

Gradient-based genetic algorithms in image registration

Igor V. Maslov^a, Izidor Gertner^b

^aDept. of Computer Science, City Univ. of New York/Graduate Center

^bDept. of Computer Science, City Univ. of New York/City College

ABSTRACT

Two modifications of Genetic algorithm (GA) are proposed that employ gradient analysis of the fitness function and are integrated with the main genetic procedure. Combination of the relative weighted error factor and adaptive size of the mutation pool accelerates convergence of the iterative process and indicates when the global optimum solution is found. Local gradient correction of the initial pool during iterations refines the search procedure. Computational experiments show that both modifications can increase efficiency of GA when they are applied to an image registration problem.

Keywords: Image registration, multi-sensor system, affine transformations, genetic algorithms, gradient search

1. INTRODUCTION

Image registration is an important part of multi-sensor image processing. Readings from the same object, obtained from different sensors, should be placed into a common coordinate system. Registration problems for a set of 2-dimensional (2D) images $\{S_i\}$ can be formulated as a problem of finding correct mappings from an image S_i to a reference image S_0 . A search in a space of all possible affine transformations for an image S_i enables one to find a vector of parameters (DX, DY, θ) that would minimize the difference between S_i and S_0 in the area of their intersection. In this area, DX and DY become displacements, and θ is a rotation of S_i relative to S_0 . Among the methods that have been applied to solve this problem, Genetic algorithm (GA) has proven to be a potentially effective iterative procedure, in cases where parameter space is very large¹. GA is also well suited for parallel processing.

The potential power of Genetic algorithm as a computational basis for image registration is weakened by the following factors:

- Firstly, the number of iterations required to achieve a reasonably good solution is usually large.
- Secondly, as with some other heuristic algorithms, GA does not provide a clear indication as to when the optimum solution is found and the iterative process can be terminated. GA can find solutions that are local minima and remain there for a number of iterations while on its way towards the global minimum point in the parameter space.
- Thirdly, even if the global minimum point is found, the final solution may require refinement. Randomly drawing parameter vectors from the parameter space and performing genetic operations on these vectors, GA builds a discrete model of the original continuous space. The granularity of this model will affect the accuracy of the result. Refining a solution obtained with GA may require employment of more sensitive methods².

This paper proposes two modifications of Genetic procedure applied to image registration:

- Combination of the relative weighted error factor and adaptive size of the mutation pool.
- Local gradient correction of the replication pool.

These modifications are based on gradient analysis of fitness function and are seamlessly integrated into a genetic iterative search. The remaining sections of the paper discuss the details of these modifications. Section 2 provides an overview of test problem and discusses proposed gradient-based modifications as compared to the original algorithm. Section 3 analyzes the computational efficiency of these modifications. And, finally, Section 4 presents the conclusions regarding the proposed modifications.

2. GRADIENT-BASED MODIFICATIONS OF GENETIC ITERATIVE PROCEDURE

2.1. Test problem

Genetic algorithm exploits an iterative procedure similar to the process of natural evolution. Iterations of GA correspond to generations of evolution. A set of potential solutions, called a gene pool or a chromosome, is applied to a problem at every iteration step. Gene pool for the next iteration (or generation) is obtained using three operations on current pool – replication,

crossover, and mutation. Fitness function measures the relative quality of each gene participating in these operations. Only the fittest genes can be replicated. In relation to image processing, vector of parameters (DX, DY, θ) is used as a gene, and a set of these vectors forms a chromosome that participates in an iterative process.

Computational experiments are based on the model of sensor readings explored by Richard R. Brooks and S.S. Iyengar¹ and shown in Figure 1. Two sensors **S1** and **S2** contain 256×256 gray scale noisy images that overlap. The second image **S2** is displaced by $DX = 91$, $DY = 91$ from the center of the first (reference) image **S1**, and rotated by $\theta = 2.74889$ radians. Figure 2 shows the exact solution with overlap area for both images.

Fitness function that is used in selection of the fitter genes is computed as the squared residual error of gray levels of images over the area of their intersection (overlap):

$$\varepsilon = \frac{\iint_S (g1(x, y) - g2(x, y))^2}{S}, \quad (1)$$

where $g1(x, y)$ and $g2(x, y)$ are the gray levels of images **S1** and **S2**, and S is the area of their intersection. In discrete form, expression (1) becomes:

$$\varepsilon = \frac{\sum_N (g1(x, y) - g2(x, y))^2}{N^2}, \quad (2)$$

where N is the number of pixels in the intersection area of **S1** and **S2**.

