Robotics and Autonomous Systems 57 (2009) 441–450

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Differential evolution solution to the SLAM problem

Luis Moreno [a,*], Santiago Garrido [a], Dolores Blanco [a], M. Luisa Muñoz [b]

[a] *Department of Systems Engineering and Automation, Universidad Carlos III, Avda. de la Universidad 30, 28911 Leganés, Madrid, Spain*

[b] *Department of Computer's Architecture, Universidad Politécnica de Madrid, Campus de Montegancedo, Facultad de Informática, Madrid, Spain*

ARTICLE INFO

Article history:
Received 13 March 2007
Received in revised form
26 May 2008
Accepted 28 May 2008
Available online 18 June 2008

Keywords:
Evolutive algorithm
SLAM
ELF
Differential evolution
Mobile robot

ABSTRACT

A new solution to the Simultaneous Localization and Modelling problem is presented in this paper. The algorithm is based on the stochastic search for solutions in the state space to the global localization problem by means of a differential evolution algorithm. This non linear evolutive filter, called Evolutive Localization Filter (ELF), searches stochastically along the state space for the best robot pose estimate. The set of pose solutions (the population) focuses on the most likely areas according to the perception and up to date motion information. The population evolves using the log-likelihood of each candidate pose according to the observation and the motion errors derived from the comparison between observed and predicted data obtained from the probabilistic perception and motion model.

The proposed SLAM algorithm operates in two steps: in the first step the ELF filter is used at local level to re-localize the robot based on the robot odometry, the laser scan at a given position and a local map where only a low number of the last scans have been integrated. In the second step, the aligned laser measures and the corrected robot poses are used to detect whether the robot is revisiting a previously crossed area (i.e., a cycle in the robot trajectory exists). Once a cycle is detected, the Evolutive Localization Filter is used again to estimate the accumulated residual drift in the detected loop and then to re-estimate the robot poses in order to integrate the sensor measures in the global map of the environment.

The algorithm has been tested in different environments to demonstrate the effectiveness, robustness and computational efficiency of the proposed approach.

Published by Elsevier B.V.

1. Introduction

Localization and map building are closely linked problems, and learning maps need to simultaneously solve both problems. These problems are often referred to as *simultaneous localization and mapping*(SLAM). Building maps when robot poses are known is a tractable problem with limited computational complexity, because it is only necessary to manage the error associated with each individual sensor measurement at the time of modelling. However, in the SLAM case, uncertainty in measurements, uncertainty in robot pose estimates and a partially learned map, which contains the residual errors (the difference between the true pose value and the pose estimate) remaining after the localization process make the SLAM problem complex.

Two main approaches to effectively solve the simultaneous localization and mapping problem have emerged in the last decade. The first approach uses a feature-based model of the environment and the extended Kalman filter to manage the associated uncertainty. This approach is extremely compact, and its computational cost has been considerably improved in the most recent work. On the other hand, however, the linear nature of the basic method requires linearisation of the motion and perception models which causes problems in the long term. Moreover, the technique has difficulties modelling many environments due to the limited set of feature models used (e.g., cluttered laboratories or offices).

The second group of solutions uses particle filters to obtain a solution to the SLAM problem. This group of solutions can use certainty grid map models, or a feature map to represent the environment [27], and sequential Monte Carlo methods [7] to estimate the posterior probability distribution functions. This approach has proved to be very robust from a statistical point of view in the management of the uncertainties present in the problem. Its disadvantage is that the number of particles required increases the computational cost, and the algorithm robustness is heavily dependent on this because each particle has a statistical significance associated with it.

In spite of the advances in stochastic search optimization methods in the last decade, they have received limited attention in the field of localization and SLAM problems. In this work we present a new solution to the grid-based SLAM problem, based on the stochastic search of the best pose estimate. This approach uses

* Corresponding author. Tel.: +34 91 624 9918; fax: +34 91 624 9430.
E-mail addresses: moreno@ing.uc3m.es (L. Moreno), sgarrido@ing.uc3m.es (S. Garrido), dblanco@ing.uc3m.es (D. Blanco), mmunoz@fi.upm.es (M.L. Muñoz).

0921-8890/$ – see front matter. Published by Elsevier B.V.
doi:10.1016/j.robot.2008.05.005

a differential evolution method [24] to perturb the possible pose estimates contained in a given set, until the optimum is obtained. By properly choosing the cost function, a maximum *a posteriori* estimate is obtained. This method is applied at a local level to re-localize the robot and at global level to solve the data association problem. The method proposed integrates sensor information in the map only when cycles are detected, and the residual errors are eliminated, avoiding a high number of modifications in the map or the existence of multiple maps, thus decreasing the computational cost compared to other solutions.

Our approach has been validated on a set of data obtained from our experiments, and data extracted from the Radish repository. The results show that the proposed method is able to satisfactorily close environment cycles, to generate accurate metric maps.

The paper is organized as follows. A global evolutive localization filter (ELF) is formulated and explained in Section 3. Next, Section 4 introduces the ELF algorithm details, Section 5 deals with the key aspects of the ELF algorithm application to the SLAM problem and the SLAM algorithm proposed. The following sections present the experimental results, and discuss the advantages and disadvantages of the algorithm.

2. State of the art

The SLAM problem has been one of the most interesting theoretical problems in mobile robotics since 1986 when the seminal work of Smith, Self and Cheeseman [23,22] introduced the concept of a stochastic map to establish uncertain spatial relationships between features detected in the environment. They use a covariance matrix to encode the correlations between all pairs of landmarks included in the map, and the Extended Kalman filter provides the mechanism for integrating and updating new available information in the map. This approach uses some kind of geometrical features (points or segments) as the basic information element of the map. The stochastic map idea has been widely used in the last ten years but two problems have attracted the attention of many researchers: the computational complexity of the map grows quadratically with the number of landmarks (scaling problem), and the second is related with the data association problem (convergence problem).

The computational drawback derives from the dimension increase of the covariance matrix that for a two-dimensional point feature mapping of the world is $(2N + 3)^2$, where N is the number of landmarks included in the map. Due to the correlations between all pair of landmarks being maintained, any new sensor information will affect all others included in the map. That means the EKF algorithm have to do an operation on every feature element included in the map, and consequently the computational cost increase quadratically with N. This scaling problem has lead to different approximations that exploit the fact that an observation of a landmark has a weak effect on the position of distant landmarks. Several researchers have developed techniques to alleviate the computational burden: the sparse extended information filters [26], the decoupled stochastic mapping [15], the compressed filter [13], the sequential map joining [25] and the constrained local submap filter [28] have improved the scalability problem of the standard EKF-based version the SLAM problem. These methods employs a single, globally referenced coordinate frame for state estimation. A different approach to solve the scaling problem is the hybrid metrical/ topological approach used in ATLAS [5]. The Atlas approach used multiple connected local maps to integrate global and local mapping, limiting the representation of errors to local regions and adopting topological methods by which local submaps can be managed to provide a global map.

With regard to the convergence of this solution, there are some controversial aspects. If we assume the noise is Gaussian and the observation and motion models can be linearized almost perfectly, Dissanayake's work [8] has demonstrated three important convergence properties: first, "the decreasing of the determinant of any submatrix of the map covariance matrix as observations are made successively"; second, "the landmark estimates become fully correlated in the limit"; third, "the covariance associated with any single landmark pose estimate reaches a lower bound determined only by the initial covariance in the vehicle location estimate at the time of the first sighting of the first landmark". However, the convergence proofs given in [8] are only for a linear case and a number of researchers have shown that EKF SLAM can produce inconsistent estimations [14,6,12,17]. The theoretical analysis and the recent results presented by Dissanayake [9] confirm that this inconsistency may occur when the robot orientation uncertainty is large.

