# A Genetic Algorithm for Simultaneous Localization and Mapping

Tom Duckett
Centre for Applied Autonomous Sensor Systems
Dept. of Technology, Örebro University
SE-70182 Örebro, Sweden
`http://www.aass.oru.se`

***Abstract*— This paper addresses the problem of simultaneous localization and mapping (SLAM) by a mobile robot. The SLAM problem is defined as a global optimization problem in which the objective is to search the space of possible robot maps. A genetic algorithm is described for solving this problem, in which a population of candidate solutions is progressively refined in order to find a globally optimal solution. The fitness values in the genetic algorithm are obtained with a heuristic function that measures the consistency and compactness of the candidate maps. The results show that the maps obtained are very accurate, though the approach is computationally expensive. Directions for future research are also discussed.**

## I. Introduction

Maps are very useful for navigation by mobile robots in complex environments, being needed for tasks such as self-localization, path planning, manipulation of objects, and interaction with humans. To navigate in unknown environments, an autonomous robot requires the ability to build its own map while simultaneously maintaining an estimate of its own position. This is a hard problem because the same, noisy sensor data must be used for both mapping and localization. We can separate two major sources of uncertainty in solving this problem:

(i.) the continuous uncertainty in the positions of the vehicle and the observed environmental features (e.g., due to sensor noise, uncertain execution of motor commands, etc.), and

(ii.) the combinatorial labelling problem of data association (e.g., landmark identification, feature recognition, place recognition, etc.) in which a correspondence must be found between sensor measurements and the features already represented in the map.

Most current solutions to the SLAM problem consider only the first type of uncertainty, and assume that the data association problem is solved when observations are integrated into the map (e.g., it is typical to assume that all landmarks can be identified uniquely). However, this assumption is bound to fail sooner or later for robots operating in complex environments. In short, data association failures produce localization errors, which can lead to catastrophic errors in the map. Without this assumption, the robot must somehow search the *space of possible maps*, since alternative assignments in data association can induce very different maps.

In this paper, a new approach is proposed in which SLAM is defined as a global optimization problem, and the objective is to search the space of possible robot trajectories. There is *no explicit data association or self-localization phase*. Instead, the recorded odometry data of the robot is used as a model from which an initial population of possible trajectories is randomly generated. Each of these trajectories is then evaluated by constructing a global occupancy map using the recorded range-finder data of the robot along the travelled path. A fitness value is calculated for each of the candidate solutions, based on the consistency and compactness of the maps produced, and the search proceeds using a genetic algorithm (GA). GAs are a well-known search technique in which simplified, numerical forms of the biological processes of selection, inheritance and variation are used to improve the average fitness of the population through successive iterations [7].

The rest of this paper is structured as follows. After a brief review of related work, SLAM is defined as a global optimization problem (Section 3), followed by a description of the genetic algorithm developed to solve it (Section 4). Experimental results using sensor data from a real robot are then presented, followed by conclusions and suggestions for future work.

## II. Related Work

A classical solution to the SLAM problem is the Extended Kalman Filter (EKF), a linear recursive filter that estimates the absolute position of the robot and all of the landmarks in the map (see e.g., [2], [5]). The EKF requires analytic models of the vehicle motion and observations, it makes a number of assumptions which are often violated in practice, and it will fail whenever data association fails.

Nebot et al. [14] extended the EKF with a Monte Carlo sampling technique for handling possible data association errors. The EKF is used for normal operation, only switching to the sampling technique to resolve ambiguities in landmark identification, e.g., when a large loop is to be closed. While this approach may reduce the number of data association errors, it does not preclude them in all situations. Measurements for incorrectly identified landmarks would still be integrated by the EKF, producing an incorrect map.

Gutmann and Konolige [6] considered mapping of large, circular environments using a combination of topological and metric representations. An incremental version of the Lu and Milios algorithm for registration of laser scans [10] was first used to find the most consistent metric representation given the current topology. Then a global correlation procedure [8] was used to detect when a previously visited location had been reached. This would trigger iteration of the local registration algorithm to obtain a new metric map based on the corrected topology, thus "closing the loop". A drawback of this approach is that false matches in the global registration procedure could produce catastrophic mapping errors.

