ELSEVIER

Artificial Intelligence 84 (1996) 339-354

Artificial Intelligence

Automatic correlation and calibration of noisy sensor readings using elite genetic algorithms ☆

R.R. Brooks *, S.S. Iyengar, J. Chen

Louisiana State University, Baton Rouge, LA 70803-4020, USA

Received February 1995; revised March 1996

Abstract

This paper explores an image processing application of optimization techniques which entails interpreting noisy sensor data. The application is a generalization of image correlation; we attempt to find the optimal gruence which matches two overlapping gray scale images corrupted with noise. Both tabu search and genetic algorithms are used to find the parameters which match the two images. A genetic algorithm approach using an elitist reproduction scheme is found to provide significantly superior results.

Keywords: Genetic algorithms; Tabu search; Sensor fusion; Noise reduction; Image matching

1. Introduction

Robotics is a field which is frequently cited when applications for machine learning algorithms are given. Booker et al. describe a fictitious machine responsible for tracking and capturing prey as one possible application for classifier systems [5]. Classifier systems have also been proposed for use in solving robot navigation problems by Zhou [18]. In addition to genetic-algorithm-based methods, the literature contains many applications of connectionist learning methods, neural networks, to robotics [8, 10, 13]. This is natural since robotics attempts to make machines which are capable of dealing with their environment. If the environment cannot be fully defined beforehand, robotic

☆ Originally submitted as a Research Note. This work was supported in part by the Office of Naval Research grant N 00014-94-1-0343, and JPL-Cal.Tech. BMA 958309 (LSU code 115-35-6105) on Advances in Distributed Sensor Fusion.

* Corresponding author. E-mail: rrb@bit.csc.lsu.edu.

PII S0004-3702(96)00012-4

devices will need some kind of learning mechanism in order to adapt to unforeseen constraints.

The task of constructing reliable robotic systems is complicated by the state of modern sensor technology. The currently available sensor devices have limited accuracy and return readings which are corrupted with noise [12, 15]. The presence of some kind of noise in sensor data is unavoidable, which severely complicates the task of interpreting sensor inputs. Decision making based on partially faulty and potentially contradictory data is one of the problems limiting the advance of robotics technology.

Machine learning algorithms, in general, derive general concepts from specific instances of data received by the system. This causes many traditional learning algorithms to be intolerant of noisy data [6]. This is not surprising. The task of inducing a concept is challenging even when data is consistent. On the other hand, machine learning programs based on the genetic programming or connectionist paradigms seem to be able to tolerate noise in the input data.

This paper explores a signal processing application of genetic algorithms which involves interpreting noisy sensor data. The problem is essentially a generalization of image correlation. The generalization is done in a way that has not been treated previously, and is suited to many real-life applications. It is especially adapted for use in "active vision" applications where observations are a part of the dynamic processing.

Applying genetic algorithms to a problem entails finding the proper presentation of the problem in terms of parameters and a fitness function [14]. In this paper, two different genetic algorithms are used to correlate imperfect sensor data. Since the application is essentially an optimization problem, a heuristic search method is also implemented for comparison. All three algorithms treat data corrupted with noise which is approximately Gaussian. The quality of the solutions found, as well as the number of iterations needed, is studied as the variance of the noise introduced to the data is increased.

The problem of finding a solution which is globally acceptable is also investigated. The genetic algorithm using an elitist reproduction scheme is found to produce the best results: both in terms of solution quality and noise tolerance.

2. Problem statement

Two sensors return two-dimensional gray scale data from the same environment. The sensors have identical geometric characteristics, return readings covering a circular region, and it is known that these readings overlap. Both sensors' readings contain noise. What is not known, however, is the relative position of the two sensors. Sensor 2 is translated and rotated by an unknown amount with relation to sensor 1.

If the size, or the contents, of the overlapping areas were known, it would be possible to perform a correlation using the contents of the overlap on the two images and find the point where they overlap directly. Since this information is unavailable, this approach is impossible.

The best way to solve this problem depends on the nature of the terrain being observed. If unique landmarks can be identified in both images then it is possible to attach the two images at those points. Depending on the number of landmarks available minor

