Proceedings Paper:

Boland, Miguel and Soares Indrusiak, Leandro orcid.org/0000-0002-9938-2920 (Accepted: 2018) Analysis of the use of genetic algorithms for indoor localisation via cloud point matching. In: Proceedings of the Genetic and Evolutionary Computation Conference. The Genetic and Evolutionary Computation Conference, 15-19 Jul 2018 , JPN (In Press)

https://doi.org/10.1145/3205455.3205499

Analysis of the use of genetic algorithms for indoor localisation via cloud point matching

Miguel d. Boland
migueldboland@gmail.com
University of York

Leandro Soares Indrusiak
University of York

ABSTRACT

A system's ability to precisely locate itself in a known physical environment is key to its capacity to interact with the environment in an intricate manner. The indoor localisation problem has been approached in a variety of ways, ranging from the identification of pre-defined features or topologies to the more general cloud-point matching.

Cloud point matching can be achieved using a variety of algorithms, each with benefits and drawbacks. Recent improvements have focused on the application of genetic algorithms to solve the initial 'global' search for a solution, before refining this solution to a precise position through a non-genetic algorithm. This project aims to demonstrate the inefficiency of genetic algorithms applied to the global search problem for the issue of indoor localisation; this is thought to be caused by the solution space's low dimensionality, solution landscape topology and the inefficacy of crossover operators in the algorithm. Based on our assumptions of map topologies, we conclude that significant redundancies can be found in some purely genetic heuristics and suggest further development of landscape analysis to allow the use of algorithms appropriate to the scenario's complexity.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

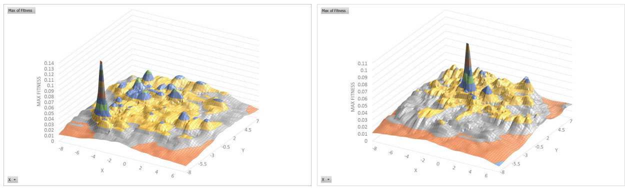
Genetic Algorithms, Indoor Localisation, R

ACM Reference Format:
Miguel d. Boland and Leandro Soares Indrusiak. 2018. Analysis of the use of genetic algorithms for indoor localisation via cloud point matching. In *GECCO '18: Genetic and Evolutionary Computation Conference, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3205455.3205499

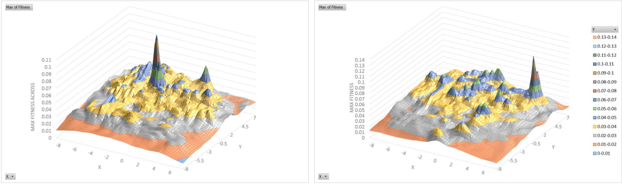
1 INTRODUCTION

The problem of line-of-sight indoor localisation was first resolved through the matching of cloud point data (obtained from a line-of-sight sensor such as a Li-Dar) to retrieve tuple (x, y, θ) describing the location and orientation of a robot in a known environment. This was first achieved by algorithms such as the Iterative Closest Point (ICP) algorithm Besl and McKay [4], and a long line of alternative heuristic algorithms Lu and Milios [23], Diosi and Kleeman [11] [29], Biber and Strasser [5], Donoso-Aguirre et al. [13], Konecny et al. [19] and various improvements on the ICP's convergence speed [12] [32], dataset optimisation [33] [25] or precision metrics [14].

Performing indoor localisation without a priori knowledge of the robot's pose increases the difficulty to this problem, as a global search for the position must now be performed, rather than simply a pose refinement. Using test cases from Lenac et al. [20], we initially found the fitness landscape of indoor localisation problems to be relatively smooth and unimodal, as visible in Figure 1 which displays the best matching translation and rotations of poses at various points and orientations on the map. Mitchell [27] details that GAs are most applicable in non-smooth, non-unimodal search spaces. We may expect these to exist in the context of indoor localisation as feature-dense reference maps, but this amounts to a finer resolution of the current problem, requiring a higher sampling density and computational power but remaining equally solvable. Furthermore, Grefenstette [17] states that "If [a search] space is well understood and contains structure that can be exploited by special purpose search techniques, the use of genetic algorithms is



Scan 18 Scan 110



Scan 115 Scan 240

Figure 1: Fitness landscapes for randomly selected scans in dataset.

GECCO '18, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5618-3/18/07…$15.00
https://doi.org/10.1145/3205455.3205499

generally computationally less efficient". Given the full knowledge of the map and scan data, and the relative ease with which we can construct a fitness landscape relating the two, we can see that the general problem definition contrasts greatly with intended applications of genetic algorithms.

We therefore aim to demonstrate redundancies in the behaviour of genetic algorithms applied to a subset of the indoor localisation problem with knowledge of the environment but no a priori pose. This is achieved by creating improvements to a benchmark genetic algorithms to demonstrate the ability of a simple non-genetic heuristic algorithm to outperform a genetic algorithm in terms of efficiency, as measured by pose precision and computational time.