Bailey [1] and other researchers [9] have signalled that the robot orientation uncertainty is the main cause of the inconsistency in EKF SLAM. Extensive simulation work has been done to show that the inconsistency does exist, and much of the work pointed out that linearisation is the cause of the inconsistency. Up to now, the only theoretical explanation can be found in [14], and this work only deals with the case of a robot in a stationary situation. Julier and Uhlmann [14] have shown that when the robot is stationary, the state estimate of the robot will remain unchanged if and only if the Jacobians satisfy certain conditions, and the in case of a moving robot if the Jacobians are evaluated at the *true* states then always hold. This result has been confirmed in [9] that has shown that some convergence properties hold if all the Jacobians are evaluated at the true states, but when the Jacobians are evaluated using estimated states, inconsistent states can result. The inconsistency is significant when the orientation uncertainty is large, and small if the orientation uncertainty is small.

The linearisation problem is structural and cannot be avoided. Castellanos et al. have also shown [6] that linearization errors produce inconsistency problems in the standard EKF solution for SLAM. These problems can be reduced, but cannot be eliminated in the long-term, as the basic problem is non-linear.

A second widely investigated aspect concerns the scaling properties of the stochastic map solution to the SLAM problem. The $O(n^2)$ complexity of the basic solution (where n is the number of features in the map) has been a serious bottleneck. Several researchers have developed techniques to alleviate the computational burden: the sparse extended information filters [26], the decoupled stochastic mapping [15], the compressed filter [13], the sequential map joining [25] and the constrained local submap filter [28] have improved the scalability problem of the standard EKF-based version the SLAM problem. However, the linearisation problem and the data association problem ambiguities are not completely solved.

A third point rarely considered in literature is the inherent difficulty of feature-based models to map cluttered and irregular areas, and to be used by other tasks in a mobile robot, particularly in planning and navigation activities. This necessitates the introduction of additional software modules, whose computational cost is ignored. Feature-based representations are attractive because of their compactness, but are difficult to use for path-planning purposes and are unable to model arbitrary features

A second group of SLAM algorithms is based on the occupancy grid mapping method. Grid-based techniques are computationally expensive and have a larger memory requirement than EKF-based SLAM methods. This group of solutions uses the Rao–Blackwellized particle filters to estimate the posterior probability distribution functions [19,18]. The algorithm complexity in feature map implementations is $O(M \log K)$, where M is the number of particles an K is the number of landmarks. When a grid-based map is used, the cost of evaluating the map for each potential trajectory considerably increases the computational cost of the method. Rao–Blackwellized particle filters [10] have offered an effective method of solving the data association problem for detecting loop closures. However, the main problem of this approach derives from the fact that each particle of the filter represents a different map of the environment. This requires obtaining as many maps as particles being used in the filter at each algorithm step. A second problem, common in Bayesian filters, is the difficulty of estimating an adequate number of particles. If the number is low, the algorithm cannot estimate the posterior properly, and if the pose number is large the computational cost increases rapidly. This difficulty is usually solved through trial and error.

A third group of SLAM algorithms derives from the Iterative Closest Point (ICP) method [4] and are referred to scan matching methods. The goal of scan matching is to find the relative pose between two range scans obtained at different (typically consecutive) positions, at which the scans were taken. The scan matching approach matches points to points and was introduced in the SLAM field by Lu and Milios [16] applied to laser scan matching. This technique is essentially a local pose estimation method based on an implicit environment model (the environment model is a vector of points). This method is essentially local and this characteristic limits its use to improve the odometry data (a data realignment or tracking of robot poses) previous to a global modelling of the environment done with other method.

Not many global optimization methods have been applied to the SLAM problem. Duckett [11] used a genetic algorithm approach to solve the grid SLAM problem. The search was carried out in the space of possible robot trajectories, and candidate solutions were coded as a vector of correction factors. The evaluation function used to determine the quality of the candidate solutions was a heuristic function that considered two aspects: the map consistency of the sensory information contained in the gridmap (the degree of disagreement between the sensor readings is calculated as by taking the minimum of the occupancy and empty values for all cells in the map), and the map compactness (the idea was to reward the GA for producing smaller, more compact maps, this was done by fitting a bounding box to the map that indicates the total area covered by cells with occupancy index > 0). This approach was extremely slow, because each element of the set of possible solutions represented a robot path, and a global map needed to be evaluated for each possible solution. The accuracy of the map obtained was poorer than other methods, and it did not offer a solution to the loop closing problem.

In this article we present an alternative solution to the SLAM problem that avoids the EKF linearisation problems, and the inability of feature-based maps to obtain a complete environment model. It also avoids the computational cost problem in grid-based mapping with Rao–Blackwellized approaches. The technique proposed uses a stochastic search optimization approach, based on a solution perturbation method called differential evolution to obtain the best pose estimate. This technique is used at a local level to re-localize the robot pose and at global level to solve the data association problem existent at loop closing.

The advantage of our approach is twofold. First, the solution proposed builds only one certainty grip map, thus avoiding unnecessary operations. This map is built at loop closing time, when the data association has been solved, and the residual error can be eliminated. Second, the computational efficiency of the method reduces the computational cost.

3. Localization problem formulation and solution

The SLAM solution we propose is based on the application at local or global scale of a differential evolution search method to obtain optimal solutions to the localization problem. In this section, the localization problem is formulated as an optimization problem, and the stochastic search method called Evolutionary Localization Filter (ELF) is developed.

From a Bayesian point of view, the localization problem can be formulated as a probability density estimation problem where the robot seeks to estimate a posterior distribution over the space of its poses conditioned on the available data. The sensor data can be divided into two groups of data $Y_t \equiv \{z_{0:t}, u_{1:t}\}$ where $z_{0:t} = \{z_0, \ldots, z_t\}$ contains the perception sensor measurements and $u_{1:t} = \{u_1, \ldots, u_t\}$ contains odometric information. To estimate the posterior distribution $p(x_t|Y_t)$, probabilistic approaches resort to the *Markov assumption*, which states that future states only depend on the knowledge of the current state, and not how the robot got there, i.e. they are independent of past states. The recursive determination of the posterior probability density can be computed in two steps:

- *Measurement update*. Applying the Bayes's rule to the last element of the measurement vector Y_t and assuming that the observation z_t is conditionally independent of the previous measurements given the state x_t, yields

$$p(x_t|Y_t) = \frac{p(z_t|x_t, Y_{t-1})p(x_t|Y_{t-1})}{p(z_t|Y_{t-1})} = \frac{p(z_t|x_t)p(x_t|Y_{t-1})}{p(z_t|Y_{t-1})} \quad (1)$$

where the denominator of (1) is obtained by marginalization in (2).

$$p(z_t|Y_{t-1}) = \int_{\Re^n} p(z_t|x_t)p(x_t|Y_{t-1})\mathrm{d}x_t. \quad (2)$$

- *Prediction*. The effect of a time step over the state given the observations up to time t is obtained by observing that

$$p(x_{t+1}|Y_t) = \int_{\Re^n} p(x_{t+1}|x_t, u_t, Y_t)p(x_t|Y_t)\mathrm{d}x_t = \int_{\Re^n} p(x_{t+1}|x_t, u_t)p(x_t|Y_t)\mathrm{d}x_t \quad (3)$$

where the assumption that the process x_t is Markovian, and then x_{t+1} is independent of Y_t has been considered.