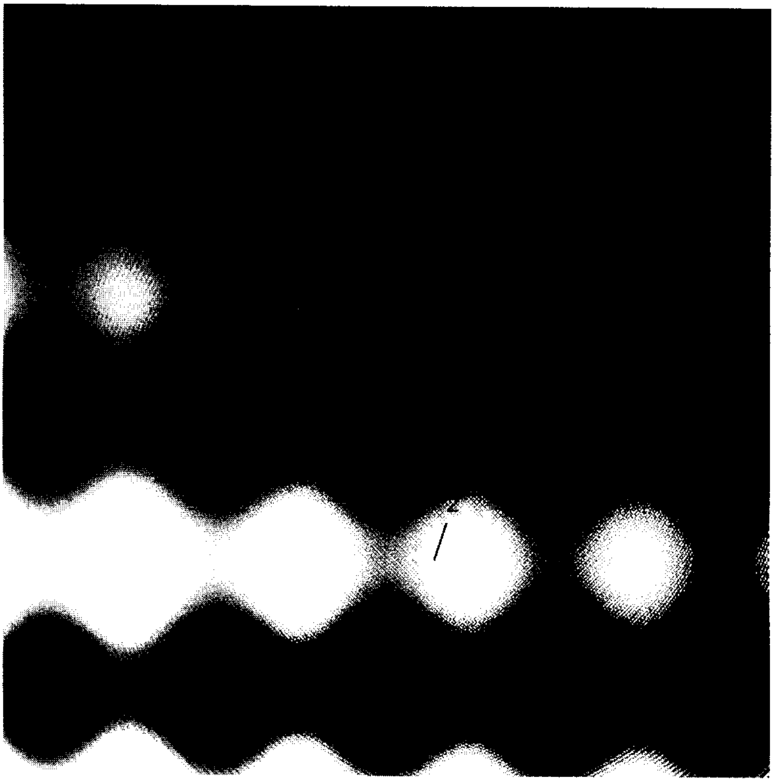

Fig. 1. View of terrain used for matching sensor readings. Position and orientation of sensors in gray.

adjustments may be needed to fit the readings exactly. It is assumed in this work that landmarks are not readily available in the sensor images.

The model which has been used to represent the terrain here has several periodic components combined with non-periodic elements. The equation used is:

$$\begin{aligned}\mathrm{terrain}(x,y) = 100.0 + \tfrac{1}{100}(&-40x + 45y - 0.003xy + 0.02x^2 - 0.01y^2 \\ &-20y\sin(\tfrac{1}{18}x) + 35y\cos(\tfrac{1}{29}y) \\ &-35\sin(\tfrac{1}{4}x - \tfrac{1}{12}y) + 12x\cos(\tfrac{1}{100}xy)).\end{aligned}$$

This equation was found through trial and error, and the result is vaguely reminiscent of mountainous terrain. This terrain model is shown in Fig. 1. It consists of a 512×512 gray scale array.

This model has been chosen since it presents two characteristics which are necessary for the problem to be solvable, but not trivially solvable. Since it has non-periodic elements, there will be a unique best match for the two sensors. The periodic elements in the model mean that this match is not obvious: an algorithm which searches for the best match will have to be capable of dealing with local minima in the search space. The effects of the local minima is aggravated by the noisy character of the sensor data.

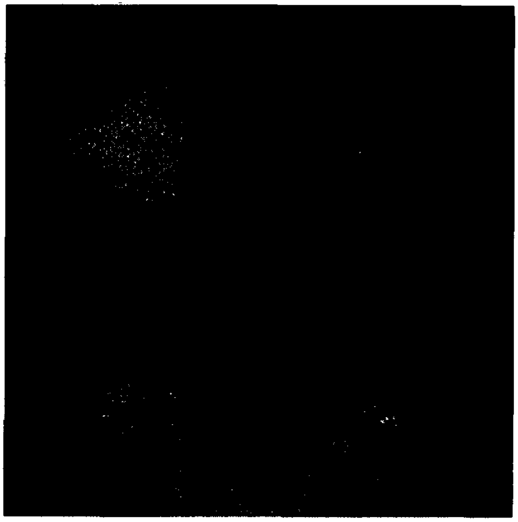

Fig. 2. Sensor 1 reading with an error variance of 33.

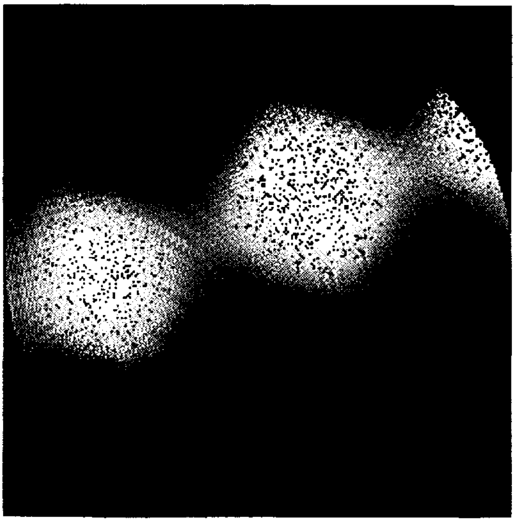

Fig. 3. Sensor 2 reading with a noise variance of 1.

A noisy sensor reading is shown in Fig. 2, which gives an example of readings from the terrain returned by sensor 1. Fig. 2 has a variance of 33. Since the gray scale used has only 256 levels, a variance of 33 obscures a large amount of information in the sensor readings. This reading is from the center of the terrain. It has not been rotated, its center is at point (256,256) using the upper left-hand corner as the origin.