2 EXISTING WORK

The use of genetic algorithms to search for data-matching solutions was pioneered by Brunnstrom and Stoddart [7] to find the correspondences between detailed surface models. This was achieved by using a chromosome design based on a transformation, translation and rotation in three dimensions. A simplified X/Y translation and rotation chromosome is used as the basis for all further genetic algorithms for 2D indoor localisation.

Robertson and Fisher [30] later presented a GA alternative to the ICP algorithm to avoid requiring a priori pose knowledge and the tendency to converge to sub-optimal or incorrect solutions. These were solved through the GA's ability to search for a global maxima, rather than simply refine a pose to the local minima. This is implemented using a 3-tuple matching our problem definition (x, y, θ), and results in significantly better global search results than a single ICP run, thereby demonstrating the potential of genetic algorithms within the field.

Polar Scan matching (PSM) is a variation of Robertson and Fisher's initial genetic algorithm which is adapted for the direct use of raw data from a laser range scanner, therefore reducing the computational costs of the operation. Ze-Su et al. [35] believe this would represent two $O(n)$ searches: one for the translation estimation, and one for the orientation estimation. This approach is found to be more precise and efficient than ICP in the given examples [35], although given the variation in performance of algorithms in scenes [12] this result may not be generalisable. As demonstrated by Ze-Su et al., the method is applied to identify two complete sets of data, rather than mapping a subset of the data (the area visible around the robot) into the full set of data (the full map); further adaptation may therefore be required for the method to function for general indoor localisation.

Recent improvements in the performance of GAs were suggested by Lenac et al., but involve a trade-off in accuracy with execution time due to the use of a rasterized environment.

The concept of combining the global search of a GA with the accuracy of the ICP algorithm has been introduced in a variety of concepts. Brunnstrom and Stoddart [7] first proposed the idea of applying a low-accuracy global search using a GA, before refining the most promising individual poses using an ICP algorithm using the pre-aligned poses. Using a fitness function defined by minimising the sum of the distance between pairs of closest points (each pair composed of a point in the scan and a point in the map), Brunnstrom and Stoddart presents an objective set of results demonstrating the algorithms ability to roughly estimate the 3-tuple pose modification, but produces no statistical data regarding the success rate or accuracy of the algorithm.

The hybrid approach was independently presented by Martínez et al. [24], resulting in a method which is indistinguishable from a standard GA, but is however quicker to execute as the GA search can be completed in a coarser accuracy. This utilises a fitness function similar to the PSM algorithm [35], thereby reducing the complexity of the fitness function to $O(n)$. When compared to a standard ICP and GA, the hybrid GA-ICP method performs as well as the GA and better than ICP, with a computation time between that of an ICP and GA. As no statistical analysis is performed, it is difficult to demonstrate this hybrid approach to be superior to other available methods (as ICP is known to be a local search algorithm, and is therefore not representative of other global search algorithms [34]). As such, further evaluation of the GA-ICP algorithm in a larger variety of environments would be required to find the strengths and weaknesses of the approach relative to differing environments. One should note these papers used a basic form of ICP, and as such were quickly improved upon as discussed below.

Further hybrid algorithms include a combination of GA/TrICP [22] which improved on previous GA/ICP algorithms [7][24], and a rasterized GA / mbICP algorithm by Lenac et al. [21].

3 METHODOLOGY

The data used to evaluate our algorithms is taken from Lenac et al. [21], where a robot's exploration of a room was simulated using the Player-Stage software [2]. This produced a series of scan scenarios, each composed of a verifiable scan pose (x, y, rotation) and polar scan coordinates (distance, rotation) which mimic LiDar scanner data. The map of the environment (into which our algorithm will locate itself) was then calculated from these scans into a cartesian dataset, and then subsampled to a tolerance of 0.2 units such as to speed up the algorithm's execution Figure 2. We should note that the data utilised had no specified scale: this can be estimated using the specifications of an off-the-shelf Li-Dar range finder [1].

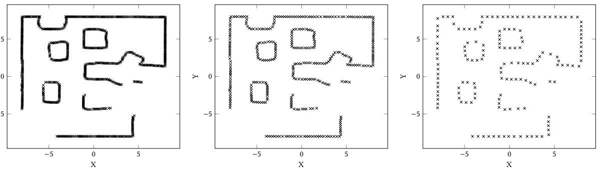


Figure 2: Effects of sub-sampling on features in the reference map, with tolerances of 0.1, 0.2 & 0.5 units respectively.

The algorithms were each run 30 times using the same scan (scan110) and map, over a range of time limitations; these restricted the number of generations for which the algorithm evolved, thereby providing a comparison of the algorithms over a set of possible requirements (which could be constricted by the application, available processing power, etc). Paired T-Tests could then be conducted across the average efficiency, categorised into buckets by real execution time. No values were excluded as outliers, as we aim to create a consistent system accounting for the stochastic nature of genetic algorithms.

