered, where the robot has to localize itself under an unknown initial pose based on noisy and incomplete observed information. Over the past years, global localization has received considerable attention from both industrial and academic communities, and a flurry of new algorithms and techniques have emerged, including grid-based approaches [1], topological approaches [2], Monte Carlo Localization (MCL) [3,4] multi-hypothesis tracking [5], and so on. Among them, the MCL algorithm, or particle filter localization, is the best-known probabilistic techniques inheriting all the advantages of the Sequential Markov Chain [6]. Basically, MCL realizes a recursive form of a Bayesian filter (BF) [7,8], so as to carry potentially multi-modal probabilistic density functions (pdf) and to relax the linearity and Gaussian assumptions.

The basic idea of MCL is to establish population-based representations of the entire pdf, which describes the probability of the true pose of a real robot. Each sample, or the so-called "particle", serves as a belief of a state. By iteratively estimating the optimal solution, MCL can localize a robot as it moves, by virtue of its capability to carry out the Bayesian filter continuously even though the sensor readings are nonlinear. Therefore, under highly non-linear and non-Gaussian noise environments, MCL has been shown very effective in many cases [9].

Unfortunately, conventional MCL could not stably and reliably localize a robot all the time, particularly when the robot is in a highly symmetrical indoor environment consisting of rooms or corridors of identical appearance. Under such cases, MCL has to be able to solve these potentially ambiguous situations due to similar weights of particles that easily lead to quickly loss of population diversity, which results in particles being trapped into local optima. As a consequence, premature convergence [9,10] of particles occurs, resulting in catastrophic localization failure.

To improve the performance of global localization of MCL by avoiding premature convergence, various approaches have been proposed, which can be categorized into two families: (1) MCL-based algorithms and (2) geometric features based algorithms. The first family includes Clustered MCL [8], Adaptive Dynamic Clustered MCL [11], Coevolution Based Adaptive MCL (CEAMCL) [12], and Local Selection Based MCL [13]. Clustered MCL and Adaptive Dynamic Clustered MCL incorporated the idea of clustering for particles and modified the proposal distribution considering the

* Corresponding author.
E-mail addresses: jhsu@ntnu.edu.tw, wilonda@gmail.com (C.-C. Hsu).

http://dx.doi.org/10.1016/j.asoc.2016.11.020
1568-4946/

probability mass of various clusters to retain multiple hypothesis, where the former one adopted static number of particles while the latter dynamically evolved the clusters and adaptively changed the population size. Although the success rates of both approaches were improved, the computational burdens were too heavy for real-time robot navigation systems. CEAMCL combined Clustered MCL with co-evolutionary mechanism of ecological species by applying crossover and mutation operators to search for optimal samples in each species. Though the computational time is reduced, the success rate of global localization is unfortunately sacrificed. Local Selection Based MCL integrated the niching methods of local selection algorithm, which took less time for global localization. Nevertheless, the success rate was far from satisfactory, and a great amount of additional parameters of local selection needed to be properly set. Apart from the MCL-based algorithms, the geometric features based approaches include a method using the concept of matching signatures obtained from environmental features [14], a line-segment relation matching technique integrating hypothesis tracking framework [15], and a feature driven method to track multiple hypothesis [16]. Nevertheless, geometric feature based methods give unfavorable results if their maps are incomplete or if any unforeseen obstacles are encountered. Though the above mentioned algorithms had tried to recover localization failure from premature convergence, there is still room for further improvements in terms of success rate and computational time.

In this paper, we propose a global localization approach based on MCL, where a MOPSO employing a novel archiving strategy based on dominated tree method [18] is used to deal with premature convergence of the population in highly symmetrical environments. To detect whether premature convergence occurs, three probabilistic-based rules are proposed. As soon as one of the rules is satisfied, the MOPSO is introduced to obtain a uniformly distributed Pareto front by optimizing two objective functions characterizing weights and mean inter-particle distance [17], which respectively represent convergence and diversity of the MCL population. The Pareto frontiers are then adopted to facilitate the resampling process of MCL. As a result, the proposed approach is capable of balancing weights as well as distribution of particles in a population, allowing better exploration of the whole environment while maintaining its convergence ability at the same time. Therefore, higher opportunities are given for particles to converge to the real pose of the robot.