Fig. 3 gives a corresponding example of sensor 2 readings used in this project. It has a noise variance of 1. Note that the relation between the readings is not intuitively obvious, and that several possible correlations exist. There is, however, one registration which is the best mapping of the sensor 2 data onto sensor 1. This reading is centered at the lower right-hand edge of the sensor 1 reading, which is point (347, 347). It has been rotated by 2.74889 radians (157.5 degrees).

Thus, the problem to be solved is: given noisy gray scale data readings from sensor 1 and sensor 2, find the optimal set of parameters (x-displacement, y-displacement, and angle of rotation) which defines the center of the sensor 2 image relative to the center of the sensor 1 image. These parameters are optimal in that they give the best mapping of sensor 2's readings to the readings from sensor 1.

3. Algorithms

Three different approaches were used to solve the problem. All of them can be presented as methods for finding a global optimization in the presence of local minima.

3.1. Fitness function

The algorithms used need to compare the quality of different sets of parameters. All the approaches used in this research were implemented using the same fitness function.

The noise which is introduced is approximately Gaussian. Gaussian noise, also known as white noise, follows a normal distribution and has an expected value of zero. The noise introduced here follows a normal distribution as much as possible, but the gray scale used has a range limited to 256 discrete values. A pixel with a gray scale of 255 cannot have a larger value due to noise since the resulting value will go beyond the gray scale, but noise can reduce the value of that pixel. A similar effect exists for pixels with a gray scale value of 0. In spite of this, the assumption was made that, over the entire intersection, there will be no appreciable bias to the noise.

The fitness function is derived by first computing the intersection between sensor 1 and sensor 2 using the parameter set to be evaluated. The gray levels of every pixel from sensor 1 in the intersection are compared with the gray level of the corresponding sensor 2 pixel. If $\mathrm{read}_1(x,y)$ is the value returned by sensor 1 at point (x,y) and $\mathrm{read}_2(x',y')$ is the reading returned by sensor 2 at point (x',y'), point (x',y') is found by reversing the translation and rotation defined by the parameters being tested. It is possible to present the difference of $\mathrm{read}_1(x,y)$ and $\mathrm{read}_2(x',y')$ as:

$$\begin{aligned}&\mathrm{read}_1(x,y) - \mathrm{read}_2(x',y')\\&\quad = (v_1(x,y) + \mathrm{noise}_1(x,y)) - (v_2(x',y') + \mathrm{noise}_2(x',y')),\end{aligned}$$

where $v_1(x,y)$ and $v_2(x',y')$ are the actual gray scale values and $\mathrm{noise}_1(x,y)$ and $\mathrm{noise}_2(x,y)$ are the noise in the sensor 1 and sensor 2 readings respectively.

This expression can be rewritten as:

$$\begin{aligned}&\mathrm{read}_1(x,y)-\mathrm{read}_2(x',y')\\&\quad=(v_1(x,y)-v_2(x',y'))+(\mathrm{noise}_1(x,y)-\mathrm{noise}_2(x',y')).\end{aligned}$$

If we square this value and sum it over the entire intersection this becomes:

$$\begin{aligned}&\sum(\mathrm{read}_1(x,y)-\mathrm{read}_2(x',y'))^2\\&\quad=\sum((v_1(x,y)-v_2(x',y'))+(\mathrm{noise}_1(x,y)-\mathrm{noise}_2(x',y')))^2.\end{aligned}$$

Note that when the parameters are correct the gray scale values $v_1(x,y)$ and $v_2(x',y')$ will be identical, and this expression becomes:

$$\sum(\mathrm{read}_1(x,y)-\mathrm{read}_2(x',y'))^2=\sum(\mathrm{noise}_1(x,y)-\mathrm{noise}_2(x',y'))^2.$$

Since all noise follows the same distribution with the same variance the expected value of this is identical for all intersections of the same area and, as such, the minimum value for the function over all intersections of a given area. Variation in this value thus consists of two parts: the difference in the gray scale values of the noise-free image, and a random factor which is distributed according to a Chi-square distribution of unknown variance. The number of degrees of freedom for the Chi-square distribution is the number of pixels in the intersection.

It is possible to have small intersections which match coincidentally. In order to favor intersections of larger area we divide by the number of pixels in the intersection squared. The fitness function thus becomes:

$$\sum(\mathrm{read}_1(x,y)-\mathrm{read}_2(x',y'))^2/(\text{number of pixels in the intersection})^2.$$

The expected value of a Chi-square function is the number of degrees of freedom, and the number of degrees of freedom in this case is equal to the number of pixels in the intersection. In the case of a perfect fit (i.e., $v_1(x,y)=v_2(x',y')$) the expected value of this function is therefore within a constant factor of:

$$1/(\text{number of pixels in the intersection}).$$

This function is the summation of the error per pixel squared over the intersection of the sensor 1 and sensor 2 readings. As shown above, the unique global minimum of this function is found when using the parameters which define the largest intersection where the gray scale values of sensor 1 are the same as the gray scale values of the translated and rotated sensor 2 reading.