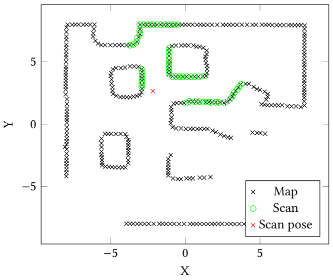


Figure 3: Scan 110 as located in the reference map, with pose.

4 SOLUTION EVALUATION

We define a combined error metric $E = d_{hp} \times |R_h - R_r|$, where d_{hp} is the absolute distance between each estimated pose, and R_h, R_p are the respective rotations from North of the hypothetical and reference pose. As such, a smaller error represents a more accurate pose. This allows the outputs of our algorithms to be evaluated independently from their fitness functions, and therefore allow us to compare output poses. As we are only searching for a single pose, we will only consider the best individual from each run's final generation as the output of an algorithm. These form the set of results for each algorithm which will be evaluated using statistics appropriate to individual experiments. An additional measure of efficiency is defined as the product of the combined error and the execution time of the algorithm, thereby allowing quicker algorithms to be precise algorithms.

5 BENCHMARK ALGORITHM

An existing GA by Robertson and Fisher [30] was adapted as a benchmark algorithm for the purpose of this project; it consists of a standard GA with incremental/decremental mutation, parameter-specific crossover (parameters are x, y, θ) and tournament based selection. Different termination conditions were used: these were either generation based (maximum number of permitted iterations) or time based (maximal allowed execution time, which permits a generation to finish if it was started before the time limit). The mutation rate, crossover rate, population size and number of generations were sequentially optimised for a given scan to maximise the algorithm's performance in our test data, creating a robust benchmark.

The fitness function defined by Robertson and Fisher [30] was inverted from a minimisation ($E = \sum_i |S_i - M_I|$ where S is a (x, y) point in the scan and M is S's closest point in the map) to a maximisation $M = \frac{1}{1+E}$; This provides a more accentuated curve of fitness in the hope of improving the convergence capabilities of both the benchmark and new algorithm, in addition to adhering standards for GAs established by Eiben and Smith [15]. This was validated using the fitness landscape in Figure 1, which corresponded to the solution pose.

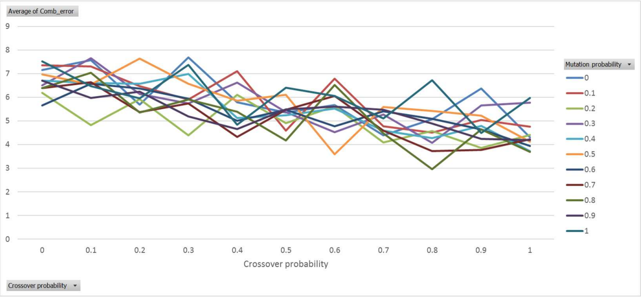


Figure 4: Effects of crossover and mutation rates on the benchmark algorithm's execution time.

Mutation and crossover probabilities were optimised for our test case (scan110) by execution the algorithm from 60 randomly generated poses; the lowest average pose error (defined in section 4) was selected, resulting in a optimal parameters of CXPB=0.8 and MUTPB=0.8, as visible in Figure 4. A similar analysis was conducted using these parameters to mutation sizes, as sampled from a normal distribution with $\mu = 0, \sigma = 1.0$; this was deemed to provide a balance between the frequency in small mutations (to adequately refine the final pose) and larger mutations (to increase the convergence rate of the pose from the initial pose). The optimised standard deviation of this distribution for our test scan (scan 110) was verified by running the algorithm 30 times using previously defined parameters and a varying mutation size, as visible in Figure 5. Population sizes and number of generations were set to 50, such as to provide a more practical execution time which would mimick the specifications of an embedded system, whilst increasing the difficulty of the problem for all algorithms evaluated.

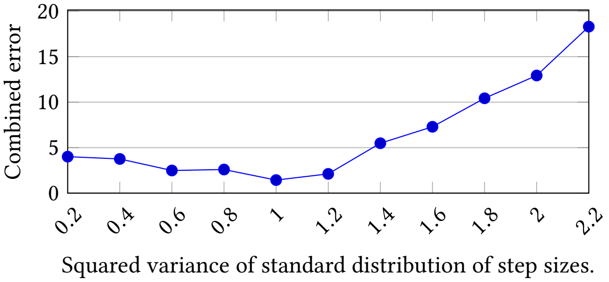


Figure 5: Effects of varied mutation sizes on performance of the benchmark algorithm.