Eqs. (1)–(3) are the solution to a recursive Bayesian estimation problem. These equations integrate statistical and sensorial information about the system. In general, the multidimensional integrals in expressions (2) and (3) have no explicit analytical solutions for non-linear and non-Gaussian models. The latter is a general statistical solution to the estimation problem but is difficult to obtain, and to manage for general non-linear and non-Gaussian problems. Each candidate parameter value in $\Re^n$ yields a value of $p(x|y)$, reflecting the posterior probability of the robot pose given the data up to time t. This posterior needs to be weighted according to a given criterion to determine an estimate $\hat{x}$ of the true pose value. Two common choices of cost function are the mean-square error and the *maximum a posteriori* estimators. The minimum mean-square estimate is defined by

$$\hat{x}_{MS} = \arg\min_{x_t^*} \int_{\Re^n} (x_t - x_t^*)^T (x_t - x_t^*) p(x_t|Y_t)\mathrm{d}x_t. \quad (4)$$

It is well known that the optimal estimate in the mean-square sense is the conditional mean. This estimate is also referred to as the *conditional mean estimate*. Since the posterior probability distribution is multi-modal in the global localization problem, this estimate is inconvenient. Another common choice is the *maximum a posteriori* estimator,

$$\hat{x}^{MAP} = \arg\max_{x_t} p(x_t|Y_t). \quad (5)$$

The solutions derived from the Bayesian approach concentrate on maintaining the statistical coherency in the posterior probability distribution according to all received information up to a given moment. The localization algorithm concentrates on obtaining the *maximum a posteriori* estimator. This approach is less dependent on statistical assumptions, has a simpler implementation, is robust from a statistical point of view and has a computational cost lower than Bayesian methods.

3.1. Localization as a MAP optimization problem

The localization problem is basically an optimization problem, where the robot seeks to estimate the pose which maximizes the *a posteriori* probability density.

$$\begin{aligned}\hat{x}_t^{MAP} &= \arg\max_x p(x_t|Y_t)\\ &= \arg\max_x p(z_t|x_t, u_{t-1}, Y_{t-1})p(x_t|x_{t-1}, u_{t-1}, Y_{t-1})\\ &= \arg\max_x p(z_t|x_t)p(x_t|x_{t-1}, u_{t-1})p(x_{t-1}|Y_{t-1})\\ &= \arg\max_x \prod_{i=1}^{t} p(z_i|x_i)\prod_{i=1}^{t} p(x_i|x_{i-1}, u_{t-1})p(x_0). \end{aligned} \quad (6)$$

The *maximum a priori* (MAP) estimate expression can be easily stated as an optimization problem subject to constraints (the motion and observation models of the robot). The MAP estimate is the solution $\hat{x}_t$ to the following problem,

$$\hat{x}_t = \max_{x_t} \prod_{i=1}^{t} p_e(z_i|x_i)\prod_{i=1}^{t} p_v(x_i|x_{i-1}, u_{t-1})p(x_0)$$

subject to:

$$\begin{aligned} x_{t+1} &= f(x_t, u_t) + v_t, \quad t = 0, 1, \ldots\\ z_t &= h(x_t) + e_t, \quad t = 0, 1, \ldots \end{aligned} \quad (7)$$

where p_e express the probability density function for the observation noise e, and p_v indicates the probability density function for the motion noise v. We will use the notation $f_0(x)$ to refer to the objective function to maximize. The problem of finding an x that maximizes $f_0(x)$ among all x that satisfy the conditions $x_{t+1} = f(x_t, u_t) + v_t$ and $z_t = h(x_t) + e_t$ is limited to finding the optimal value between the set of all feasible solutions.

In general, the calculation of estimates for this optimization problem has no explicit analytical solution for non-linear and non-Gaussian models. The difficulties are due to: non-linearities in motion and perception models, environment symmetries and sensor limitations (observational symmetries). The problem is highly simplified when the initial probability distribution is Gaussian, because the problem becomes uni-modal and then it is possible to obtain, even analytically, an estimate (since the problem can be converted into a quadratic minimization problem if non linear motion and observation models can be approximated by a linear Taylor series expansion about the current estimate $\hat{x}_t$). This situation lead to the well known Extended Kalman Filter solution of the pose tracking problem.

3.2. Recursive formulation of the optimization problem

The MAP estimation, formulated as an optimization problem subject to constraints in Eq. (7) is not practical from a computational point of view. To implement the global localization algorithm in a robot, a recursive formulation is required. If we observe the objective function $f_0(x_t)$, it can be reformulated in a more convenient form by taking logarithms and then expressed recursively in terms of $f_0(x_{t-1})$ in the following way:

$$\begin{aligned} f_0(x_t) &= \sum_{i=1}^{t} \log p_e(z_i|x_i) + \sum_{i=1}^{t} \log p_v(x_i|x_{i-1}, u_{t-1}) + \log p(x_0)\\ &= \log p_e(z_t|x_t) + \log p_v(x_t|x_{t-1}, u_{t-1}) + f_0(x_{t-1}). \end{aligned} \quad (8)$$

If we are able to solve the optimization problem at time $t-1$, we have a set of m possible solutions $\{x_{t-1}\}_{1:m}$ which satisfy the optimization problem up to time $t-1$. The MAP optimization problem can be reformulated as

$$\hat{x}_t = \max_{x_t} \log p_e(z_t|x_t) + \log p_v(x_t|x_{t-1}, u_{t-1}). \quad (9)$$

subject to conditions in Eq. (7) and starting with $\{x_{t-1}\}_{1:m}$ as the previous set of feasible solutions. Then, by perturbing and searching new solutions to (9), we obtain a recursive version of the MAP estimate. In the following section an algorithm is proposed to obtain the evolutionary MAP estimate for the global localization problem according to the ideas introduced in this section.

4. Evolutive localization filter algorithm

The algorithm proposed to implement the evolutive localization filter is based on the Differential Evolution (DE) method proposed by Storn and Price [24] for global optimization problems over continuous spaces, and the Evolutive Localization Filter (ELF) proposed by Moreno [20] to solve the global localization problem. The Evolutive Localization Filter uses a parallel direct stochastic search method which utilizes n dimensional parameter vectors $x_i^k = (x_{i,1}^k, \ldots, x_{i,n}^k)^T$ to point each candidate solution i to the optimization problem at iteration k for a given time step t. This method utilizes N parameter vectors $\{x_i^k; i = 1, \ldots, N\}$ as a sub-optimal feasible solutions set (population) for each generation t of the optimization process. Depending on the kind of localization problem we are trying to solve, the initialization changes. When no information about the initial pose is available, the initial population is chosen randomly to cover the entire parameter space uniformly (the area where the solution is located). If the initial pose is known, the initial population of possible solutions is obtained by distributing the poses according to a Gaussian distribution around the initial pose.

4.1. Fitness function

According to the recursive optimization problem under consideration, the natural choice for a fitness function is the objective function:

$$f_0(x^t) = \log p_e(z_t|x_t) + \log p_v(x_t|x_{t-1}, u_{t-1}). \quad (10)$$

This expression contains the state transition probability density $p_v(x_t|x_{t-1}, u_{t-1})$ and the observation probability density $p_e(z_t|x_t)$, derived from the robot motion model and the observation model. The observation probability $p_e(z_t|x_t)$ can be calculated by predicting the observation value of the noise-free sensor assuming the robot pose estimate is $\hat{x}_t$, the sensor relative angle with respect to the robot axis is α_i, and a given environment model m, that in our case is estimated. Let $\hat{z}_{t,i} = h(\hat{x}_t, m, \alpha_i)$ denote this ideal predicted measurement. For practical purposes, $\hat{z}_{t,i} = h(\hat{x}_t, \hat{m}, \alpha_i)$ is computed using a ray tracing method. Assuming the measurement error $e_{t,i}$ is Gaussian, centered at $h(\hat{x}_t, \alpha_i)$ and with a σ_e standard deviation (that is $e_{t,i} \approx N(h(\hat{x}_t, \hat{m}, \alpha_i), \sigma_e)$), then the probability of observing $z_{t,i}$ with sensor i can be expressed as,

$$p_e(z_{t,i}|\hat{x}_t) = \frac{1}{(2\pi\sigma_e^2)^{1/2}} e^{-1/2\frac{(z_{t,i}-\hat{z}_{t,i})^2}{\sigma_e^2}}. \quad (11)$$

And, assuming conditional independence between the individual measurements, the individual sensor beam probabilities are integrated into a single probability value:

$$p_e(z_t|\hat{x}_t) = \prod_{i=0}^{N_s} p(z_{t,i}|\hat{x}_t) = \prod_{i=0}^{N_s} \frac{1}{(2\pi\sigma_e^2)^{1/2}} \mathrm{e}^{-1/2\frac{(z_{t,i}-\hat{z}_{t,i})^2}{\sigma_e^2}}, \tag{12}$$

where N_s is the number of sensor observations.