In practice, this fitness function adequately reflects the quality of the answers represented by a given set of parameters, as is shown by the simulation. Note that all three algorithms used in this paper depend on this fitness function being an accurate measure of the quality of a potential answer.

Other fitness functions could exist which adequately represent the quality of potential answers. Since these functions measure the same effects as the fitness function derived in this section, the values returned by these hypothetical functions must be approximately equal to the values given by the fitness function used here. Two accurate metrics of the same phenomenon must give similar readings. For this reason, replacing the fitness function with another equally valid function would not severely affect the results of the experiments presented in this paper. The fitness function derived in this section is suited to measuring the gruence of noisy images since it has a global minimum where the non-stochastic portion of the data provides the optimal answer, and the stochastic portion of the data is represented by a consistent known statistical distribution over the entire answer space.

3.2. *Tabu search*

The first method which is used is called "tabu search". This search is often used as an alternative to simulated annealing. It is similar to simulated annealing in that it provides a method for adapting existing search heuristics to problem spaces which contain local minima.

Tabu search involves modifying an existing heuristic search by keeping a list of the nodes in the search space which were visited most recently by the search algorithm. These points then become "tabu" for the algorithm, where "tabu" means that these points are not revisited as long as they are on the list. This simple modification will allow a search algorithm to eventually climb out of shallow local minima in the search space. It requires less computation than simulated annealing, while providing roughly equivalent results. Several questions are being studied as to how to optimize tabu searches, such as the optimal length for the tabu list [3], and methods for implementing parallel searches [17]. Our implementation uses a tabu list that is considered infinite.

The implementation of the tabu search used here relies on a "greedy" heuristic and starts with all parameters set to zero, i.e., there is neither translation nor relative rotation between the two readings. The search can move one pixel in each x-direction, one pixel in each y-direction, or by rotating plus or minus one degree. The algorithm evaluates how well the two sensor readings would match, using the fitness function, for each of these six possibilities. Naturally, the search chooses to visit the next node in the direction with the minimum value for the fitness function. When one of these parameter sets is visited it is placed on the tabu list.

Values on the tabu list are disqualified for consideration in the future. Should the search arrive at a neighboring point later, the fitness function value given to parameter sets on the tabu list is set to a very large value.

As each parameter set is visited by the search, the value attributed to it by the fitness function is compared to the parameter set already visited with the smallest value for the fitness function up to this point. If the value is smaller the parameter set now becomes the best fit found.

Each iteration of the algorithm is run on a new instance of the noisy sensor 1 and 2 readings.

It is impossible to find a clear stopping criterion for this algorithm since the only way to be sure that the global minimum for the fitness function has been found is through an exhaustive search of the search space. This study was done by comparing the results from a given number of iterations of the tabu search with the results obtained by performing the same number of iterations with the genetic algorithms.

3.3. Genetic algorithms

Genetic algorithms are a computational paradigm which has been implemented successfully as a solution to many optimization problems. In this paradigm, possible answers to the problem are stored as strings. A large set of these strings then forms a gene pool. The quality of these possible answers can be evaluated using a fitness function. The relative quality of the answers provided by the strings is used to create a new generation of strings, where the contents of strings providing answers of high quality are more likely to continue into the next generation. A general discussion of genetic algorithms can be found in [16].

A large number of strategies exist for determining the contents of a new generation. This project used two different strategies to contrast their effectiveness.

The strings used to characterize the problem here consisted of the same parameters used for the tabu search: the offset in the x-direction, the offset in the y-direction, and the angle of rotation. Resolution in the x- and y-directions is one pixel. Angles vary with a resolution of one degree. The fitness function used has also been described above.

The gene pools consisted of 150 sets of parameters which were initialized with random values at the start of the program.

The two different genetic algorithms which were used differed only in their reproduction strategies.

Classic

The first strategy has been described by Holland [11]. Each string in the gene pool is evaluated by the fitness function. Based on the quality of the answer represented by the string it is assigned a probability of being chosen for the pool of strings used to produce the next generation. Those with better answers being more likely to be chosen. In our implementation the values of the fitness functions for all members of the gene pool were summed. The value of the fitness function for each member of the pool was then divided by the sum giving the probability of that string being used for mating. A mating pool is then constructed by choosing strings at random from the gene pool following the probability distribution derived.