6 ALGORITHM DESIGN

On the assumption of a smooth fitness landscape, we implement an alternative algorithm to accelerate the convergence of the GA. Building on David E. Goldberg [10]'s solution to balancing the conflict between exploration and exploitation, which is achieved through higher growth ratios followed by building block discovery through mutation, we diverge by instead utilising mutation to optimise our current local maxima and assume our initial population was sufficiently dense and spread across the map to find all local maximas. The evolutionary behaviour represented a form of elitism, where the top *n* percentile of the population was retained at each generation and duplicated into offspring, which increase the genetic diversity via crossover or mutation according to a set probability. This is known at the $(\mu + \lambda)$ algorithm [3] [31], where μ represents the number of fittest individuals to select at each generation, and λ represents the number of offspring to generate. The mutation rate was set to 1.0, whilst no crossover was used; this minimised the average resultant combined error by maximising the incremental movement of individuals and reduced the occurance of destructive crossover. A high elitism rate of 0.95 was found to minimise the combined error when evaluated across a possible range of [0, 1]. Mutation sizes were drawn from a normal distribution with $\mu = 0, \sigma = 1.05$. This was deemed to provide a balance between the frequency in small mutations (to adequately refine the final pose) and larger mutations (to increase the convergence rate of the pose from the initial pose). The near-optimality of this distribution for our test scan (scan 110) was verified by running the algorithm using previously defined parameters and a varying mutation size. An additional improvement of the population's initialisation involved arranging individuals into a grid layout; this improved the probability that all local minimas would be explored, as the elitism rate rarely removes individuals from the population. In order to provide an adequate breadth of search without a large computational overhead, we also experimented with a grid-like instantiation of individuals (see Figure 6). This enabled the algorithm to evaluate a larger number of candidate individuals, of which the top 50 are used as a primed population in the elitist algorithm previously described.

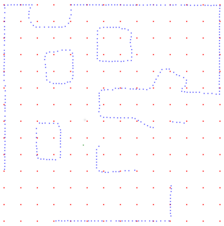


Figure 6: A grid based initial population

7 EVALUATION AGAINST BENCHMARK

These two improvements were evaluated against our benchmark GA in succession, with the latter grid-arrangement including the former elitist selection and parameters. All 3 were evaluated 30 times using optimal parameters for each algorithm, and scan110 as a representative of scans of the dataset.

7.1 Elitist selection

This found the elitist algorithm to be more precise with limited algorithmic capacity, with a mean combined error of 0.191 against 1.839. The result was validated using a two-tailed T-Test (with unequal variance) between the set of combined errors (N=30, p<0.01). However, the elitist algorithm also required slightly more time to run (see Table 1), which brings into question the efficiency of the algorithm.

	Combined error		**Execution time (s)**	
	Benchmark	**Elitism**	**Benchmark**	**Elitism**
Mean	1.839	0.191	46.923	48.522
Stdev	3.151	0.478	9.160	9.057

Table 1: Combined error and execution time over 30 runs of benchmark and elitist algorithms

Further to the previous experiment, the code was modified to loosely constrain the available execution time; this functioned by halting the algorithm if, at the end of a generation, the elapsed time was larger than a specified threshold. As the subsequent results had non-exact execution times, the results were weighed according to their execution time, such that the statistics were ran on values representing the product of the execution time and combined error. Lower values therefore represent better efficiency in resolving the problem.

The elite algorithm was found to produce more efficient results across the set of target execution times Figure 7. This is most visible by the comparatively low median (0.550 against 12.960), along with a more efficient upper quartile (2.110 against 82.407). This demonstrates that the elite algorithm produces consistently more efficient solutions than the benchmark algorithm. Poor efficiency values still occur, as demonstrated by the large whiskers.

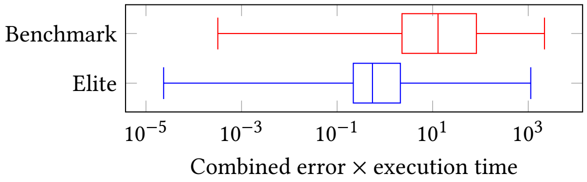


Figure 7: Performance of each algorithm with 50 population, as many generations as possible within the time frame and optimal parameters for each algorithm.

These results can be further analysed in Figure 8, where the results are bucketed by execution time. A T-Test (paired across time buckets) demonstrates the efficiency of the elitist algorithm (i.e: lower execution time × combined error) per execution time (means of 23.622 vs 100.865, p<0.001, N=360). This is not the case for all time limits; benchmark runs limited to 5-10s are not significantly more efficient (23.724 average combined error × execution time) than the elite algorithm (26.231). This may be due to the benchmark GA's use of crossovers to rapidly explore areas of the map which are not yet populated; an equivalent breadth of search is not available in our algorithm due to the small mutation sizes and lack of crossover; this does appears to disrupt the mutation-based refinement, leading to a higher error in the benchmark algorithm.

