Image Segmentation based on Genetic Algorithm

Dr. Jane J. Stephan
E-mail: janejaleel@yahoo.com
Abeer Diaa Alnakshabandi
E-mail: abeeralnakshabandi@yahoo.com

Iraqi Commission for Computers and Informatics
Informatics Institute for postgraduate studies
Baghdad-Iraq

Abstract

One of the most difficult tasks in image processing is the determination of a suitable set of features which can be used to segment images. In this research, the standard deviation that represents feature of image has been used in image segmentation as indicator to determine the isolation of one object from another or an object from a background. This feature has been used by a Genetic Algorithm (GA) to become a fitness function that will help in searching process for the optimal solution. The value of standard deviation is high in the case of a difference between various diverse regions of the image and small in one region. Using this feature in maximizes the difference among different regions and minimizes the interclass variance, a GA is used to evolve a sub-image convolution kernel to produce kernel with a best features that can be used in the segmentation of image. The space-filling curve approach has been used to convert the kernel from a one dimensional (1-D) form into a two dimensional (2-D) form. The evolution process of a genetic algorithms are done on a kernel in array form, while the convolution process between a kernel and image is used a kernel in a matrix form.

Keywords: genetic algorithms, convolution kernels, image analysis

1.Introduction

Data reduction encompasses the process of image segmentation, filtering and feature extraction [1].

Feature extraction is the process of taking an image and producing another image which has areas with a defining characteristic highlighted with respect to the background. Features such as edges, gradients, other spatial features can be detected by the application of convolution kernels and represented by the convolution kernel itself. Many of the more common feature extraction methods rely on this method of representation [2].

While many features can be extracted using a convolution kernel, finding an appropriate kernel is not an easy task. Specific kernels exist for edges or lines in a single direction, while other exists for zero-crossings and textures. To find one for a specific feature requires identifying the aspects of the feature that distinguish it from all the others and characterizing it in terms that can be represented in a convolution kernel. This generally requires intense study of a group of already classified or segmented images before an acceptable solution can be discovered [3, 4].

2. Image segmentation

Segmentation subdivides an image into its constituent parts or objects. The level to which this subdivision is carried depends on the problem being solved. That is, Segmentation should stop when the objects of interest in an application have been isolated. It is one of the most difficult tasks in image processing and acts as the first step in image analysis. This step in the process determines the eventual success or failure of the analysis. In fact, effective segmentation rarely fails to lead to a successful solution. For this reason, considerable care should be taken to improve the probability of rugged segmentation[5].

3. Genetic Algorithm

Genetic Algorithms are adaptive methods which may be used to solve search and optimization problems. They are based on the genetic processes of biological organisms. This algorithm maintains a population of structures, that evolve according to rules of selection and other operators, that are referred to as "search operators", (or genetic operators), such as recombination and mutation which operates by combining and mutating the representations of solutions in one population to form a next population of possible solutions for evaluation. Each individual in the population receives a measure of its fitness in the environment. Reproduction focuses attention on high fitness individuals, thus exploiting the available fitness information. Recombination and mutation perturb those individuals, providing general heuristics for exploration. An effective GA representation and meaningful fitness evaluation are the keys of the success in GA applications [6, 7].

4. Image segmentation using Genetic Algorithm

This research presents a method of applying GAs to the problem of finding an appropriate feature which can be used to segment images. The GAs includes the methods of measuring performance and a suitable representation of convolution kernels for the application of the GA approach [8, 9].

4.1 Representation of kernels

A fundamental problem of GAs is the representation of the solution, or genome that will be optimized. The representation which is chosen will affect all the aspects of the breeding process, since the breeding operations of necessity operate on this representation with no knowledge of the problem. Many genomes are represented in the form of a vector due to the simplicity of defining storage and working with the elements. The crossover and mutation operation conceptually work on a stream of data in array format. This means that no special algorithms for performing these two basic GA processes need to be developed.

4.1.1 Kernel values mapping

A space-filling curve is one approach that has been used to overcome the discontinuities problem which appears at the end of each row of the matrix in using the computers array representation of a matrix & sporadic insertion or deletion of elements when changing kernel size. Each array element in this approach is related to its neighbors within its local group in two dimensions. While convolution kernels can be either even or odd sized, odd is often more desirable from an implementation sense due to its symmetry and its clearly denoted center element.