The new generation is then formed by mixing the elements of two strings in the mating pool chosen at random. This is generally called crossover. In both genetic algorithms used in this study a crossover probability of one was used.

Since our strings consisted of three elements, the result of this mixing always consisted of two elements from one parent and one from the other. Which element was switched was chosen at random and all three were equally likely.

A certain amount of mutation exists in the system, where one element at random is replaced by a random value. In our implementation mutation occurs once with every 700th string processed.

Elite

The second strategy applied has been described in a recent paper by Bean [4]. This strategy is described as elitist since 20% of the strings with the best fitness function values are copied directly into the next generation.

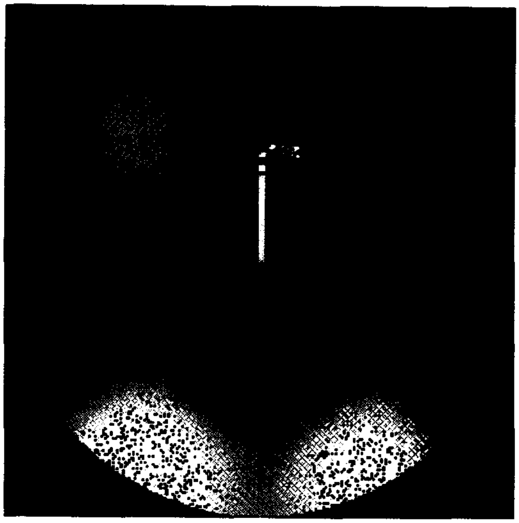

Fig. 4. Path taken by 325 iterations of a tabu search algorithm in searching for match with variance of 1.

In addition to this, in our implementation, 3% of the strings for the next generation are the result of random mutations. In this implementation the random mutations were strings where all three elements were chosen at random.

The rest of the new generation is formed by performing crossover between random strings in the current generation. The choice is done entirely at random, no weighting based on the quality of the string is performed.

Bean reports that this strategy has been found to be stable experimentally. Its implementation is straightforward.

4. Results

All three algorithms were applied to the same sensor readings. Each iteration of the algorithm was performed against a new sensor reading. All sensor 1 and 2 readings covered the same region, but new noise values were introduced each time. This resembles the situation which would be found in a dynamic environment. Using sensor readings which covered regions that change over time should make the problem easier to solve since local minima in the search space would tend to be transient.

The results which follow were obtained by using increasing values for the variance of the Gaussian noise. Comparison between the various methods was based on the value of the fitness function for the best reading found.

4.1. Tabu search

Fig. 4 shows the path taken by the tabu search algorithm when searching for an optimal match between the two sensor readings with a noise variance of 1. The search

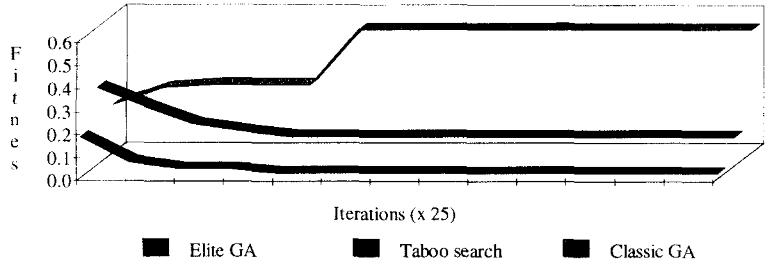

Fig. 5. Fitness function results.

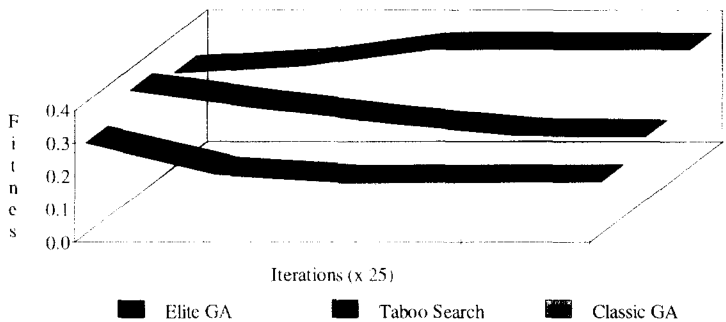

Fig. 6. Fitness function results.

started at the middle of the sensor 1 reading. Note that the correct answer would have been at the bottom right-hand edge of the sensor 2 reading. It is interesting to note that even in the presence of noise with a variance of 90 which is strong enough to obscure most of the information contained in the picture, the search took approximately the same path as with very little noise.

The charts in Figs. 5 and 6, which show the value of the best parameter set found by the search, confirm this observation. The algorithm tended to move towards locally optimal values and performed in a stable manner.

Unfortunately the answer found was not close to the globally optimal values for the parameters.