Thrun et al. [16] used an expectation maximization (EM) algorithm for robot mapping, which alternates between an 'E-step' that estimates the trajectory of the robot given the current map and an 'M-step' that estimates the map given the current trajectory. While effective, this technique is basically a gradientbased hill-climber or local optimization technique, so it depends upon a good initial solution to avoid becoming trapped in a local optimum.

Montemerlo et al. [11] introduced a hybrid approach using a particle filter to track the pose of the robot, where each particle is associated with a set of Kalman filters estimating the position for each landmark. This approach has the advantage that it is able to represent and search between multiple hypotheses for the full map (i.e., each hypothesis comprises the robot pose and all landmark positions). It has the disadvantage that the particle set must be large enough to include a particle sufficiently close to the true pose of the robot at all times, which may not be practical when closing very large loops.

## III. SLAM as a Global Optimization Problem

In an optimization problem, the aim is to minimize or maximize some objective function. In a global optimization problem, there may be many solutions which are locally optimal. The goal is to find the one best solution, avoiding the local optima [15].

In this paper, SLAM is treated as a continuous global optimization problem with the following elements:

(i.) The search is carried out in the space of possible robot trajectories. A trajectory can be defined as a vector $[\delta_1, \alpha_1, ..., \delta_N, \alpha_N]$ where $\delta_j$ and $\alpha_j$ are the relative distance and rotation travelled by the robot in one small step $j$, and there are $N$ steps in total.

(ii.) The robot's own measurements of its trajectory are used as a generative model. In general, these measurements will be corrupted by noise, e.g., due to odometer drift error. We use the recorded odometry trace of the robot to generate candidate solutions by applying different correction factors to the measured values of $\delta_j$ and $\alpha_j$.

(iii.) Candidate solutions are coded as a vector of correction factors. The trajectory of the robot is first divided into $M$ segments, where $M \ll N$ in general, in order to reduce the number of variables optimized to a manageable level. One solution consists of a vector $[\Delta\delta_1, \Delta\alpha_1, ..., \Delta\delta_M, \Delta\alpha_M]$ where $\Delta\delta_k, \Delta\alpha_k$ are the correction factors applied to the distance and angle measurements within one segment $k$. We assume that the noise properties are uniform along the trajectory within each segment.

(iv.) A set of allowed moves for generating new solutions from previous ones is defined.

(v.) An evaluation function is used to assess the quality of the candidate solutions. This is implemented by inferring a map from each candidate trajectory, then using a heuristic function to assess the quality of the maps obtained.

With this approach, a global search algorithm can be used to search the space of possible trajectories.

The goal is to find the vector of noise parameters, and therefore the inferred trajectory, which produces the best map. In this paper, the SLAM problem is thus solved by a genetic algorithm, described as follows.

## IV. The Genetic Algorithm

Genetic algorithms are a global search technique which mimic aspects of biological evolution, namely the process of natural selection and the principle of "survival of the fittest". They use an adaptive search procedure based on a population of candidate solutions or "chromosomes". Each iteration or "generation" involves a competitive selection procedure that favours fitter solutions and rejects poorer solutions. The successful candidates are then recombined with other solutions by swapping components with one other; they can also be mutated by making a small change to a single component. The procedure is repeated for many generations, producing new solutions that are biased towards regions of the search space in which good solutions have already been found.

### *A. Chromosome Encoding*

Each chromosome is encoded as a string of floating point numbers corresponding to the correction factors applied to the recorded odometry data. In the experiments presented here, the odometer trace was divided into segments of 3 meters in length. For the environment of Fig. 1, there were 50 segments corresponding to the 150 meters travelled by the robot.

For each segment $k$, the chromosome contains two floating point numbers $-d_{max} \leq \Delta\delta_k \leq +d_{max}$ and $-a_{max} \leq \Delta\alpha_k \leq +a_{max}$ that encode the correction factors applied to the distance and angle measurements respectively. That is, the measured values of $\delta_j$ and $\alpha_j$ obtained from the robot's odometry are assumed to lie within a fixed range of their true values (in the experiments presented here, a range of $\pm 2\%$ was assumed for both the distances and angles). The initial population of chromosome is obtained by randomly initializing the values of $\Delta\delta_k$ and $\Delta\alpha_k$ in this range. It would also be possible to assume some parametric distribution (e.g., Gaussian) for the correction factors during initialization.