The second probability required to calculate the objective function, $p_v(x_t|x_{t-1}, u_{t-1})$, can be obtained in two steps: the prediction of the noise free value of the robot pose $\hat{x}_t = f(\hat{x}_{t-1}, u_{t-1})$, where the robot pose estimate is $\hat{x}_t$ and the motion command at t is u_t. Assuming the motion error is Gaussian with covariance matrix P (that is $v \approx N(f(\hat{x}_{t-1}, u_{t-1})), P)$, then the $p_v(x_i|x_{i-1}, u_{t-1})$ probability can be expressed as:

$$p_v(x_i|x_{i-1}, u_{t-1}) = \frac{1}{\sqrt{|P|(2\pi)^n}} \mathrm{e}^{-1/2(x_i-\hat{x}_i)P^{-1}(x_i-\hat{x}_i)^T}. \tag{13}$$

Replacing the expressions of p_v and p_e into the objective function (10) to optimize at iteration t,

$$\begin{aligned} f_0(x_t) &= \log \prod_{i=0}^{N_s} (2\pi\sigma_e^2)^{-1/2} \mathrm{e}^{-\frac{(z_{t,i}-\hat{z}_{t,i})^2}{2\sigma_e^2}} \\ &\quad + \log(|P|(2\pi)^n)^{-1/2} \mathrm{e}^{-\frac{1}{2}(x_i-\hat{x}_i)P^{-1}(x_i-\hat{x}_i)^T} \\ &= \sum_{i=0}^{N_s} \log(2\pi\sigma_e^2)^{-1/2} - \sum_{i=0}^{N_s} \frac{(z_{t,i}-\hat{z}_{t,i})^2}{2\sigma_e^2} \\ &\quad + \log[(|P|(2\pi)^n)^{-1/2}] - \frac{1}{2}(x_i-\hat{x}_i)P^{-1}(x_i-\hat{x}_i)^T \end{aligned} \tag{14}$$

which can be reduced to minimize the following function

$$\tilde{f}_0(x_t) = \sum_{i=0}^{N_s} \frac{(z_{t,i}-\hat{z}_{t,i})^2}{2\sigma_e^2} + \frac{1}{2}(x_i-\hat{x}_i)P^{-1}(x_i-\hat{x}_i)^T. \tag{15}$$

The differential evolutive filter will minimize iteratively the fitness function (15).

4.2. Evolutive localization filter algorithm

The Evolutive Localization Filter (ELF) uses n dimensional parameter vectors $x_i^k = (x_{i,1}^k, \ldots, x_{i,n}^k)^T$ to represent each candidate solution i to the optimization problem at iteration k for a given time step t. The filter will generate a new set of parameter vectors by perturbing an existing vector through the addition of one or more weighted difference vectors to it (see Fig. 1). If the resulting vector yields a lower objective function value than the original unperturbed vector, the newly generated vector replaces the vector to which it was compared; otherwise, the old vector is retained. The perturbation scheme generates a variation v according to the following expression,

$$v = x_i^k + F(x_{r_2}^k - x_{r_3}^k) \tag{16}$$

where x_i^k is the parameter vector to be perturbed at iteration k, $x_{r_2}^k$ and $x_{r_3}^k$ are parameter vectors chosen randomly from the population and r_2 and r_3 are different from running index i. F is a real and constant factor which controls the amplification of the differential variation $(x_{r_2}^k - x_{r_3}^k)$.

To increase the diversity of the new generation of parameter vectors, a crossover mechanism is introduced. The vector denoted by $u_i^k = (u_{i,1}^k, u_{i,2}^k, \ldots, u_{i,n}^k)^T$ is the new parameter vector with

$$u_{i,j}^k = \begin{cases} v_{i,j}^k & \text{if } p_{i,j}^k < \delta \\ x_{i,j}^k & \text{otherwise} \end{cases}$$

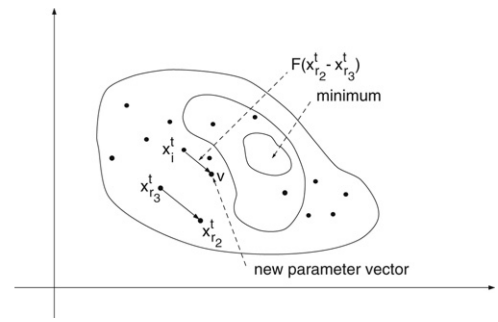

Fig. 1. New population member generation.

where $p_{i,j}^k$ is a randomly chosen value from the interval [0, 1] for each parameter j of the population member i at step k and δ is the crossover probability and constitutes the crossover control variable. The random values $p_{i,j}^k$ are made anew for each trial vector i.

To decide whether or not vector u_i^k should become a member of generation $i+1$, the new vector is compared to x_i^k. If vector u_i^k yields a better value for the objective fitness function than x_i^k, x_i^k is replaced by u_i^k for the new generation; otherwise, the old value x_i^k is retained for the new generation.

In accordance with the previous ideas, the basic evolutive localization filter algorithm consists of the following steps:

- *Step 1: Initialization*.
  The initial set of solutions is calculated, and the fitness value associated with each of the points in the state space is evaluated. If no information about initial position is available, the initial set of pose solutions is obtained by drawing the robot poses according to a uniform probability distribution over the state space. The initial robot pose estimate is fixed at an initial value.
- *Step 2: Evolutive search*.
  (a) For each element of the set of robot pose solutions and according to the map, the expected sensor observations are obtained $\hat{z}_{t,i} = h(\hat{x}_t, \alpha_i, m)$, $i = 1, \ldots, N_s$. The expected observations, the sensor observations, the robot pose estimate and the robot pose element are used to evaluate the loss function for each robot pose element in the set of solutions.
  (b) A new generation of perturbed pose solutions are generated according to the perturbation method described in the previous subsection. For each perturbed solution, the expected observations are calculated and the loss function evaluated. If the perturbed solution results in a better loss function, this perturbed solution is selected for the following iteration, otherwise the original is maintained.
  (c) The crossover operator is applied to the resultant population (solutions set).
  (d) The robot pose element of the set with lower value of the loss function is marked as best robot pose estimate. Go to step 2b a given number of iterations.
- *Step 3: Updating*. The best robot pose element of the population is used as the updated state estimate and then used in the state transition model to predict the new state according to the odometry information. Next, the displacement is evaluated and the whole population is moved according to this displacement. Then go to step 2.