4.2. Classic reproduction scheme for genetic algorithms

Fig. 7 shows the values of the gene pool after one generation of the genetic algorithm using the classic reproduction scheme. Fig. 8 shows the same scenario after 100 generations. Note that the parameter set at the lower right-hand corner which is closest to the correct answer is no longer present.

The gene pool values found after a number of generations with variance values of 50 and 90, show that even in the presence of noise the values contained in the gene pool tend to converge. Unfortunately the convergence is not towards the globally optimal values.

Fig. 7. Gene pool values after one generation of the genetic algorithm using classis reproduction.

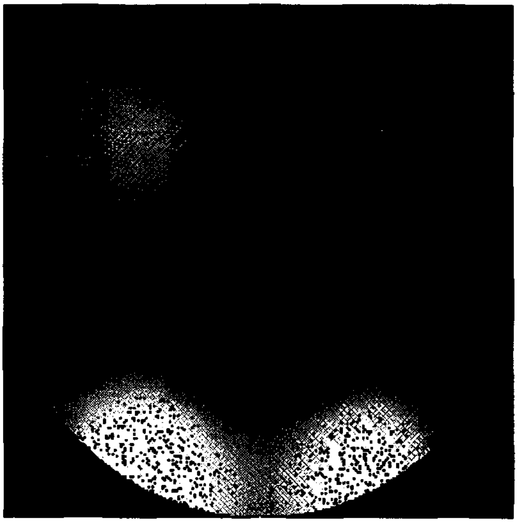

Fig. 8. Results of classic reproduction GA after 100 generations with a variance of 1.

Figs. 5 and 6 show the relationship of the fitness function value of the best parameter set to the number of generations used by the algorithm. Oddly enough, the value tends to increase instead of decrease as would be expected. The algorithm tended to remove the parameter sets with extreme values for the fitness function. This converged towards values which were stable but far from the global optimum.

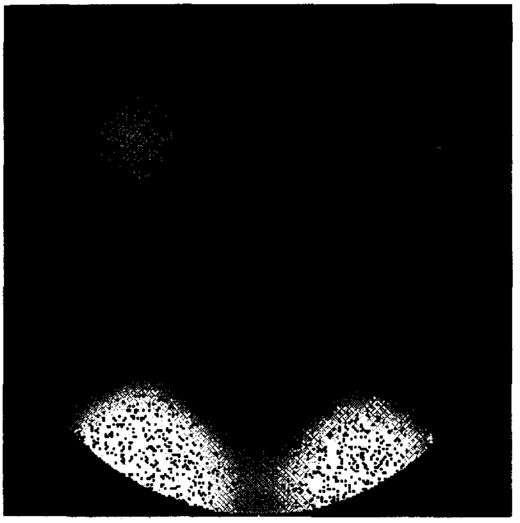

Fig. 9. Gene pool after 150 generations of the genetic algorithm with elite reproduction and variance 1.

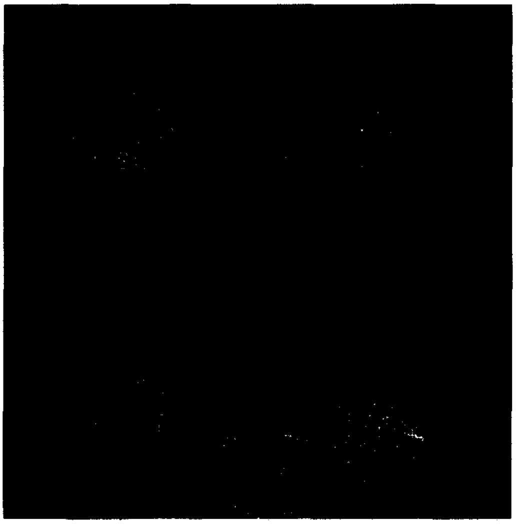

Fig. 10. After 100 generations with variance 50, the gene pool found by elite reproduction scheme.

4.3. Genetic algorithm with elite reproduction scheme

Fig. 9 shows the gene pool found by the elite reproduction scheme after 150 generations. Notice that Figs. 9 and 10 both contain values very close to the globally optimal value. The genetic algorithm with the elite reproduction scheme tended to converge towards the globally optimal value even in the presence of moderate noise.

Table 1

Variance	x-displacement	y-displacement	Rotation
1	89	91	2.74744
10	92	92	0
20	91	91	2.74744
30	89	89	2.74744
50	86	−18	2.79768
70	−48	6	6.02138
90	0	5	1.23297

The results after 75 generations with the variance of the error set to 90 show that the amount of information in the image has been severely compromised, and that no parameter sets are found near the globally optimal answer.

The graphs in Figs. 5 and 6 verify that this algorithm tended to converge rapidly towards very good solutions to the problem. In fact, the shapes of the graphs are surprisingly similar considering the differences in the images they are treating.