An approximation to a space-filling curve was chosen which minimizes the effects of these difficulties while still allows a GA to operate on the kernel. This concatenation of convolution kernel elements will be called a spiral kernel and is shown in figure (1).

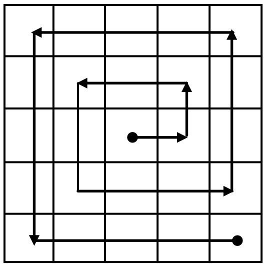


Figure (1) Spiral kernel for mapping kernel values to GA genome

The spiral kernel starts in the center of the matrix and can take one step in any direction; this case will then turn perpendicular to its previous direction and travel one more element. It will continue to turn in the same direction after each n steps, n will increment for every other turn. This continues until the matrix has been filled.

4.1.2 Kernel size

Change of size operations translates to resetting the array length, and center operations on the matrix can be bounded by two points on the array. There are no hard discontinuities, and each element is moved a relatively small distance with respect to most of its neighbors. The limitation which remains is that the row and column size must be the same. This is quite acceptable since zero padding a convolution matrix does not affect its results if the center does not change , and row/column size will always be the same in the populations given to the algorithm.

4.1.3 Weighted kernels

All kernels in each generation must have been weighted in order to discriminate them from each others, and to make sure that the kernel does not repeat. The equation of the weighed kernel is

$$\sum_{s=0}^{(n\times n)-1} A(s) = 0$$

$$W_K = \sum_{s=0}^{(n\times n)-1} 3^s \left(A(s)+1\right)$$

where $n \times n$: is kernel size

$A(s)$: is a kernel value

(one of three cases -1, 0, and 1)

W_K : is kernel weight

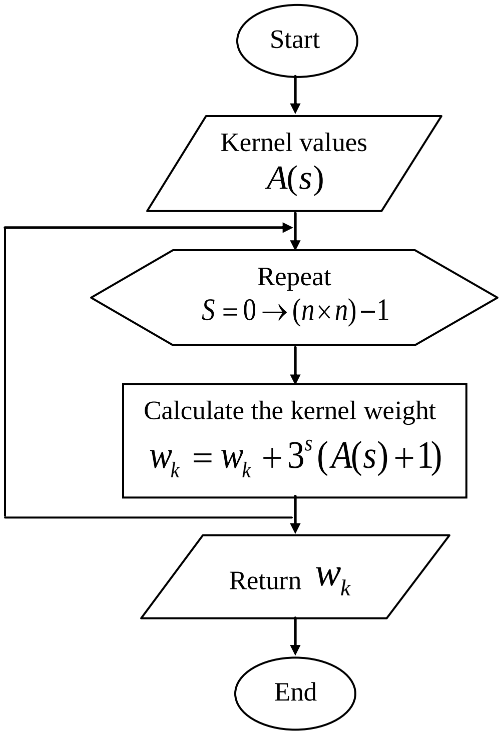

Start
Kernel values
$A(s)$
Repeat
$S = 0 \rightarrow (n\times n)-1$
Calculate the kernel weight
$w_k = w_k + 3^s (A(s)+1)$
Return w_k
End

Figure (2) Flow Chart of subroutine calculate a kernel weight

4.2 Measure of performance [10] (GA Fitness Function)

One approach of evolving a segmentation convolution kernel is to maximize the difference among the numerical values which results from the application of the kernel to various diverse regions of the image. This can be done while at the same time minimizing the interclass variance. This approach will result in the maximum probability of correct classification and minimum probability of incorrect classification for linearly separable regions. The standard deviation is a fitness function obtained by convolving the complex region of image with a kernel value, and then computing the following equation that illustrated in figure (3):

$$M = \frac{1}{R \times C} \sum_{i=1}^{R} \sum_{j=1}^{C} \hat{I}(i, j)$$

$$\sigma = \sqrt{\frac{1}{R \times C} \sum_{i=1}^{R} \sum_{j=1}^{C} (\hat{I}(i, j) - M)^2}$$

where $\hat{I}(i, j)$: is convolution pixel (result of convolution kernel with image region)

M : is mean of convolution pixel

$R \times C$: is region size (R is Row and C is column)

σ : is standard deviation

To evaluate the fitness function, the image is manually segmented. This is done by replacing all pixels of a specific feature, or region, with a distinct integer value. All Pixels not covered during this process are set to zero to characterize them as background noise. The resulting map clearly defines which region each pixel is in, so that the statistics can be found by region rather then by image. The mean and variance is then calculated for each region.