### *B. Fitness Function*

The fitness function contains domain specific knowledge that is used to assess the quality of the candidate solutions. An important property is that it must be very fast, since it may be executed thousands of times in one run of the algorithm.

In this paper, a much simplified version of the occupancy grid model [12] is used to construct a map for each candidate solution, then two heuristics are combined to obtain a fitness value for each map. The whole procedure runs in approximately 0.95 seconds on a 200MHz Pentium II processor for one candidate solution using the map data of Fig. 1 with 25971 readings from a SICK laser scanner (only 1 out of every 5 readings is used in each scan, where the scans are taken at intervals of 10 *cm*, and the remaining readings are discarded). The procedure is described as follows.

The recorded odometry data is corrected using the correction factors encoded in the chromosome. Then the corrected odometry is combined with the recorded laser range-finder data of the robot to build an occupancy grid with a resolution of 10 *cm*. Because of the use of laser data, we do not need complex ray or cone models; a simple line model is sufficient. For each cell $i$ in the grid, two quantities are calculated: $occ_i$, the number of laser readings which indicate that the cell is occupied, and $emp_i$, the number of readings which indicate that the cell is empty. The following heuristic functions are then calculated:

*1) Map Consistency (MC1):* The idea here is to measure the overall consistency of the sensory information contained in the gridmap. We try to measure the degree of disagreement or "conflict" between the sensor readings. A similar metric was proposed by Murphy et al. [13] based on Dempster-Shafer theory. The measure is calculated as

$$MC_1 = \sum_i min(occ_i, emp_i)$$

by taking the minimum of the $occ_i$ and $emp_i$ values for all cells $i$.

*2) Map Compactness (MC2):* Early experiments with the map consistency measure showed that the genetic algorithm could produce maps with very low conflict but which were obviously incorrect to the human eye. The GA would try to avoid conflict – rather than resolving it – by "twisting" certain critical areas of the map away from one another, when the represented regions were in fact physically adjacent (see Fig. 2).

The idea here is to reward the GA for producing

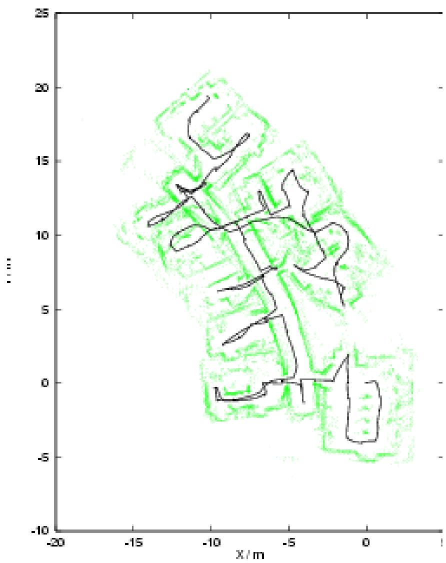


Fig. 1. Raw sensor data from the Artificial Intelligence Lab, Freiburg, as in [6], showing the odometry trace and laser range-finger readings.

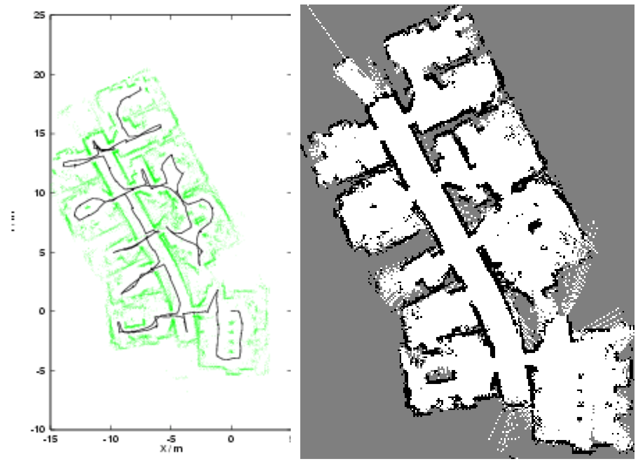


Fig. 2. Corrected sensor data obtained using only the fitness function $F = MC_1$ and the corresponding gridmap.

smaller, more compact maps. This is done by fitting a bounding box to the map that indicates the total area covered by cells with $occ_i > 0$. The measure is calculated as

$$MC_2 = (x_{max} - x_{min}) \times (y_{max} - y_{min})$$

where $x_{max}$ and $x_{min}$ refer to the maximum and minimum $x$-coordinates of the bounding box, measured in number of grid cells.

