Image Noise Elimination Using Evolutionary Algorithm

Dr. Matheel Emad AL-Deen Abdulmunim*

Abstract

Evolutionary algorithms like Genetic Algorithms (GA) and Genetic programming (GP) are domain-independent problem solving approaches in which computer programs are evolved to solve, or approximately solve, problems. Learning methods for image denoising consist of minimizing a functional, which incorporates both the data and some penalty term. Choosing the penalty term to involve the total variation of the image has the advantage of cleaning speckles without smoothing out the edges. In this paper, our goal is to investigate the use of genetic algorithms to minimize the functional.

ازالة التشويش من الصور باستخدام خوارزميات التطور

الخلاصة

خوارزميات التطور مثل الخوارزميات الجينية والبرمجة الجينية هـي أسـاليب لحـل المشاكل الغير المعتمدة على مجال معين بحيث إن برامج الحاسبة توضع لحل المشاكل. طـرق التعلم لرفع التشويش من الصور يضم تقليل فعال والذي يضم البيانات وبعض صـيغ الجـزاء. اختيار صيغ الجزاء لتطبيقها على مجموع التغير للصورة أعطى ايجابيات لأزالة التشوش مـن نوع سبيكلص دون تنعيم الحواف أو إخفائها.الهدف من هذا البحـث هـو تحقيـق اسـتخدام الخوارزمية الجينية لتقليل التشويش بشكل فعال.

Introduction:

Genetic Algorithm (GA) is based on the Darwinian principle of reproduction and survival of the fittest and is analogous to naturally occurring genetic operations such as crossover and mutation [1].

GA has been used in a wide range of applications and has succeeded in solving many problems such as prediction and classification, image and signal processing, optimization, financial trading, robots and autonomous agents, artificial life, and neural networks [2].

In the last decades, the field of image processing became more and more attractive, sustained by the continues advance of electrical and computer engineering. The increasing of the computing (processing) power allows considering successfully various applications. The typical image processing system consists of several different building blocks that perform distinctive tasks.

Among them, the noise removal is one of the most common encountered processing steps. In the same time, the information engineering introduced new concepts and techniques, has among them, the evolutionary computing emerged as a promising solution for several problems. The evolutionary computing domain consists of a set of nature-inspired algorithms and paradigms, as the neural networks and the GAs.

Many researchers investigated the applications of GAs in image (or

644

signal) processing, highlighting the benefits obtained by their use in increasing some quality measures or an adaptive behavior of different classical methods. This contribution investigates the use of GAs in image denoising [3].

This paper proposes a computational effective method for the genetic based noise removal that uses a model-free approach (no assumptions being made on the noise distribution or the image contents). The main originality is that the proposed method uses artificially (synthesized) test images for the noise estimation and filter design.

Noise Elimination:

The variation method for image processing proceeds by minimizing a functional, thereafter referred to as energy, which depends on the image and its space derivatives (gradient). The one that is considered is the sum of two terms: one represents the deviation from a data image z, which may be marred by noise, blur or speckles, and the other incorporates the variation in the function (in the mathematical sense). The last term penalizes oscillations and irregularities, but does not remove jump discontinuities altogether. Such discontinuities are considered necessary to preserve the information content (sharpness) of the image.

The energy we minimize is:

$$\tfrac{1}{2}\lvert Au - z \rvert^{2} + \alpha \int \lvert \nabla u \rvert$$

where z is the data image and the second integral term represents the variation in the candidate image u. When there is blurring, A represents the action of blurring operator, and z is the blurred and noisy image. Here we will assume no blur and take A to

be the identity. This is the "purely denoising" case. If u is smooth, the second term clearly measures its average oscillation over the domain; but this integral term may remain bounded even if u has jump discontinuities (as one expects of an image). For example, if u is the characteristic function of the unit disc, then the second integral term will be exactly two π, which is the amount of "falloff" of u integrated over the boundary of the unit disc.

Iterative methods for carrying out the minimization have already been implemented [4, 5]. The convergence rate to the denoised image seems to depend on its smoothness. Figure 1 illustrates the use of these methods in the one-dimensional case.

Note that for a critical value of the coefficient a, the minimizer of the energy develops the sharp edge at the same location as the original image.

3. Genetic Algorithms (GAs):

The GA is actually not a single algorithm, but rather, a class of methods, which purport to minimize a function, defined on a domain by [5, 7, 8]:

discretizing the domain so that a vector of fixed length N (chromosome) codes each value of the state variable.

randomly selecting an initial population of chromosomes by sampling from the domain using a uniform distribution.

letting this initial population evolve over a given number of generations. The best fit member of the last generation is taken as the solution. The evolution rules, which vary, may be as follows:

a) Selection: based on their fitness, randomly select couples for mating.

b) Each couple will produce one (or, alternatively, two) off springs by randomly generating a "crossover" rule.

c) The off springs now forms the new population, which is of the same size as the previous population.

d) An intermediate (and desirable) step between 3b and 3c is to mutate randomly some of the offsprings.