The overall effect is that as the region converges to a single value from the transform, the variance will approach zero. In early trial runs the variance was the only measure of fitness. The GA quickly evolved an all zero matrix which always has a small variance; no matter which image is being convolved.

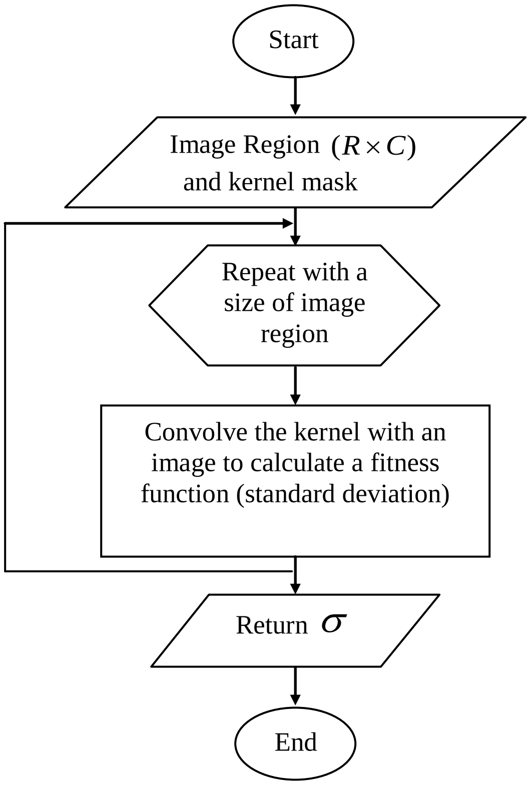

Start
Image Region $(R \times C)$ and kernel mask
Repeat with a size of image region
Convolve the kernel with an image to calculate a fitness function (standard deviation)
Return σ
End

Figure (3) Flow Chart of subroutine calculate a fitness function of kernel

4.3 The Evolutionary process [11]

Each iteration of the GA ends with the creation of a new set of genome /kernels based on the fitness of members of the previous generation. Each kernel is converted from its matrix form into the array form of the genome. Three methods are then applied to create a new set of genomes, i.e, the next population.

An elitist approach is used in which the kernel with the maximum fitness is copied into the new population without alteration. The remaining slots in the population are then filled by applying the crossover operator to the other population members. Each slot filled by a crossover generated kernel is then passed through a mutation operator occasionally.

Elitist is a mechanism which ensures that the chromosomes of the most highly fit member(s) of the population are passed on to the next generation without being altered by genetic operators. Elitist can very rapidly increase performance of GA and brings about a more rapid convergence of the population, because it prevents losing the best found solution. If they do lose the best genome, then the GA could take many generations to rediscover this genome. It is important to note that the GA does not operate by converting a random string from the initial population into a globally optimal string via a single mutation. Only advance after enough generations had passed to have mutation provide a better solution to converge towards. When the convergence never saturates the population, the speed at which it happens allows other advances to occur before the population becomes stagnant.

4.3.1 The Crossover operator

Crossover is used to swap parts among genomes to produce new kernels. By selecting a point between the first and last genome elements, a crossover point is established. Two genomes are passed to the crossover operator and two children are returned. Each of the children receives all the genome preceding the crossover point from its respective parent, and everything past that point on the genome from the other parent. Parents are chosen by generating two random Gaussian distributed numbers, with a zero mean and a constant variance, and using the absolute value as an index into the list of genomes ordered by fitness. If crossover occurs often enough, then an offspring will be produced that contains the good parts of both parents and the other offspring will die off due to their receiving only the bad genetic material. Crossover also has the effect of randomizing genomes so that other areas of the solution space are searched by default for an optimal genome while retaining at least some common thread to part of the population.