A weighted Euclidean distance between an approximate and the exact solution is used throughout this paper to measure the accuracy of the obtained solution:

$$\delta = \sqrt{\left(\frac{\Delta(DX)}{DX}\right)^2 + \left(\frac{\Delta(DY)}{DY}\right)^2 + \left(\frac{\Delta(\theta)}{\theta}\right)^2}, \quad (3)$$

where $\Delta(DX)$, $\Delta(DY)$, and $\Delta(\theta)$ are the differences between the approximate and the exact values of DX , DY , and θ .

Brooks and Iyengar found that Genetic elitist algorithm has superior performance over classical GA. In their experiments with the elitist scheme, the next generation gene pool is formed as follows:

- 20% of genes with the best values of fitness function are replicated, i.e. directly copied from the current generation of gene population,
- 3% of the pool is subject to random mutations,
- the rest of the pool is formed by random crossover between the genes in the current population.

As one can see in Figure 3, convergence to the optimum solution is a relatively slow process. It reaches the global minimum point ($DX = 89$, $DY = 84$, $\theta = 2.78081$) on iteration 54. The value of the minimum error at this point is $\varepsilon = 0.062$, and the weighted Euclidean distance from the exact solution is $\delta = 0.086$.

The process falls into the local minimum point ($DX = -34$, $DY = 73$, $\theta = 2.78081$) on iteration 9, with the minimum error $\varepsilon = 0.14062$. It stays at approximately the same point until iteration 45. The overlap between the two images at this point (Figure 4) is very different from the exact solution shown in Figure 2. Algorithm does not provide a clear indication that this is only a local minimum and we have to continue the search. A detailed analysis of changes in the gene pool between iterations helps explain why a stall occurs. Figure 5 shows a graph of fitness function for the gene pool across several iterations. Once a locally optimal parameter vector finds its way into the replication pool, it tends to propagate through the entire pool as a result of crossover operations. Only mutations of the reserved part of the pool help keep its variety and prevent the search from being trapped at the local minimum point.

2.2. Adaptive size of mutation pool

A reasonable remedy that may prevent the process from stalling at a local minimum may be increasing the amount of mutations to maintain the variability of the gene pool. The size of the mutation pool can be controlled by the rate at which the gradient of the fitness (error) function is decreasing from iteration to iteration.

In order to find a relationship between the rate of convergence of the error function and the size of the mutation pool, a weighted error factor F_{we} is introduced as follows. For every iteration, the sum of the error gradients in the replication pool is computed and weighted over the minimum error value on the iteration:

$$F_{we} = \frac{\sum_R \nabla E}{E_{\min}}, \quad (4)$$

where ∇E is the gradient of the error for each parameter vector in current replication pool, E_{\min} is the minimum error value for current iteration, and R is the size of the replication pool.

The value of F_{we} for iteration 0 (initial iteration) is chosen as the base value. Then the relative value of F_{wer} is computed for every subsequent iteration as follows:

$$F_{wer} = (F_{we} \text{ for iteration 0}) / (F_{we} \text{ for current iteration}) . \quad (5)$$

The size of the mutation pool for the next generation is increased by the factor F_{wer} . As the error function is flattening from iteration to iteration, the value of F_{wer} and, consequently, the size of the mutation pool is growing. A larger number of new genes are introduced into the gene population. This process helps the algorithm to climb out from local minimum points and continue its search for the global optimum solution.

In the alternative approach, the sum of errors in the replication pool is used as the weight coefficient to compute F_{we} :

$$F_{we} = \frac{\sum_R \nabla E}{\sum_R E} . \quad (6)$$

Computational results show, however, that computing F_{we} according to formula (4) provides superior performance over (6).