See Figure 2 for best understanding of a GA.

Current Progress

As a beginning, we are confining the image to be one-dimensional, so that N, the size of the state variable, is the size of the discretization of the interval. Since the energy is a convex functional, a preliminary step consist in testing the genetic algorithm on the simplest convex problem, namely, minimize x^2 over an interval.

Our control parameters include all the probabilities governing the random events, and how to generate the crossover rule. While we do not expect the genetic algorithm to outperform the direct (iterative) method, it will be of interest to see how well it approximates the solution, and how the execution time grows with the size of the problem [6].

Another advantage of the genetic algorithm compared with direct methods is that it is much easier to code. We are using MATLAB in our current project: see Figure 3.

For the GA some tests have been made with the scalar problem for coding and improving purposes. After this step, we started implementing the one-dimensional case using the first term of the functional to be minimized. The first term of the functional is discretized as:

$$\Sigma \, \Delta x \, (\, u - z \,)^2$$

where Δx is the width of each pixel. We tested the dependency of the final solution u with respect to the following parameters.

The number of binary bits coding the intensity of each pixel, the number of generations and the size of the population. Figures 4-6 show those dependencies. As Figure 4 shows, increasing the number of bits improves the approximation. Figure 5 shows that increasing the number of generations in the range 5 to 20 also improves the approximation.

For the case of the size of population selected Figure 6 we found that it is not necessary to increase this number indefinitely to find a good solution, in our test a population of 40 was enough to find good approximations considering the effect of the other parameters.

Future Work

Preliminary experiments with larger number of pixels seem to indicate a degradation of performance. Our next task is to check whether this is true and if so, why; It is imperative to address this question before going on to the two-dimensional case. After that the complete functional, including the penalty term must be implemented. Other improvements must be to add more statistics to the code in order to track the performance of the algorithm. In addition, we have to do some refinement of the fitness selection procedure.

References

1. Kent A. and Williams J. G., Encyclopedia of computer science and technology, Marcel Dekker Inc, 1988.
2. Koza J. R., Genetic programing on the programming of computers by means of natural selection, Massachusetts Institute of Technology, book, 1993.
3. Vertan C., Buzuloiu V, Reduced computation genetic algorithm for noise removal, Bucharest "Politehnica",PP 45-52 University, Romania , 2000.
4. Vogel, C.R and Oman M. , Iterative methods for total variation denoising, SIAM J Sci Comp 17:227-28, 1996.
5. Bogees, G. and Mukheeth A., The Application of Genetic Algorithims to the scheduling of Engineering Units, Dec31, 1997.
6. Gershenfeld N., The Nature of Mathematical Modeling, Cambridge University Press, 1999.
7. Goldberg D., Genetic Algorithms in Search, Optimization and Maching learning, Addison Wesley, 1989.
8. Negron – Marrero P.V, A genetic algorithm for computing singular minimizers of the calculus of variations, SIAM J Sci Comp 34:177-20 ,2000.
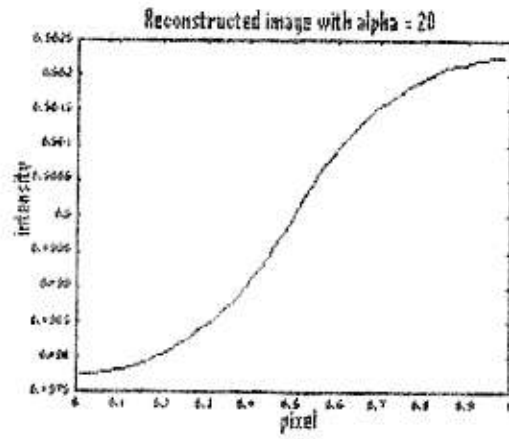
Fig. 1: Varying penalty term reconstruction of a single
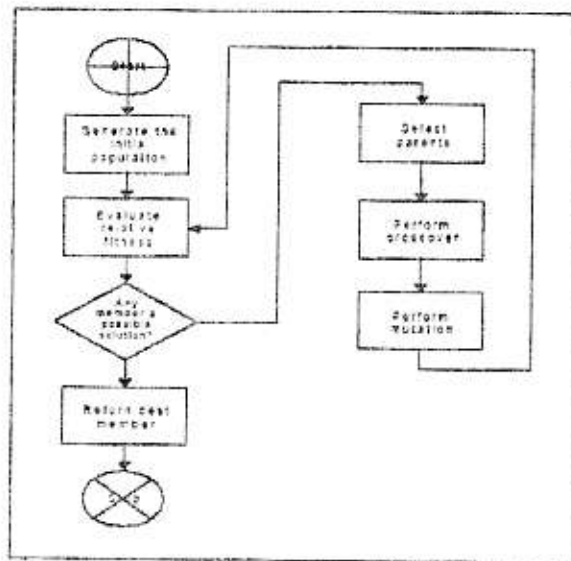step image in the absence of noise by an iterative method.