5. Differential evolution approach to SLAM problem

In the simultaneous localization and mapping problem case, the map needs to be estimated concurrently with the pose to generate the expected measurements. In Fig. 2, the data flow in SLAM is

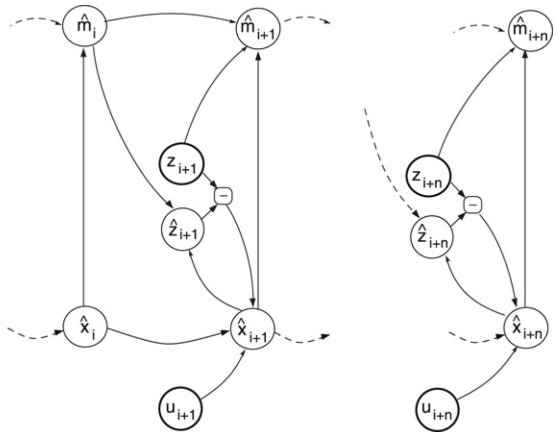

Fig. 2. SLAM data flow.

shown. Note how the SLAM problem solution depends on the data association problem at local and global level.

The most extended Bayesian formulation for the SLAM problem is to estimate the posterior probability of the trajectories, the map associated to it given the observations $z_{1:t}$ and the odometry measurements $u_{0:t}$:

$$p(x_{1:t}, m_1^t|u_{0:t}, z_{1:t}) = p(m_1^t|x_{0:t}, z_{1:t})p(x_{1:t}|z_{1:t}, u_{0:t}). \tag{17}$$

In this expression (17), the posterior over maps $p(m|x_{0:t}, z_{1:t})$ can be easily computed by means of a mapping process when $x_{0:t}$ and $z_{1:t}$ are known. Obtaining the posterior over the trajectories $p(x_{1:t}|z_{1:t}, u_{0:t})$ is much more complex. This approach has been shown to be effective but is computationally very expensive as each pose sequence has an associated map which needs to be calculated to make sensor predictions. At each iteration step the probability associated to augmented states, including a candidate sequence and the map obtained by fusing the sensor data with this sequence, is calculated. The map associated to a given sequence (an occupancy grid map in our case) is obtained by fusing the observed data up to time t, $z_{1:t}$ according to the estimated robot's pose, $\hat{x}_{0:t}$, as $m_0^t[\hat{x}_{0:t}, z_{1:t}]$, where $p(m_{ij})$ expresses the occupancy probability of the cell i, j in the map m according to the data received up to time t. For notation convenience we express it as $m_0^t[\hat{x}_{0:t}, z_{1:t}] = \hat{m}_0^t$.

The method proposed in this paper is quite different in that it exploits the ELF algorithm capability to operate at local and global level. The algorithm uses a two step approach. The first step exploits the local data coherency idea to partially eliminate the model inconsistency problem by using *local models* to re-localize the robot. Even if the global position is not correct these local models maintain the local data consistency. This local data coherency is used to estimate the robot's pose precisely but cannot completely eliminate the pose error, and the residual remaining error is accumulated over the motion. This idea is in some sense similar to scan matching but instead of using the sensor data directly a local model is built and the robot is re-localized. This avoids the lack of robustness present in most scan matching approaches when the pose difference between scans is significant, particularly in orientation.

The second step of the algorithm exploits a different concept to avoid the global inconsistency problem. The key idea is to delay the sensor data integration in the *global map* until the residual error is eliminated. The residual error can be eliminated when a loop is detected and the global data association is done. Both ideas are combined in the solution proposed in this paper to obtain an accurate global model based on the application of the Evolutive Localization Filter at local and global level.

5.1. Local data association

One way to eliminate the fast degradation of the estimation process, and consequently of the mapping process, is to re-localize the robot by using a local map which integrates the last n observations perceived by the robot. This process constitutes a local data association and the estimated pose is referred to as $\hat{x}_t^L$.

$$\begin{aligned}\hat{x}_t^L &= \arg\max_x p(x_t|Y_t, \hat{m}_{t-n}^{t-1}) \\ &= \arg\max_x p(z_t|x_t, \hat{m}_{t-n}^{t-1})p(x_t|x_{t-1}, u_{t-1})p(x_{t-1}|Y^t, \hat{m}_{t-n}^{t-1}) \\ &\approx \arg\max_x p(z_t|x_t, \hat{m}_{t-n}^{t-1})p(x_t|x_{t-1}, u_{t-1}). \end{aligned} \tag{18}$$

The last term of expression (18), $p(x_{t-1}|Y^t, \hat{m}_{t-n}^{t-1})$ is not considered in this step, because its only objective is to reduce the residual error to a tractable level. This solution allows the robot to be re-localized efficiently and avoids pose estimate degradation. This re-localization is equivalent to a maximum likelihood estimate, but is obtained with a stochastic search method, avoiding the noise problems of gradient descend methods. An equivalent method which is gradient based is the iterative closest point proposed by [4] used in different scan matching approaches [16]. Other iterative point matching methods use related concepts [29] and have similar problems.

In Fig. 3 the re-alignment effect on the robot's poses for the environment used in test 1 (Section 6 in this paper) can be observed. The data has been obtained by recording odometry and sensor information using a laser scanner (without any re-localization) and it has been aligned using the ELF filter locally and a local map of the environment where the last 20 laser sensor scans are integrated. Obviously, this local association does not eliminate the residual error accumulation which tends to increase with distance travelled [2]. This problem is addressed in the following subsection.

5.2. Global data association

According to the previous notation, we can also estimate the robot's pose by using the global map, up to a given time, to obtain the *maximum a posteriori* estimate over the full pose sequence. We refer to this as $\hat{x}_t^G$.

$$\begin{aligned}\hat{x}_t^G &= \arg\max_x p(x_{0:t}|Y_t, \hat{m}_0^{k-1}) \\ &= \arg\max_x p(z_t|x_{0:t}, \hat{m}_0^{t-1})p(x_t|x_{t-1}, u_t)p(x_{0:t-1}|Y_{t-1}, \hat{m}_0^{k-1}). \end{aligned} \tag{19}$$

The objective of the global data association is to estimate the residual error accumulated in a loop. This global data association does not require a complete global map, in fact, only the area of the global map which is currently being revisited is required in order to localize the residual error accumulated in a given loop. This characteristic is used to integrate sensor data in a lazy mode into the global map. Fig. 4 shows the robot pose at the end of a detected cycle (test example number 1). The estimated robot trajectory after the local data association is shown in magenta, and the last position of the loop is marked with an additional red cross overlapped, and the observed sensor measures taken at that pose are represented as red points in the figure. Notice the misalignment between the information contained in the global map done up to that step. The observed misalignment is originated in the accumulation of small residual errors remaining after each step of the local data association. Using this position as a center, a global localization is started to re-estimate the best pose according to the global map. The new estimated pose shows the total residual error accumulated in this loop. The resulting estimated pose is

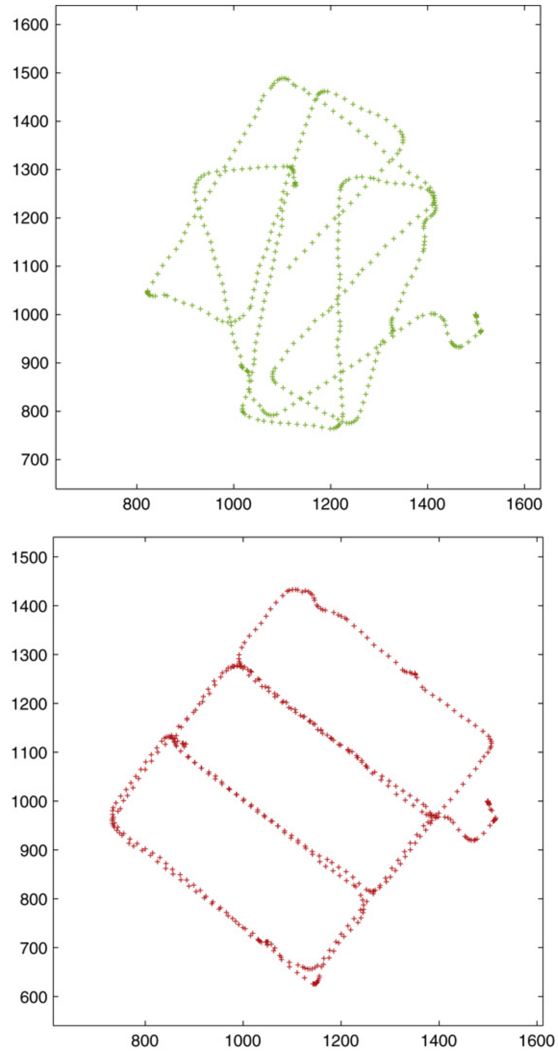

Fig. 3. Initial odometry and re-aligned poses for test environment 1.

shown in Fig. 4 as a blue cross, and the observed sensor measures corresponding to that pose are shown as blue points. It can be observed that the matching between the observed data (at the re-estimated pose) and the global map is much better than the obtained in the local data association. The global map used to estimate the residual error accumulated in the pose estimate at the end of the first loop does not integrate data observations corresponding to the robot poses included in the detected loop (Fig. 4). The integration of these observations into the global map is delayed until the residual error is corrected.