*Fitness Value:* Finally, the two metrics are combined as

$$F = MC_1 + wMC_2$$

where the weight $w = 0.3$ determines the relative importance of the two heuristics in the fitness function. Note that better maps produce lower values of $F$.

### C. Selection, Crossover and Mutation

The selection phase in a genetic algorithm involves creating a "mating pool" by picking individual solutions that are fitter with higher probability. The selected individuals in the mating pool are then combined to make a new population using the crossover operator, with occasional small random changes due to the mutation operator, described as follows.

In the GA presented here, the population is first sorted according to the values produced by the fitness function. Individuals are then assigned an offspring count that is solely a function of their rank, using the scheme proposed by Baker [1]. In this method, each string l is assigned an expected offspring count $e_l$. The fittest individual is assigned a value of $e_l = \lambda_{max}$, the weakest individual is assigned a value of $e_l = \lambda_{min}$, and the other members of the population are assigned an intermediate value by linear interpolation. In this paper, values of $\lambda_{max} = 1.5$ and $\lambda_{min} = 0.5$ were used.

The standard selection method in a GA is the weighted roulette wheel [7]. However, this scheme has the problem that the best individuals may fail to produce offspring in the next generation, resulting in a so-called stochastic error [9]. Instead, a scheme known as Remainder Stochastic Sampling Without Replacement is used. Each individual is first allocated offspring according to the integer part of $e_l$ in completely deterministic fashion. The remaining strings needed to fill out the population are then obtained randomly by generating new offspring for each string with probability equal to the fractional part of $e_l$. "Without replacement" means that strings can only be selected once in the non-deterministic phase, so that a string with $e_l = 1.5$ would receive either 1 or 2 offspring in total.

Pairs of selected strings are then combined by crossover. Multiple crossover sites are used, so that the encoded values in the two mating strings are completely mixed up in the two strings produced. This is achieved by randomly choosing between the two parents at each site in the chromosome. Crossover is carried out with probability $p = 0.85$, otherwise the selected strings are left unchanged.

Mutation is carried out by picking single values

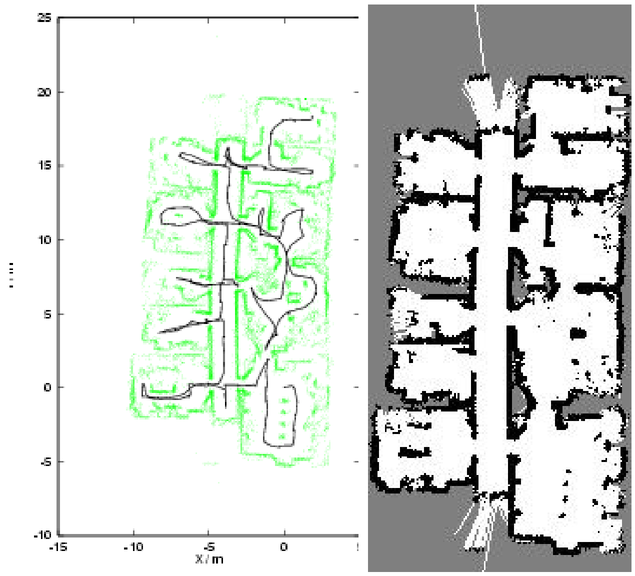


Fig. 3. Corrected sensor data obtained using the full fitness function $F = MC_1 + wMC_2$ and the corresponding gridmap.

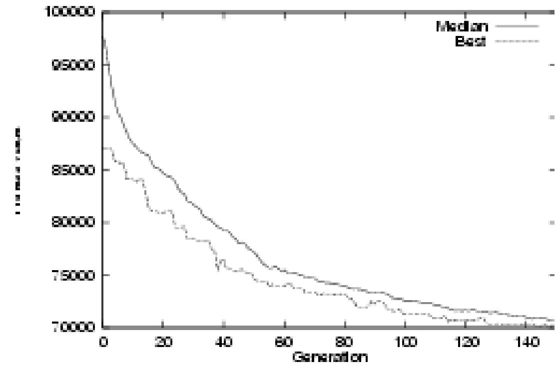


Fig. 4. Fitness values over time for the genetic algorithm using the data of Fig. 1, showing the fitness of both the best and median members of the population in each generation.

within the strings with very low probability $p = 0.005$ and replacing those values with randomly generated values, as upon initialization.