Fitness-rank based selection is capable of rapidly selecting promising individuals, duplicating them and mutating their offspring to find an optima near them. This results in an efficient algorithm for a majority of time scales, but can still fail to produce a precise pose given an arbitrary amount of computational time (as demonstrated

Figure 8: Average run time × combined error for each algorithm, using as many generations as possible in time limit, optimal parameters for each algorithm and population of 50.

by the large upper percentiles in Figure 7). This may be due to the stochastic nature of the genetic algorithms, or more particularly the population initialisation, and may occur if no individual's local maxima is the global maxima. Combining elitism with a grid-instantiation was next evaluated using the same methodology with which our algorithm was previously compared to the benchmark. The elitist algorithm including a grid-based initialisation of 200 individuals (followed by the evolution of the 50 best individuals) was executed 30 times across a range of time limits, producing the data found in Figure 9.

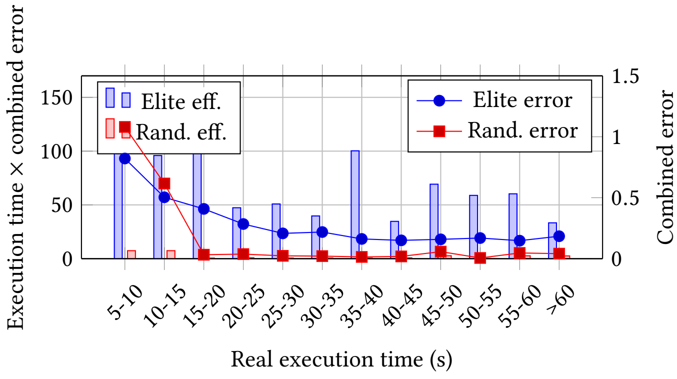


Figure 9: Performance of grid-based initialisation compared to random instantiation, both using elitist selection.

The grid-initialised algorithm was found to be more efficient when compared to a random initialisation (T-Test paired across bucketed real execution time, average execution time × error of 2.26 vs 69.00, p<0.001, N=360). This is likely due to a reduction in the number of algorithm failures, where no adequate pose was found within the time limit; we would expect this to occur less frequently when sampling the map at a higher frequency and consistency. The effects of this are visualised in Figure 10, where the standard deviations for each execution time group are generally lower for grid-based initialisation than the standard algorithm. This is found to be statistically significant for two thirds of the groups using an F-Test (p<0.001), therefore demonstrating that grid-based initialisations greatly increase the accuracy of the algorithm. We should note that optimal grid density or parameters were not explored for the map or scan, and it could therefore be possible to further accentuate the effect of this method on the output pose.

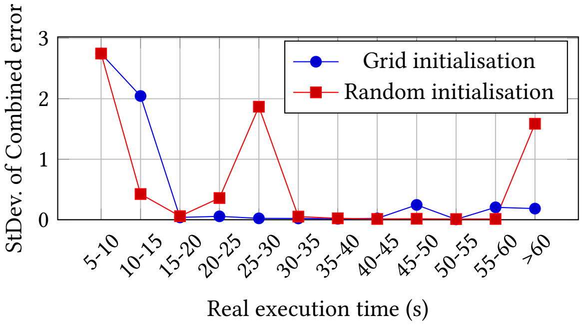


Figure 10: Improved pose accuracy and algorithm success from grid-based initialisations compared to random initialisation.

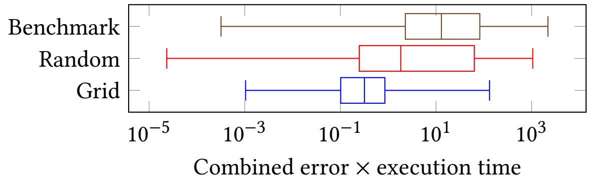


Figure 11: Performance of each algorithm over 30 executions for various time limits.

We should additionally note that while the feature was envisioned using a grid pattern, the use of a larger initial population which is randomly distributed around the map does achieve a similar performance. This was validated over 30 runs of the algorithm, which produced a mean result (in execution time × combined error) of 3.74 across all execution time buckets, compared to the grid-based algorithm's mean of 2.255. A paired T-test across execution time buckets confirmed that the mean standard deviations were not statistically discernible from grid-based initialisations (N=360, p=0.295).

As such, large initial populations of individuals are hugely beneficial to the evolution of a pose through elitism selection, as they increase the density with which the map is sampled, thereby increasing the probability that an individual will be placed in a position from which it would crawl (via mutation and duplication) towards the global maxima. This contributes to improving the consistency of pose retrieval, as well as reducing the worst case pose, as demonstrated by the small IQR of the box plot representing the

Grid algorithm's results in Figure 11.

We therefore demonstrate that optimisations are possible for purely genetic localisation algorithms, improving the accuracy, computational requirements and precision of our localisation within a map. The optimised algorithm has been reduced away from a volatile genetic algorithm, which benefits from crossover to explore a large map with little computational power. In it's place, we rely on finding a reasonable initial pose estimation through a sparse brute force, before refining it through repeated mutation to ascend the fitness gradient. As the mutation has a likelihood of approximately 50 % of being counter-productive (as it can move the individual towards or away from the optima), it stands out as a significant inefficiency in the algorithm, leading us to a necessary comparison with a gradient ascent algorithm, which we will design and evaluate in section 8.