In spite of the fact that the algorithm converged towards good solutions even in the presence of overwhelming amounts of noise, there was a limit to its ability to find the globally optimal solution. Note that the globally optimal parameter values are: x-displacement = 91, y-displacement = 91, rotation = 2.74889 radians.

The values found by the algorithm are shown in Table 1. These values show that the algorithm does not always find the globally optimal values, but it tends to do a good job even in the presence of moderate amounts of noise. However, once the noise reaches a point where it obscures too much of the information present in the image it no longer locates the optimal values.

It is also worth noting that the quality of the answers found is not strictly a function of the noise variance. This is a consequence of the stochastic nature of genetic algorithms. Since the original gene pool and mutations are chosen at random, the quality of the answers found by the algorithm is also partially random in nature. Genetic algorithms are based on comparing the results of a number of random choices. The answers found by the elitist reproduction scheme show this strategy can be effective in finding reasonable answers to combinatorial optimization problems. On the other hand, the results found are non-deterministic in nature and the quality of the algorithm's results depend partially on the random nature of this method.

5. Conclusion

The problem posed was to find ways of automatically calibrating two noisy sensors using optimization methods. This has been shown to be possible as long as the noise is held within certain limits.

Several attempts have been made to solve this type of problem, this is the first attempt which matches noisy gray scale images. Several algorithms exist for roughly equivalent problems, the other algorithms are more specific in that they assume that the image consists of a small number of distinct features which can be matched [1,7], that specific shapes are to be matched [9], or that the relative displacement is small [2]. This paper assumes none of those restrictions, the tests were run using gray scale images with many possible approximate matches and a large displacement. The approach can be used on any arbitrary gray scale image.

It is necessary to register the readings from sensor 2 with the readings from sensor 1. This can be done by finding three parameters: the offset of the two sensors in the x-direction, the offset in the y-direction, and the angle of rotation between them.

Three methods have been attempted: tabu search, genetic algorithms using a classical reproduction scheme, and genetic algorithms using an elitist reproduction scheme. Of the three, the genetic algorithm using the elitist reproduction scheme has tended to produce the best results. It often found even close to globally optimal results.

All three approaches are able to deal with data containing reasonable amounts of noise. The genetic algorithm using the elitist reproduction scheme was able to continue to find answers with two out of three parameters very close to the optimum even when the noise variance was set to 50. With possible values ranging only from 0 to 255 noise at that level severely distorts the actual image. It is therefore safe to say that the genetic algorithm paradigm when implemented with an elitist reproduction scheme is very tolerant to noise.

Tabu search is more sensitive to local minima than the genetic algorithms, since its searching mechanism only considers points in the search space which are in its immediate vicinity. It quickly converges to the local minima. It may take a prohibitively long time, however, for the search to climb its way out of the local minima. Genetic algorithms have the advantage that they can simultaneously process many different possible answers and are not constrained to looking for local minima.

The elitist reproduction scheme preserves the members of the gene pool which have the best performance. This contrasts with the classical scheme where the quality of the best answers may degrade as the algorithm progresses. The elitist scheme keeps the best answers from preceding generations forcing the quality of the best answer to be monotone increasing, as shown in Figs. 5 and 6. This is why the elitist scheme is most suited to this application where a single best answer is sought. Fig. 10 shows an elitist gene pool with a few near optimal answers and several far from optimal answers.

The classical reproduction scheme produces a new generation by mixing genes between members of the current gene pool. Higher quality members of the gene pool are more likely to be chosen for reproduction. Genetic algorithms that use the classical scheme produce a new generation of higher average quality than the previous generation. The average quality of answers using the elitist reproduction scheme described in this paper may decrease from one generation to the next. Figs. 11 and 12 illustrate this point by comparing the average error of gene pools derived by this application. As the process continues, the classical gene pool will contain a diverse number of answers, and the average quality of answers in the gene pool tends to increase. Fig. 8 illustrates a gene pool containing a number of good possible answers which are unfortunately far from the optimal answer. When using the classical scheme a near optimal answer is unlikely to be carried over unchanged into the next generation. For this reason, the quality of

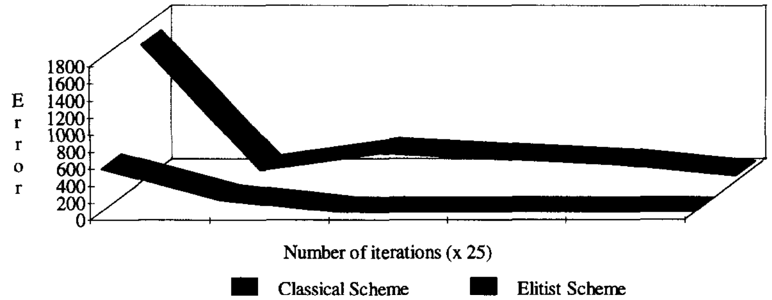

Fig. 11. Average answer quality.