## V. Experimental Results

The algorithm was first tested using data recorded by a Pioneer I mobile robot equipped with a SICK laser scanner at the Artificial Intelligence Lab of the University of Freiburg [6]. The raw sensor data is shown in Fig. 1. The genetic algorithm was run for 150 generations with a population size of 50, i.e., a total of 7500 candidate maps were evaluated, taking around 2 hours on a 200MHz Pentium II processor. Fig. 4 shows the resulting fitness values produced, illustrating the convergence of the algorithm over time. The corrected sensor data obtained from the fittest solution, together with the corresponding gridmap, is shown in Fig. 3.

## VI. Conclusions and Future Work

The major contribution of this paper is that the problem of simultaneous localization and mapping (SLAM) has been defined for the first time as a global optimization problem. A genetic algorithm was developed to solve this problem, producing a global solution by searching the space of possible robot maps.

Genetic algorithms are particularly useful for solving environment modelling problems because they exploit the building block hypothesis, whereby better and better strings are constructed from the best partial solutions (i.e., schemata or building blocks) of previous generations [4]. Most real world environments can be decomposed spatially into natural building blocks such as rooms, corridors, open spaces, etc.

A major benefit of the approach is that it makes very few assumptions about the underlying problem, and the only critical parameter is the weighting between the consistency and compactness heuristics in the fitness function. A disadvantage of the approach at present is that it requires large amounts of computation.

There are other possible algorithms for performing global optimization [15], such as branch and bound algorithms, simulated annealing, etc., which should also be investigated. A further possibility would be to combine the global search algorithm with a local optimization method, e.g., EKF [5], consistent pose estimation [10], relaxation [3], etc. In the case of the GA, this could dramatically reduce the overall computation time.

Another important direction is to implement an online version of the algorithm suitable for use in real-time on a self-navigating mobile robot. This would involve continually searching the space of possible maps while the robot is in motion, automatically extending the candidate maps with the incoming sensor information. The current best map would be used for decision making, e.g., path planning. A further possibility would be to investigate implementation of the algorithm on parallel processors, given the parallel nature of evolutionary computation.

Future work should also include more detailed theoretical and empirical analysis of the problem and the algorithm required to solve it, e.g., with respect to the fitness criteria needed for different environments. Experimental comparisons with other approaches should also be considered.


## Acknowledgments

The author thanks Steffen Gutmann and the AI Lab of Freiburg University for sharing their data, and also Udo Frese and Alessandro Saffiotti for some very interesting discussions on the SLAM problem.


## VII. References


[1] J. Baker. Adaptive selection methods for genetic algorithms. In *Proceedings of the International Conference on Genetic Algorithms and Their Applications*, 1985.

[2] M.W.M. Gamini Dissanayaka, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[3] Tom Duckett, Stephen Marsland, and Jonathan Shapiro. Fast, on-line learning of globally consistent maps. *Autonomous Robots*, 12(3):287–300, 2002.

[4] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[5] J.E. Guivant and E.M.Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001.

[6] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, 1999.

[7] J.H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI, 1975.

[8] K. Konolige and K. Chou. Markov localization using correlation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1999.

[9] C.-T. Lin and C.S.G. Lee. *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Prentice Hall, 1995.

[10] F. Lu and E.E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[11] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.

[12] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'85)*, pages 116–121, 1985.

[13] R. Murphy, B. Sjoberg, A. Schultz, and B. Adams. Dempster-shafer weight of conflict metric as an indicator of mapping errors. In *Proceedings of the IJCAI-2001 Workshop on Reasoning with Uncertainty in Robotics*, August 4-5, 2001.

[14] E. Nebot, F. Masson, J. Guivant, and H. Durrant-Whyte. Robust simultaneous localization and mapping for very large outdoor environments. In *Proceedings of the 8th International Symposium on Experimental Robotics (ISER '02)*, 2002.

[15] J.D. Pintér. Continuous global optimization: An introduction to models, solution approaches, tests and applications. *Interactive Transactions of ORMS*, 2(2), 2002. `http://catt.bus.okstate.edu:80/itorms/`.

[16] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localisation for mobile robots. *Machine Learning*, 31(5):1–25, 1998.