8 ICP ALGORITHM DESIGN & EVALUATION

Given the lack of comparison between genetic algorithm and recent classical algorithms, an additional experiment was undertaken to help highlight inefficiencies in the pose refinement of GAs. Following Yang et al. [34]'s implementation of a Branch-and-Bound & ICP algorithm, an ICP implementation [18] was adapted for use with our grid-layout algorithm. This aimed to form a representative algorithm from the non-genetic research, and preliminary results signified that further optimisation was not required to demonstrate the benefits of this approach. Following Censi et al. [8], who states that it would be possible to apply a 'classical' algorithm to a global pose localisation problem by running it from a number of random poses, the algorithm functioned by applying the ICP algorithm from 200 hypothetical poses laid across a grid on the map, using the same grid pattern and density as the Grid-GA. The previously defined combined error metric (see section 4) is then used to select the final pose estimation from the set of ICP-refined candidate poses. The refinement of candidate poses was executed in parallel using the same compute cluster as in our grid-based GA, thereby utilising an equally maximal amount of concurrency equivalent to our GAs parallel evaluation of individuals. The algorithm was run 360 times to create a dataset of comparable size to the grid-based algorithm's dataset (which was created using 30 runs over 12 target durations). As visualised in Figure 12, the grid-ICP has a similar median efficiency to the grid-based GA (medians: Grid-GA = 0.73, Grid-ICP = 1.33), but is much more consistent. This is demonstrated by the standard deviation lower standard deviation of 7.760 for the Grid-ICP, compared to 68.63 for the Grid-GA. We should note that whilst Figure 12 displays a similar spread of data using interquartile ranges, we decided to comparatively evaluate the efficiency of the algorithms using the standard deviation as we have previously decided not to exclude outliers. As the IQR is 'padded' against outlying values, it is not representative of the worst case values seen in the overflow bin of Figure 13, demonstrating why the Grid-ICP algorithm is more consistent than the Grid-GA.

A paired T-Test across the pose error × execution time results for ICP and the grid-initialised algorithms found the means to be statistically different (N=360, p<0.01). This indicates that the GA based algorithm is indeed more efficient, with a mean efficiency of 2.247 against the Grid-ICP's mean of 4.350. We further found the grid-ICP algorithm to have a much smaller mean execution time compared to the grid-GA: (3.66s, compared to 38.20s)

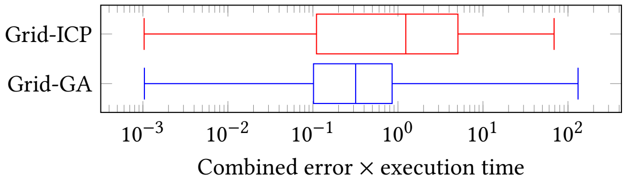


Figure 12: Performance of each algorithm over 30 executions for 12 time limits (note logarithmic x axis)

Figure 13: Histogram comparing the efficiency of Grid-ICP and Grid-GA algorithms

9 EXPERIMENTAL SUMMARY & DISCUSSION

We therefore demonstrate improvements to the application of GAs to the problem of indoor localisation; these are the application of the $(\mu + \lambda)$ algorithm, and priming the algorithm with a larger initial population. These improvements highlight the ineffectiveness of GAs to the problem of indoor localisation, as an unrefined classical algorithm can be prototyped to not only indistinguishly match the precision and efficiency of the algorithm, but would also be more suitable to an embedded application due to it's smaller execution time. Further improvements to the ICP algorithm (such as the mbICP or trICP algorithms) could further reduce the computational power requirement, as would the optimisation of grid arrangements and population density. We hypothesise that this improvement is possible due to the destructive nature of the crossover operator when dealing with co-dependent parameters (as is our case), and the inability of mutations to rapidly evolve individuals in a reliable fashion without being less efficient than random walking. Both the grid-GA and grid-ICP rely on the position of at least one individual in the 'fitness well' of the global maxima, signifying that limitations may arise in large maps which cannot be adequately sampled through a large initial population (or to run the ICP algorithm from a dense grid).

Improvements to our methodology are possible, and may improve the significance of these findings. The use of sequential parameter optimisation ignores any possible co-dependence of these parameters to maximally optimise an algorithm. This was the chosen methodology for practical reasons, as a multi-parameter search would require inaccessible amounts of computation power. Research in meta-genomic by Brain and Addicoat [6] suggests that it could be possible to use a genetic algorithm to optimise our GAs for a given set of scans and a map. This would improve the convergence speed of the candidate solutions, thereby reducing the computational time required to run the algorithm. This is not a feasibly rapid solution given available resources due to the stochastic behaviour of genetic algorithms and the long execution time of our algorithms.