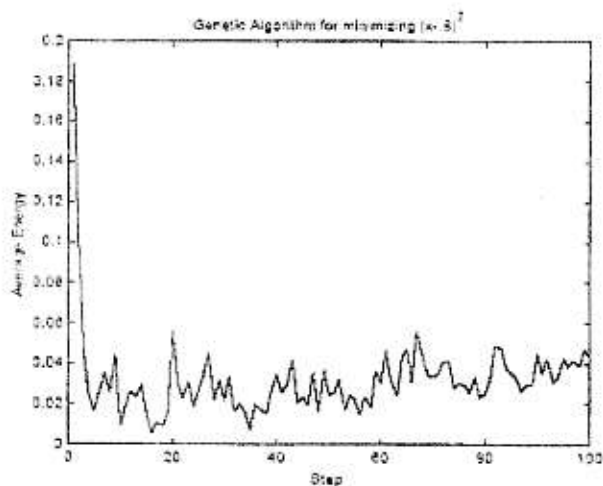Fig. 2: General GA representation in flow chart form.

**Fig. 3: Energy vs. Generations Preliminary results
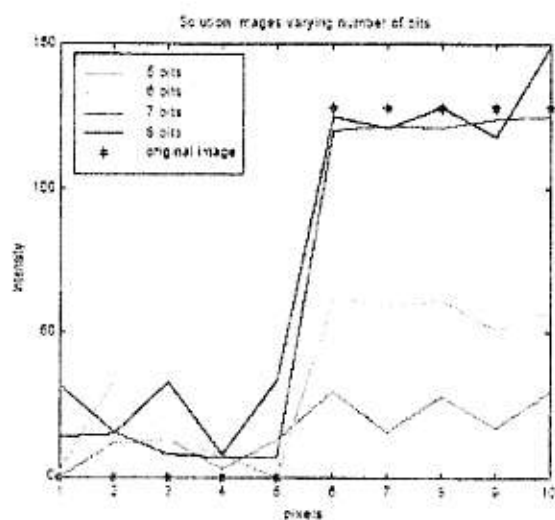for genetic algorithm**

**Fig. 4: Varying the number of bits, the solution approximates
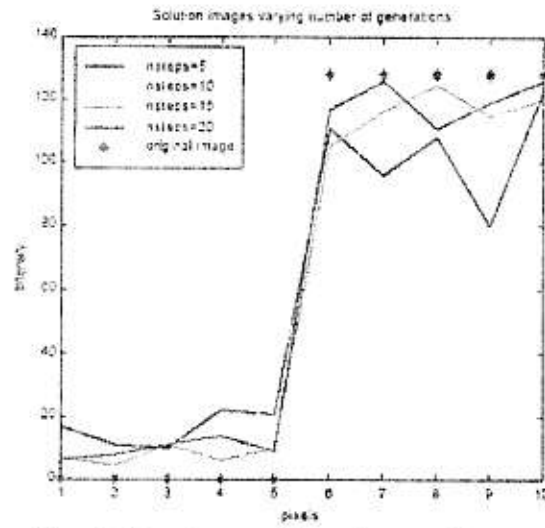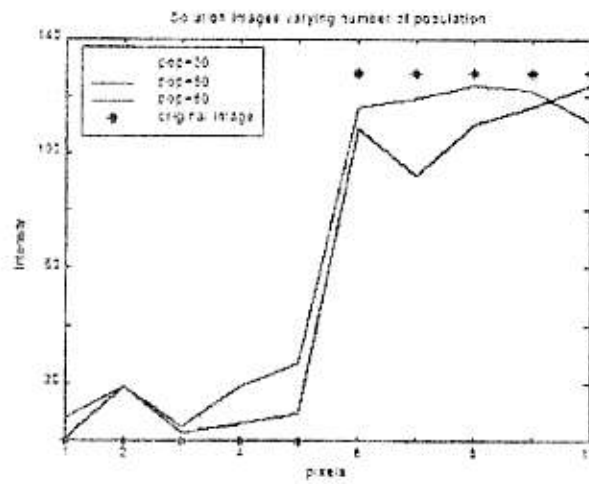the given image z.**

Fig. 5: Varying number of generations.

Fig. 6: Varying the amount of population per iteration.