5.3. Residual error correction

Assuming the modelled global map up to a given time t is $\hat{m}_0^{k-1}$, which integrates the sensor measurements up to the start of the environment loop $k-1$, to detect if x_t corresponds to a possible closing of a loop, a combined test based on the distance between x_t and points in the pose sequence, and the overlapping area between sensor scanned area and the modelled global map is used. If the test is positive, a global data association is started. When the solutions sets converge and the fitness function value of the best estimate is below a given threshold, the global estimate is used to correct the residual error accumulated in the detected loop pose sequence (from poses $k-1$ to t). The error $\Delta x_t = x_t^G - x_t^L$ is proportionally attributed to poses in the loop, and a new re-localization is done for

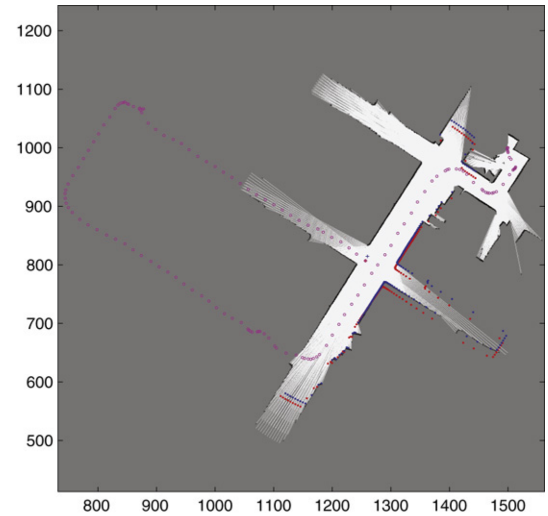

Fig. 4. Global pose estimate at the end of a detected loop. The robot trajectory is shown in magenta, the last position of the loop is marked with a red cross and red points represent sensor measures. The blue cross represents the estimated pose after the global localization and the observed sensor measures corresponding to that pose are shown as blue points.

each robot pose using the global map up to that moment instead of the local environment (this time with a very narrow optimization area).

Using these new estimates, $\hat{x}_{0:t}^{MAP}$, to integrate observations $z_{0:t}$ in the map, $\hat{m}_0^t$, it is possible to maintain the robot properly localized and to maintain a globally coherent representation of the environment observed along the robot motion up to time t. In that situation, the estimate obtained through the previous process is the following

$$\begin{aligned}\hat{x}_{0:t}^{MAP} &= \arg\max_x p(x_{0:t}|Y_t, \hat{m}_0^{t-k}) \\ &= \arg\max_x p(z_t|x_{0:t}, \hat{m}_0^{t-k})p(x_t|x_{t-1}, u_t)p(x_{0:t-1}|Y_{t-1}, \hat{m}_0^t).\end{aligned} \tag{20}$$

A proportional error attribution has been adopted, because after the initial re-localization process the residual remaining error cannot be considered Gaussian.

5.4. ELF based SLAM algorithm

According to previously explained ideas, the ELF grid map based SLAM algorithm operates as follows: using only the odometry and the sensor data obtained directly from a laser scanner, a first re-alignment step is done. In this step a re-localization of the robot pose is done by using the ELF localization algorithm operating with a local map which includes the last 20 robot poses and the laser data associated to them.

Once the robot poses have been realigned, a second step is executed to obtain the global map and the globally corrected pose sequence. The global mapping algorithm steps are the following:

(1) Current pose index is initialized to 1, and the first scan is integrated into the global map.
(2) The poses of the following scans in the sequence are compared with the current pose. If the geometrical distance between them is below a given threshold (that has been established in 3 m) or if there is an overlap between current scan area and the global map, then a potential cycle exists. If there is no cycle, the next pose in the sequence is explored until a cycle is detected.

(3) If a potential cycle has been detected, a global localization is started. The size of the explored area where the pose is re-localised depends on the length of the detected cycle. The number of elements in the solution set is related to the area to be explored.
   - If the fitness value (loss function) of the new pose estimate is below an acceptance threshold, the new pose is accepted. The detected error between the realigned pose and the re-estimated pose at the end of the cycle is the accumulated residual error remaining along the cycle after the realignment process.
   - The accumulated residual error is distributed proportionally between all poses along the cycle to correct the accumulated error at the end of the detected cycle. The error is distributed proportionally according to the magnitude of the displacement and turn at each motion.
   - For all poses in the loop:
     – The ELF algorithm is started, limiting its search area to a small area around the corrected pose. This time the global map is used. This readjusts the estimate around the corrected pose and the updated global map.
     – With the corrected pose the new scan data of the detected cycle are integrated into the global map.

(4) If no cycle is detected but the area has been previously modelled, a re-localisation is done, the error corrected and the scan integrated in the global existent map.
(5) Increment current pose and go to step 2.

In the following section, the results obtained with the proposed algorithm are shown.

6. Experimental results

To demonstrate the algorithm, two different test environments have been used. The data for the first test environment have been obtained from the Robotics Data Set Repository (Radish) [3] and they are part of the Intel Jones Farms Campus (Oregon). We thank M. Batalin for providing this data. This environment is an excellent test to show the cycle resolution capability of the mapping algorithm due to the presence of different cycles in the environment.

Fig. 4 shows the pose at the end of the first detected cycle, the new pose estimate in the global data association, together with the laser measures at original and re-estimated poses and the measures prediction at re-estimated pose. After the accumulated error is distributed between all poses included in the cycle and re-estimated around this new position according to the global map, the global data integration process in the global map is done, and the result is shown in Fig. 5, together with the next scan.

Fig. 6 shows a different cycle situation. In this situation, the robot trajectory does not pass close to previous trajectory poses, but there is an overlapping between the perceived areas. In this case, a test indicates the perceptual overlapping and the global localization algorithm is started to obtain the best global estimate for the new pose. This new global estimate allows calculation and correction of the residual alignment error up to that pose.

Fig. 7 shows the state of the global map after integrating 451 scans. Four cycles have been detected in addition to the revisiting of previously crossed areas. Note that the algorithm can solve satisfactorily the loops in the mapping of this first test.

A second test has been done with data obtained from experiments developed at our University laboratories, offices and corridors. The test site is around 60 meters length. The data scans have been obtained approximately each 80 cm using a stop and go method in order to measure the true robot pose. In this test example there are no cycles present, but there is a high degree of symmetry (a long corridor with office doors regularly distributed).