2.3. Local gradient correction of replication pool

The process of discretization of the parameter space sets a limit on finding the closest approximation to the exact optimum solution. Genetic operations used to form the new generation of chromosomes are applied to discrete values that are randomly drawn from the parameter space. In order to refine the search while moving from iteration to iteration, an additional operation of correction is introduced as follows. On every iteration, all parameter vectors (genes) in the replication pool are considered for possible correction. Each vector is compared with 6 neighbors drawn from the 3-dimensional (3D) parameter sphere of small radius R , centered at the initial vector. The gradient of the fitness function is then used to measure the fitness of each neighbor:

$$\nabla \varepsilon = (\varepsilon_i(x, y) - \varepsilon_0(x, y)) / (\text{distance between the } i\text{-th and the initial vector}), \quad (7)$$

where $\varepsilon_i(x, y)$ and $\varepsilon_0(x, y)$ are the fitness values for the i -th and the initial vector, respectively. If a neighbor is found that has the lowest negative value of $\nabla \varepsilon$, it replaces the initial vector.

Experimental results presented in the next section show that a combination of local corrections of the replication pool, together with the adaptive size of the mutation pool, improves convergence of the iterative procedure and increases computational efficiency of Genetic algorithm.

3. ANALYSIS AND COMPARISON OF RESULTS

3.1. Adaptive size of mutation pool

In the first series of computational experiments the weighted error factor F_{we} according to formula (4) was applied, in order to control the size of the mutation pool. As in the original version of the elitist algorithm (Figure 3), the process goes through a local minimum on iterations 6 – 15, with parameter vector ($DX = 109, DY = 88, \theta = 2.78081$) -- see Figure 6. However, the value of F_{wer} does not change significantly (maximum value $F_{wer} = 3.73302$ on iteration 15) in this iteration range, as compared to its initial value $F_{wer} = 1$ (Figure 7). At the same time, the value of F_{wer} grows exponentially from iteration 27, with the sum of the error gradients falling to 0. This is an indication that the global minimum point has been reached and the search can be terminated with the optimum parameter vector ($DX = 95, DY = 88, \theta = 2.78081$). The minimum error at this point is $\epsilon = 0.06028$, and the weighted Euclidean distance from the exact solution is $\delta = 0.056$.

The above result provides a criterion for indicating when a search should terminate. As opposed to monitoring an overall convergence of error function from iteration to iteration, a relative weighted error factor F_{wer} is computed and compared for every iteration. At the point where the factor begins to grow exponentially, the search may stop, because the global minimum point has been found. Although this strategy provides the best final result, it may be too restrictive and computationally expensive. Often the exact solution is not needed and an approximation would suffice. The requirement of finding the point with the exponential growth of the factor F_{wer} may be relaxed, and a reasonable threshold value may be accepted. If F_{wer} exceeds the threshold, the found solution is considered satisfactory and the search may discontinue.

Another alternative to computing weighted error factor with formula (4) is to use the sum of errors in the replication pool as the weight coefficient, as shown in formula (6). Figures 8 and 9 show, however, that the process requires a larger number of iterations to reach the final optimum point. The factor F_{wer} grows slowly until it reaches iteration 51, where it begins to grow exponentially indicating that the optimum solution has been found. Although a good parameter vector ($DX = 89, DY = 94, \theta = 2.74396$) with $\epsilon = 0.04702$ and $\delta = 0.04$ is found on iteration 33, the search cannot be stopped until after iteration 51. This result shows that using the minimum error value provides better performance than using the sum of errors as the weight coefficient.

An attempt to extend computation of the weighted error factor F_{we} to the entire gene pool instead of its replication part does not provide satisfactory results (Figure 10 and Figure 11). The relative factor F_{wer} may no longer be considered a monotonic function of error. It now has three extremes between iterations 0 and 48. Only one of them (on iteration 48) corresponds to the global minimum point with the parameter vector ($DX = 91, DY = 93, \theta = 2.75134$), error $\epsilon = 0.038$, and distance $\delta = 0.022$. Therefore, computing the factor F_{we} over the entire gene pool cannot provide a good indication as to when the optimum solution has been found and the search can be stopped.

Figure 12 shows that simply using F_{wer} as an indication to terminate the search process, without corresponding increase of the size of the mutation pool, may result in finding a local instead of the global minimum solution. The factor F_{wer} grows exponentially on iteration 27 at the local minimum point ($DX = -100, DY = 123, \theta = 5.84311$) with error $\epsilon = 0.08973$.

Experiments with different strategies that can determine when the global optimum solution is found and the search can be terminated show that the best strategy is to use the relative weighted error factor F_{wer} according to (4) and (5) in combination with the adaptive size of the mutation pool.