The remainder of this paper is organized as follows. Section 2 gives the preliminaries of the MCL algorithm. Section 3 describes the preliminaries of multi-objective particle swarm optimization including the general mathematical statements of multi-objective optimization and the archiving problems. The overall framework of the proposed algorithm is described in details in Section 4. A set of simulation results is illustrated in Section 5, and the conclusion is given in Section 6.

2. Preliminaries of Monte Carlo localization

Assume that the current state x_k depends only upon the previous state x_{k-1}, implying that the environment is a Markov chain process. On the basis of a recursive form of a Bayesian filter [19,20], a belief $Bel(x_k)$ of the current state x_k of Monte Carlo localization represented by a Gaussian distribution can be obtained from (1):

$$Bel(x_k) = \eta p(o_k|x_k) \int p(x_k|x_{k-1}, u_{k-1}) Bel(x_{k-1}) dx_{k-1}, \tag{1}$$

where $\eta = p(o_k|u_{k-1}, \ldots, o_0)^{-1}$ is a constant, $p(x_k|x_{k-1}, u_{k-1})$ is the sensor model, $p(o_k|x_k)$ is the motion model, and $Bel(x_k)$ is the current estimated state obtained from the previous estimated state $Bel(x_{k-1})$. Hence, (1) is the updated equation of Bayesian filters. Realization of (1) closely relates to the motion and sensor models,

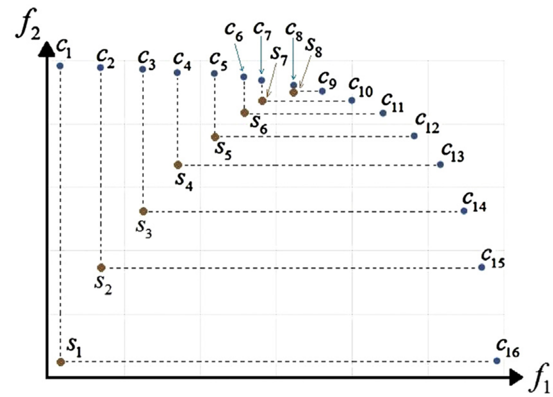

Fig. 1. The construction of a dominated tree.

in which the former is a probabilistic generation approach of robot dynamics [8], while the latter is the sensor readings with noise, respectively.

Eq. (1) is the key idea of particle filter [21]. MCL, however, is actually representing the belief function $Bel(x_k)$ as random state particles, or posterior beliefs. Each particle can be seen as an estimate of a real robot, and the continuous belief $Bel(x_k)$ is approximated from the discrete function determined by the set, being described by n weighted particles distributed according to

$$Bel(x_k) = \left\{ x_k^{[i]}, w_k^{[i]} \right\}_{i=1,\ldots,n}. \tag{2}$$

In (2), each particle $x_k^{[i]}$ has an associated weight $w_k^{[i]}$, or the importance factor. The process of approaching the optimal result and approximating the belief $Bel(x_k)$ in the particle filter method, namely the sampling importance resampling (SIR) algorithm [22], is the implementation of (1) by MCL. Even though the robot is moving, estimating the real pose of the robot can still be obtained since the SIR is time-variant. Basically, the SIR algorithm comprises three steps [22]:

Step 1: Prediction: Distribute P samples from state transition probability, which is the conditional Bayesian probability or the posterior:

$$x_k^{[i]} \sim p\left(x_k|x_{k-1}^{[i]}, u_k\right), \tag{3}$$

where the particle, x_k at time k is conditioned on particle at the previous time *k-1* and the control signal u_k at time k, where i is from 1 to P. In the MCL algorithm, the control signal is referred to the robot's odometer readings with Gaussian noise.

Step 2: Weight Assignment: By comparing virtual sensor measurements of each sample with real sensor readings, weights for all particles can be computed as a probability approach shown in (4).

$$w_k^{[i]} = p\left(o_k|x_k^{[i]}\right). \tag{4}$$

The weight probability $w_k^{[i]}$ of particle i at time k is conditioned on sensor measurements o_k. The implication of weight for each particle is the similarity in sensor model between the particle and the real robot. Note that in SIR algorithm, the weight for each particle is usually normalized.

Step 3: Resampling: Distribute new samples on the basis of normalized weights of particles. The higher the weight, the more probable it is that a particular particle can be distributed again into the subsequent iteration.

Monte Carlo Localization continuously repeats the routine from *Step 1* to *Step 3* for $\hat{M}$ iterations. Although the algorithm is under a variety of assumptions, it is certain that the samples will converge to the global optimal solution as $\hat{M}$ approaches infinity, if