Furthermore, the assumption of a single, smooth and global maxima is central to the function of both grid-based algorithms. The lack of common test-benches (as opposed to other fields such as IRIS dataset in computer vision [16]) further complicates the task of comparing algorithms in generalisable cases. As demonstrated in our landscapes in Figure 1, this is the case for our test scans and maps. However, if exploring a repeated environment where features occur with slight variations (for example offices with desks, cabinets, walls), both the Grid-ICP algorithm and grid-pattern algorithm may converge to an incorrect maxima; depending on the density of the initial grid and the location of individuals within the local topology of the maxima. The inability to later explore areas which are not sampled in the initial population through large mutations or crossovers would prevent either algorithm from searching unexplored areas, causing the algorithms to become stuck in the local maxima and return an incorrect pose.

Therefore, we could hypothesise that feature dense search spaces with associated mountainous fitness landscapes could be better explored by more volatile implementations of GAs, such as the algorithms proposed by Robertson and Fisher [30] or Lenac et al. [20], with additional refinement using a classical algorithm (as proposed by Lenac et al. [21]). Such maps could be caused by either having a poorly featured scan dataset (due to low sensor range relative to map size). Nevertheless, we should note that the use of a single test case proves no guarantee of any form of dominance between algorithms, and that although we postulate that non-smooth fitness landscapes are infeasible given our current fitness metric, the existence of these would form an edge case to our current analysis. Therefore, whilst purely elitist genetic algorithms are ineffective in our test case, a crossover operator may be necessary to explore larger maps; this would also preclude requirement of a short computation time due to the added complexity.

The lack of exploration of the solution space's topology is therefore both an unexplored and central issue to the problem of indoor localisation through GAs; this draws a strong comparison to Mitchell [27]'s statement that "GAs are most applicable in non-smooth, non-unimodal search spaces".

We should note that the findings presented here depend on an underlying assumption; the presence of a smooth fitness landscape, which we can sample with sufficient density to allow for random walking to propagate a candidate pose to the global maxima. We may expect these to exist in the context of indoor localisation as feature-dense reference maps, but this amounts to a finer resolution of the current problem, requiring a higher sampling density and computational power but remaining equally solvable. Furthermore, Grefenstette [17] states that "If [a search] space is well understood and contains structure that can be exploited by special purpose search techniques, the use of genetic algorithms is generally computationally less efficient".

We conclude that whilst it is possible to optimise GAs for a particular map topology and scan, the application of a purely genetic algorithm to the problem of indoor localisation is likely to be inferior in accuracy and execution time when compared to a hybrid algorithm capable of global search and gradient-ascending local refinement (such as [34]). This is likely due to the low dimensionality of the problem, which results in a topographically unimodal fitness landscape. The scope of these findings are limited to purely genetic algorithms, as hybrid genetic algorithms can avoid the difficulty of balancing exploration and exploitation through the use of alternate refinement algorithms [21] or modifications to the behaviour of the GA [9]. These may prove to be extremely beneficial when performing localisation in very large spaces with limited computational capabilities, but further research should be undertaken to comparatively evaluate the behaviours of GAs in these environments against other leading algorithms. The lack of any application of fitness landscape topology to the problem precludes the possibility of asserting any dominance of GA or non-GA algorithms, but, if evaluated against a robust & diverse test bench, could allow the selection of a hypothetically optimal problem for a given scenario. Such work may result from the application of explanatory landscape analysis to indoor localisation, as discussed in Mitchell et al. [28] and more recently Mersmann et al. [26].

ACKNOWLEDGMENTS

REFERENCES

[1] [n. d.]. LIDAR-Lite 3 Laser Rangefinder. http://www.robotshop.com/uk/lidar-lite-3-laser-rangefinder.html. ([n. d.]). Accessed: 2017-4-6.

[2] [n. d.]. Player Project. http://playerstage.sourceforge.net/. ([n. d.]). Accessed: 2017-4-28.

[3] Back, T and Fogel, D B and Michalewicz, T. [n. d.]. *Evolutionary Computation 1 Basic Algorithms and Operators.* IOP Publishing Ltd.

[4] P J Besl and H D McKay. 1992. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 2 (Feb. 1992), 239–256.

[5] P Biber and W Strasser. 2003. The normal distributions transform: a new approach to laser scan matching. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, Vol. 3. 2743–2748 vol.3.

[6] Zoe E Brain and Matthew A Addicoat. 2011. Optimization of a genetic algorithm for searching molecular conformer space. *J. Chem. Phys.* 135, 17 (7 Nov. 2011).

[7] K Brunnstrom and A J Stoddart. 1996. Genetic algorithms for free-form surface matching. In *Proceedings of 13th International Conference on Pattern Recognition*, Vol. 4. 689–693 vol.4.

[8] A Censi, L Iocchi, and G Grisetti. 2005. Scan Matching in the Hough Domain. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation.* 2739–2744.

[9] Chi Kin Chow, Hung Tat Tsui, and Tong Lee. 2004. Surface registration using a dynamic genetic algorithm. *Pattern Recognit.* 37, 1 (2004), 105–117.