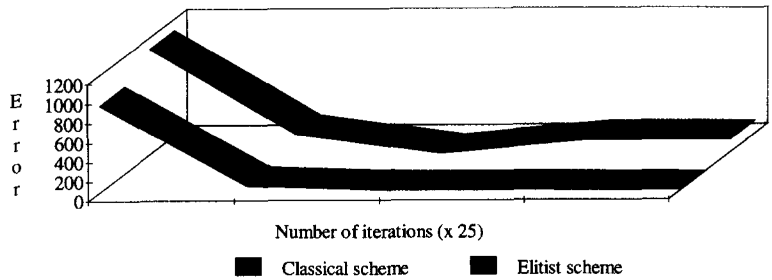

Fig. 12. Average answer quality.

the best answer found by the classical scheme may decrease from one generation to the next, as shown in Figs. 5 and 6. For these reasons the elitist reproduction scheme provides better results than the classical reproduction scheme for this application.

Tests have been made using terrain models other than the equation used for Fig. 1. The results presented in this paper are typical for all terrain models tested. The terrain models tested were all qualitatively similar in that they contained both periodic and non-periodic elements. Removing the periodic components of the terrain modifies the original problem, changing the nature of the problem space being searched. Gruences for a terrain model with little or no periodicity are much easier to find and deterministic search algorithms could then be used to solve the problem posed in a straightforward manner.

The problem posed has many possible applications. It is also worth repeating that the problem would be easier to solve in a dynamic environment. If the image varies as well as the noise, local minima would tend to be transient, in which case, this approach would be especially relevant to active vision research.

Acknowledgements

The authors wish to express their gratitude to the anonymous reviewers whose comments greatly improved the presentation of this report.

References

[1] H.S. Baird, *Model-Based Image Matching Using Location* (MIT Press, Cambridge, MA, 1985).

[2] H.G. Barrow, J.M. Tennenbaum, R.C. Bolles and H.C. Wolf, Parametric correspondence and chamfer matching: two new techniques for image matching, in: *Proceedings IJCAI-77*, Cambridge, MA (1977) 659–663.

[3] R. Battiti and G. Tecchioli, The reactive tabu search, *ORSA J. Comput.* **6** (1994) 126–140.

[4] J.C. Bean, Genetic algorithms and random keys for sequencing and optimization, *ORSA J. Comput.* **6** (1994) 154–160.

[5] L.B. Booker, D.E. Goldberg and J.H. Holland, Classifier systems and genetic algorithms, in: J. Carbonell, ed., *Machine Learning* (MIT Press, Cambridge, MA, 1990) 234-282.

[6] J. Carbonell, Introduction: paradigms for machine learning, in: J. Carbonell, ed., *Machine Learning* (MIT Press, Cambridge, MA, 1990) 3–4.

[7] I.J. Cox and J.B. Kruskal, On the congruence of noisy images to line segment models, in: *Proceedings Second International Conference on Computer Vision*, Tampa, FL (1988) 252–258.

[8] E. Davalo and P. Naim, *Des Réseaux de Neurones* (Eyrolles Editions, 1989) 188–189.

[9] L.S. Davis, Shape matching using relaxation techniques, *IEEE Trans. Pattern Anal. Mach. Intell.* **1** (1979) 60–72.

[10] S. Gulati, J. Barhen and S.S. Iyengar, Neurocomputing formalisms for computational learning and machine intelligence, in: *Advances in Computers* (Academic Press, New York, 1991).

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

[12] S.A. Hovanessian, *Introduction to Sensor Systems* (Artech House, 1988).

[13] T.L. Huntsberger, Data fusion: a neural networks implementation in data fusion, in: Abidi and Gonzales, eds., *Data Fusion in Robotics and Machine Intelligence* (Academic Press, New York, 1992) 507–535.

[14] S.S. Iyengar, L. Prasad and D. Morton, in: S.S. Iyengar, ed., *Structure of Genetic Algorithms in Biological Systems* (CRC Press, 1992).

[15] R.C. Luo and M.G. Kay, Data fusion and sensor integration: state-of-the-art 1990's, in: Abidi and Gonzales, eds., *Data Fusion in Robotics and Machine Intelligence* (Academic Press, New York, 1992) 7–136.

[16] M. Srinivas and L.M. Patnaik, Genetic algorithms: a survey, *IEEE Comput.* **27** (6) (1994) 17–26.

[17] E. Taillard, Parallel taboo search techniques for the job shop scheduling problem, *ORSA J. Comput.* **6** (1994) 108–117.

[18] H.H. Zhou, CSM: a computational model of cumulative learning, in: *Machine Learning* **5** (Kluwer Academic Publishers, Dordrecht, 1990) 383–406.