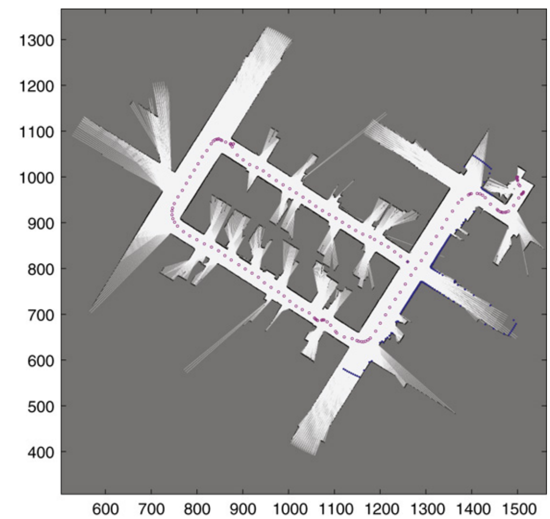

Fig. 5. Global map after error attribution and fusion at the end of the first detected loop.

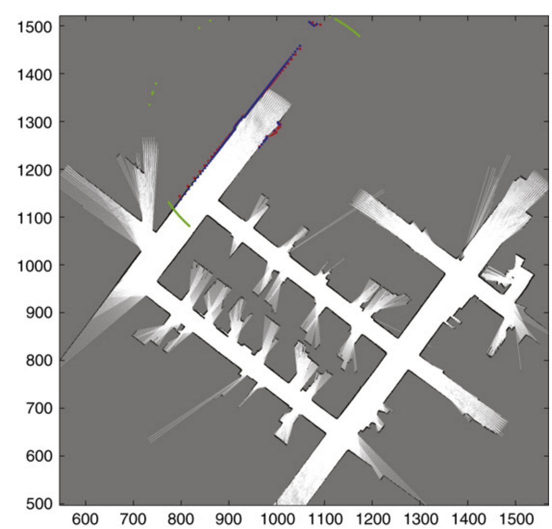

Fig. 6. New pose estimate at the end of the second detected loop.

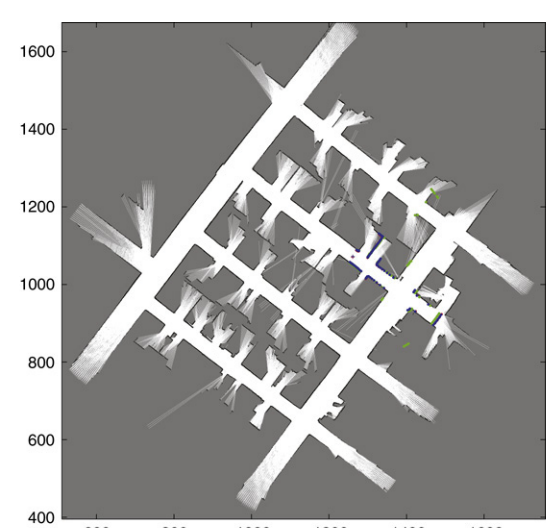

Fig. 7. Global map after 451 scans fusion in environment test 1.

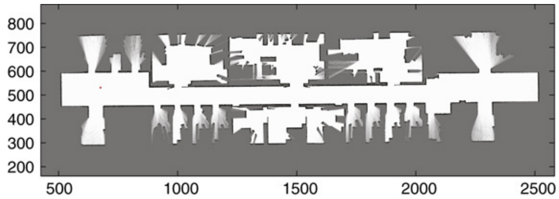

Fig. 8. Global map for the environment test 2.

Fig. 9. Real map of the environment test 2.

In the Fig. 8, the result of the complete SLAM process in the same environment of test 2 appears. As it can be noted comparing with the real map that is in the Fig. 9, the very good performance of the algorithm in the mapping process is demonstrated. In this last test the process has been carried out using a tele-operated running mobile robot.

An interesting aspect of the algorithm is the low number of pose solutions required in the population to achieve satisfactory results. For the re-localisation task, a population of 30 members with 20 iteration steps is enough, while for the global localisation task the population number used is dynamically adapted depending on the loop travelled distance. A more detailed consideration of the accuracy and convergence properties of the localization algorithm can be found in [20].

The complexity of the ELF algorithm is $O(N \times M)$, where N is the population size, M is the iteration number, N parameter is incremented linearly with the radius of a ball around the robot's pose estimate where the solution to the data association is searched, and the iteration number M is incremented linearly with the number of poses included in the detected loop. In spite of this complexity appearing to be substantial, in practical terms it is very moderate. This is because only in the case of a loop detection the global data association is started, and in case of local correction, the population set is very small. To illustrate this computational cost, using a cell size of 3 cm and for the environment test1 the time required for the first global localisation at the first detected loop (105 poses in the loop) is 15.8 s, for the second detected loop (30 poses in the loop) it is 5.6 s, and for local optimisations it is 0.47 s on average. The algorithm is not fully optimised: for instance, sines and cosines are calculated using conventional functions instead of using a tabulated version to decrease the computational cost of the algorithm.

The initial realignment left an estimated residual error lower than $0.25°$ in orientation and lower than 1 cm in position (on average). This error is eliminated in the global data association process.

7. Conclusion

The differential evolution-based solution to the grid-based SLAM problem presented in this article introduces a new possibility to accurately solve the SLAM problem. At a local scale, the ELF localisation algorithm provides a fast and accurate maximum likelihood estimate with results equivalent to other re-localisation methods like scan matching approaches. And at a global scale, the algorithm only incurs substantial cost at cycle closing detection time and the cost increases linearly with the number of poses included in the detected loop, remaining constant for the re-localisation and integration of the scans in the global map which is lower than one second for the re-location and the mapping of the scan data into the map.

The Markov chain nature of evolutionary algorithms is exploited by introducing into the loss function the sensor error innovation together with motion error innovation. In the ELF method used here each individual in the evolutive algorithm represents a possible solution to the localisation problem and the value of the loss function represents the error in explaining the perceptual and motion data. The search for this solution is done stochastically, employing an evolutive search technique. At a global scale the proposed approach is able to detect and estimate environment loops to eliminate the residual error. This allows the algorithm to accurately map large environment maps. The algorithm has been tested with data acquired with different robots equipped with laser range scanners. Tests performed with our algorithm have demonstrated the algorithm's robustness and accuracy in generating maps.

The evolutive optimisation technique constitutes a probabilistic search method that avoids derivatives. The use of derivatives present two types of problems:

- It causes strong numerical oscillations when noise to signal ratio is high.
- It requires differentiable functions as otherwise the derivatives can be discontinuous or not even existent.

Due to the low number of solutions required in the algorithm, it is possible to use the algorithm to model large environments at low computational cost.

The advantages over other SLAM algorithms are:

(1) Compared with other techniques used at a local level to relocalize the robot, it is very robust because it does not use a gradient-based optimisation like scan-matching techniques based on the iterative closest point method or classic Extended Kalman Filters solutions. The robustness of the algorithm proposed derives from its stochastic search nature.
(2) The method can accommodate arbitrary non-linear system dynamics, sensor characteristics and non-Gaussian noise, avoiding unnecessary linearisations. By introducing in the fitness function the sensor innovation together with the motion innovation, and due to the Markov Chain behaviour of the evolutive search algorithm, the set of particles evolves gradually along the most probable environment areas. This allows the algorithm to solve large loops where one scan is insufficient to eliminate the ambiguity (a similar problem occurs in the global localisation problem).
(3) Since the set of solutions does not try to approximate posterior density distributions, it does not require any assumptions about the shape of the posterior density, unlike parametric approaches.
(4) The evolutive filter focuses its computational resources on the most relevant areas by addressing the set of solutions to the most interesting areas according to the fitness function obtained.
(5) The size of the minimum solution set required to guarantee the convergence of the evolutive filter to the true solution is low, see [21] for details of the ELF performance.
(6) The algorithm is easy to implement. The ELF localisation algorithm is used at a local or global level. Moreover, the low computational cost allows online operation even in relatively large areas.
(7) Due to the stochastic nature of the algorithm search of the best robot pose estimate, the algorithm is able to cope with a high level of sensor noise with low degradation of the estimation results [21].
(8) Delayed mapping, in case of loops, eliminates the necessity of maintaining multiple maps or re-mapping each time a significant global error is detected.