3.2. Local gradient correction of replication pool

The next set of computational experiments aims to find the best strategy for applying a local correction to the replication pool during the iterative process. By using the gradient of the fitness function, as shown in formula (7), we can refine the search and obtain a better approximate solution.

We start with looking at the nearest neighbors of every parameter vector in the replication pool that change the value of each parameter by $\Delta = +/-1$. This corresponds to a 3D parameter sphere with the radius $R = 1$. Altogether, the 6 nearest neighbors (changing DX, DY , and θ by Δ) are examined. Their fitness is compared to the fitness of the initial parameter vector. The vector that has the lowest value of the fitness function (the fittest vector) is chosen to replace the initial vector in the replication pool. Although, intuitively, this approach seems to have to improve solution, results show that it does not have advantage in terms of accuracy over simply varying the size of the mutation pool. The final result ($DX = 96, DY = 86, \theta = 2.83317$) is even less accurate (error $\epsilon = 0.0608$ and distance $\delta = 0.114$) than the result without correction. At the same

time, it confirms that the factor \mathbf{F}_{wer} can be used as the criterion for terminating the search at the point of the global optimum – it grows exponentially from iteration 33 where the global minimum point has been found (see Figure 13).

Although parameter vectors are not very sensitive to small variations of their values, the corrected pool provides a smoother convergence. In order to take advantage of this feature and simultaneously accelerate convergence, the strategy for selecting the fittest neighbor has been slightly changed. The radius of the parameter sphere has been increased to $R = 3$. Each parameter vector is compared now with 6 neighbors that are randomly drawn from within the sphere. The neighbor that has the lowest negative value of the gradient of error $\nabla\epsilon$ replaces the initial parameter vector. Figure 14 shows significant improvement of convergence. A good solution ($DX = 94$, $DY = 88$, $\theta = 2.80094$) with error $\epsilon = 0.05674$ and distance $\delta = 0.05$ is found on iteration 12. The factor \mathbf{F}_{wer} grows exponentially from iteration 36 and indicates that the point ($DX = 91$, $DY = 93$, $\theta = 2.75192$) is the global minimum. This is the closest to the exact solution point that has error $\epsilon = 0.03796$ and distance from the exact solution $\delta = 0.022$.

An increase of the radius R of the parameter sphere from 3 to 6 has not improved convergence. On the contrary, it took slightly longer to reach similar to $R = 3$ result. The search has found a good point ($DX = 91$, $DY = 86$, $\theta = 2.78386$) on iteration 18 with error $\epsilon = 0.05809$ and distance $\delta = 0.056$. The exponential growth of the factor \mathbf{F}_{wer} , however, has not started until after iteration 48 (Figure 15), where the global minimum solution ($DX = 91$, $DY = 93$, $\theta = 2.75242$) has been found with error $\epsilon = 0.0387$ and distance $\delta = 0.022$.

Experimental results for different size of the parameter sphere show that there exists an optimum value of the radius R that can accelerate convergence with a local gradient search. Lower than optimum values of R do not significantly affect current values of parameters. Larger than optimum values of R may have a radical affect and diminish the evolutionary character of the iterative process.

4. CONCLUSION

The paper investigates two modifications of Genetic algorithm that employ gradient analysis of the fitness function and are seamlessly integrated with the main iterative genetic procedure. Computational experiments show that both modifications may increase efficiency of GA when they are applied to an image registration problem.

1. Combination of the relative weighted error factor \mathbf{F}_{wer} and the adaptive size of the mutation pool accelerates convergence of the iterative process and indicates when the global optimum solution is found. The factor \mathbf{F}_{wer} is computed for each iteration according to formulae (4) and (5). The size of the mutation pool for the next generation of the gene population is increased by the computed factor. This strategy helps avoid stalling at local minima and improves convergence.
2. Local correction of the initial pool during iterations refines the search procedure. Every parameter vector in the replication pool is compared with its neighbors randomly drawn from within the 3D parameter sphere of small radius centered at the initial vector. If a neighbor is found that has the lowest negative value of the gradient of the fitness function, it replaces the initial vector. This approach helps refine the search and obtain more accurate result.

REFERENCES

1. Richard R. Brooks and S. S. Iyengar, *Multi-sensor fusion: Fundamentals and Applications with software*, Prentice Hall, New York, 1998.
2. Izidor Gertner and Igor Maslov, "Heuristic approach to image registration", *Proc. of SPIE*, Vol. 4050, pp. 238-244, 2000.