[10] Kalyanmoy Deb David E. Goldberg. 1991. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms.*

[11] A Diosi and L Kleeman. 2005. Laser scan matching in polar coordinates with application to SLAM. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems.* 3317–3322.

[12] F A Donoso, K J Austin, and P R McAree. 2017. How do ICP variants perform when used for scan matching terrain point clouds? *Rob. Auton. Syst.* 87 (2017), 147–161.

[13] F Donoso-Aguirre, J-P Bustos-Salas, Miguel Torres-Torriti, and Andres Guesalaga. 2008. Mobile robot localization using the Hausdorff distance. *Robotica* 26, 02 (2008), 129–141.

[14] D W Eggert, A Lorusso, and R B Fisher. 1997. Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Mach. Vis. Appl.* 9, 5-6 (1 March 1997), 272–290.

[15] A E Eiben and James E Smith. 2015. *Introduction to Evolutionary Computing* (2nd ed.). Springer Publishing Company, Incorporated.

[16] R. A. FISHER. 1936. THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS. *Annals of Eugenics* 7, 2 (1936), 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

[17] John J Grefenstette. 2012. *Genetic Algorithms for Machine Learning*. Springer Science & Business Media. 351 pages.

[18] R Harry. [n. d.]. Iterative Closest Point (ICP) implementation on python. http://stackoverflow.com/questions/20120384/iterative-closest-point-icp-implementation-on-python. ([n. d.]). Accessed: 2017-4-12.

[19] Jaromir Konecny, Michal Prauzek, Pavel Kromer, and Petr Musilek. 2016. Novel Point-to-Point Scan Matching Algorithm Based on Cross-Correlation. *Mobile Information Systems* 2016 (24 April 2016).

[20] Kristijan Lenac, Enzo Mumolo, and Massimiliano Nolich. 2007. Fast Genetic Scan Matching Using Corresponding Point Measurements in Mobile Robotics. In *Applications of Evolutionary Computing*. Springer Berlin Heidelberg, 375–382.

[21] Kristijan Lenac, Enzo Mumolo, and Massimiliano Nolich. 2011. Robust and Accurate Genetic Scan Matching Algorithm for Robotic Navigation. In *Intelligent Robotics and Applications (Lecture Notes in Computer Science)*, Sabina Jeschke, Honghai Liu, and Daniel Schilberg (Eds.). Springer Berlin Heidelberg, 584–593.

[22] Evgeny Lomonosov, Dmitry Chetverikov, and Anikó Ekárt. 2006. Pre-registration of arbitrarily oriented 3D surfaces using a genetic algorithm. *Pattern Recognit. Lett.* 27, 11 (2006), 1201–1208.

[23] Feng Lu and Evangelos Milios. 1997. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. *J. Intell. Rob. Syst.* 18, 3 (1 March 1997), 249–275.

[24] Jorge L Martínez, Javier González, Jesús Morales, Anthony Mandow, and Alfonso J García-Cerezo. 2006. Mobile robot motion estimation by 2D scan matching with genetic and iterative closest point algorithms. *J. Field Robotics* 23, 1 (1 Jan. 2006), 21–34.

[25] T Masuda, K Sakaue, and N Yokoya. 1996. Registration and integration of multiple range images for 3-D model construction. In *Proceedings of 13th International Conference on Pattern Recognition*, Vol. 1. 879–883 vol.1.

[26] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. 2011. Exploratory Landscape Analysis. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO '11)*. ACM, New York, NY, USA, 829–836.

[27] Melanie Mitchell. 1998. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA.

[28] M Mitchell, S Forrest, and J H Holland. 1992. The royal road for genetic algorithms: Fitness landscapes and GA performance. *Proceedings of the first* (1992).

[29] F Pourraz and J L Crowley. 1999. Continuity properties of the appearance manifold for mobile robot position estimation. *IEEE Workshop on Perception for Mobile âĂę* (1999).

[30] Craig Robertson and Robert B Fisher. 2002. Parallel Evolutionary Registration of Range Data. *Comput. Vis. Image Underst.* 87, 1 (1 July 2002), 39–50.

[31] Stuart C Shapiro. 1992. *Encyclopedia of Artificial Intelligence* (2 volume set edition ed.). Wiley.

[32] David A Simon. 1996. *Fast and accurate shape-based registration*. Ph.D. Dissertation. Carnegie Mellon University Pittsburgh.

[33] S Weik. 1997. Registration of 3-D partial surface models using luminance and depth information. In *Proceedings. International Conference on Recent Advances in 3-D Digital Imaging and Modeling (Cat. No.97TB100134)*. 93–100.

[34] J Yang, H Li, and Y Jia. 2013. Go-ICP: Solving 3D Registration Efficiently and Globally Optimally. In *2013 IEEE International Conference on Computer Vision*. 1457–1464.

[35] Cai Ze-Su, Hong Bing-Rong, and Li Hong. 2007. An improved polar scan matching using genetic algorithm. *Inf. Technol. J.* 6, 1 (2007), 89–95.