Acknowledgements

The authors gratefully acknowledge the funds provided by the Spanish Government through the MCYT project DPI2004-0594.

References

[1] T. Bailey, J. Nieto, J. Guivant, M. Stevens, E. Nebot, Consistency of EKF–SLAM algorithm, in: Proc. of the IEEE Int. Conf. on Intelligent Robot Systems IROS-2006, Beijing, China, 2006, pp. 3562–3568.
[2] Y. Bar-Shalom, T.E. Fortmann, Tracking and Data Association, Academic Press, 1988.
[3] M. Batalin, The Robotics Data Set Repository (Radish). http://radish.sourceforge.net.
[4] P.J. Besl, N.D. McKay, A method for registration of 3-d shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (1992) 239–256.
[5] M. Bosse, P. Newman, J. Leonard, S. Teller, Similtaneous localization and map building in large-scale cyclic environments using the atlas framework, Internationl Journal of Robotics Research 23 (12) (December,2004) 1113–1139.
[6] J.A. Castellanos, J. Neira, J.D. Tardos, Limits to the consistency of EKF-based SLAM, in: Proc. of the 5th IFAC/EURON Symposium on Intelligent and Autonomous Vehicles IAV-2004, Lisboa, Portugal, 2004.
[7] F. Dellaert, D. Fox, W. Burgard, S. Thrun, Monte Carlo localization for mobile robots, in: Proceedings of the 1999 International Conference on Robotics and Automation, 1999, pp. 1322–1328.
[8] M.W.M.G. Dissanayake, P. Newman, H.F. Durrant-Whyte, S. Clark, M. Csorba, A solution to the simultaneous localization and map building (SLAM) problem, IEEE Transactions on Robotics and Automation 17 (3) (June,2001) 229–241.
[9] M.W.M.G. Dissanayake, S. Huang, Convergence and consistency analysis for extended Kalman filter based SLAM, IEEE Transactions on Robotics and Automation 23 (5) (October,2007) 1036–1049.
[10] A. Doucet, N. Freitas, K. Murphy, S. Russel, Rao Blackwellised particle filtering for dynamic Bayesian networks, in: Proc. of the UAI, UAI-2000, 2000.
[11] T. Duckett, A genetic algorithm for simultaneous localization and mapping, in: Proceedings of the 2003 IEEE International Cooference on Robotics and Automation, ICRA-2003, Taipei, Taiwan, September 14–19, 2003, pp. 434–439.
[12] U. Frese, A discussion of simultaneous localization and mapping, Autonomous Robots 20 (2006) 25–42.
[13] J. Guivan, E. Nebot, Optimization of the simultaneous localization and map building algorithm for real time implementation, EEE Transactions on Robotics and Automation 17 (3) (2001) 242–257.
[14] S.J. Julier, J.K. Uhlmann, A counter example to the theory of simultaneous localization and map building, in: Proc. of the Int. Conference on Robotics and Automation ICRA-01, Seoul, Korea, 2001, pp. 4238-4243.
[15] J. Leonard, H. Feder, Decoupled stochastic mapping, IEEE Journal of Ocean Engineering 26 (4) (2002) 561–571.
[16] F. Lu, E. Milios, Robot pose estimation in unknown environments by matching 2-d range scans, Intelligent and Robotics Systems 18 (1997) 249–2715.
[17] A. Martinell, N. Tomatis, R. Siegwart, Some results on SLAM and the closing of the loop problem, in: Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots Systems IROS-05, Edmonton, AB, Canada, 2005, pp. 334–339.
[18] M. Montemerlo, S. Thrun, Simultaneous localization and mapping with unknown data association using FastSLAM, in: Proc. of the Int. Conference on Intelligent Robotics and Automation ICRA-03, Taipei, Taiwan, vol. 2, 2003, pp. 1985–1991.
[19] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM: A factored solution to simultaneous localization and mapping, in: Proc. of the Nat. Conference on Artificial Intelligence AAAI, 2002.
[20] L. Moreno, S Garrido, M.L. Muñoz, Evolutive filter for robust mobile robot global localization, Robotics and Autonomous Systems 54 (2006) 590–600.
[21] L. Moreno, S Garrido, D. Blanco, Mobile robot global localization using an evolutionary MAP filter, Journal of Global Optimization 37 (2007) 381–403.
[22] R. Smith, M. Self, P. Cheeseman, Estimating uncertain spatial relationships in robotics, in: I. Cox, G. Wilfong (Eds.), Autonomous Robots Vehicles, Springer-Verlag, 1990, pp. 167–193.
[23] R. Smith, P. Cheeseman, On representation and estimation of spatial uncertainty, The International Journal of Robotics Research 5 (4) (1986) 56–67.
[24] R. Storn, K. Price, Differential Evolution- A simple and efficient adaptive heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.
[25] J. Tardos, J. Neira, P. Newman, J. Leonard, Robust mapping and localization in indoor environments using sonar data, International Journal Robotics Research 21 (4) (2002) 311–330.
[26] S. Thrun, Simultaneous mpping and localization with sparse extended information filters: theory and initial results. in: CMU-CS-02-112, 2002.
[27] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, The MIT Press, 2005.
[28] S. Williams, G. Dissanayake, H. Durrant-Whyte, An efficient approach to the simultaneous localization and mapping problem, in: Proc. of IEEE Int. Conf on Robotics and Automation ICRA-02, 2002, pp. 406–411.
[29] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, International Journal of Computer Vision 13 (2) (1994) 119–152.

Luis Moreno received the Degree in Automation and Electronics Engineering in 1984 and the Ph.D. degree in 1988 from the Universidad Politécnica de Madrid, Madrid, Spain. From 1988 to 1994, he was an associate professor at the Universidad Politécnica de Madrid. In 1994 he joined the Dpt of Systems Engineering and Automation, Universidad Carlos III de Madrid, Madrid, Spain, where he has been involved in several mobile robotics projects. His research interests are in the areas of mobile robotics, mobile manipulators, environment modelling, path planning and mobile robot global localization problems.

Santiago Garrido received the Degree in Mathematics in 1979 from the Complutense University of Madrid, the Degree in Physics science in 1985 and the Ph.D. degree in 2000 from the Universidad Carlos III de Madrid, Spain. In 1997 he joined the Dpt of Systems Engineering and Automation, Universidad Carlos III de Madrid, Madrid, Spain, where he has been involved in several mobile robotics projects. His research interests are in the areas of mobile robotics, mobile manipulators, environment modelling, path planning and mobile robot global localization problems.

Dolores Blanco received the B.S. degree in Physics from the University Complutense of Madrid, Spain in 1992. She received the Ph.D. degree in Mecatronics from University Carlos III of Madrid (UC3M) in 2002. From 1996 to 1999, she was a fellowship student at the Department of Systems Engineering and Automation of UC3M. Since 1999, she is assistant professor in the same Department. Member of Mobile Manipulator Group at UC3M. Her current research includes sensor based path planning, localization and control for mobile manipulators.

M. Luisa Muñoz was born in Madrid, Spain. She graduated in physics science from the Complutense University of Madrid and received the Ph.D. degree in 1988. During her predoctoral period she colaborated as a researcher at the Scientific Research Council (CSIC), Spain. Since 1990, she is currently an Associate Professor of Computer Technology and Architecture at the Polytechnic University of Madrid, Spain. She has been researching in a broad range of topics that covering signal processing, neural network algorithms, image processing and computational vision. Nowaday, her main research interests include evolutionary algorithm applications to mobile robotics.