In this research, a crossover operator (i.e. single point crossover) is done 25 times between each pair. The crossover point that will be passed in a crossover process is chosen 25 times in each kernel. The value of cross site is from 1 to 25.

In the case of using a population size 50, the crossover rate that will be chosen is 0.6. Thus the crossover operator is done between fifteenth pair (i.e. 50 * 0.6 = 30). The kernel1 to kernel20 have been copied without alteration in each generation from the previous generation to the next generation.

While In the case of using a population size 20, the crossover rate that will be chosen is 0.7. Thus the crossover operator is done between seven pair (i.e. 20 * 0.7 = 14). The kernel1 to kernel6 have been copied without alteration in each generation.

Figure(4) shows two individuals in a population of 8-bit strings undergoing single-point crossover; the point of exchange is set between the fifth and sixth positions in the genome, producing two new individuals that are a hybrid of their progenitors.

Parent1	0	0	1	0	1	1	0	1
Parent2	1	0	1	1	0	0	1	1
Offspring1	0	0	1	0	1	0	1	1
Offspring2	1	0	1	1	0	1	0	1

Figure (4): Single-point crossover

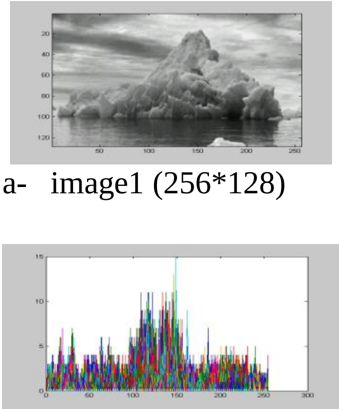
4.3.2 The Mutation operator

Each element in the genome is considered for mutation individually via a uniform random number generator. If the probability for mutation is greater than the generated generated number, then a mutation occurs. If not, the next element in the genome is considered. When an element is mutated, a Gaussian random number is generated with a mean equal to the original element, and variance specified in the program. The generated number is rounded to An integer and placed in the genome. This has the effect of attempting to hill-climb to a local solution when small deviations are generated. Random alteration on one element at a time will equivalently try random directions to see which one produces a better fitness. Larger deviations in the element due to mutation may also alter the location of the genome in the solution space where other optimal solutions might be found, much like crossover.

Uniform and large variance random numbers were tried with mutations resulting in damaged genome (low fitness on the next iteration). This appears to be due to the fact that the majority of successful genomes use the value 1,0, and -1 exclusively and when a number far away from these three values appears it causes an unbalancing of the kernel. Multiple mutations might find a solution outside the locality of the origin, but this is unlikely to occur in theGA. Higher probabilities of mutation lead to a large group of unsuitable genomes and a degenerate behavior for the population as a whole.

5. Results

After 22 generations of a population size20 in image1, the optimal kernel with the best fitness function of a minimum standard deviation ($\sigma 1$ =45.067) which is shown in figure (6) is obtained. (The population is said to be converged when all of the genes have converged and the average fitness will approach that of the best individual) .This kernel has been used to segment the original image1 (see figure (5)) into three perfect separate regions as in figure (7) by the convolution process between these images and the best kernel and then extract the features for each region (the mean and variance are then calculated for each region). Figure (8) illustrate the result of a histogram of the segmented image1.



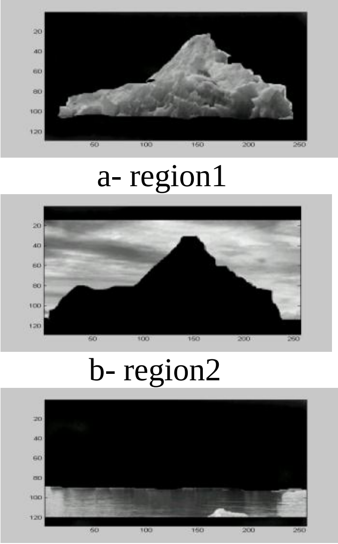
a- image1 (256*128)

b- Histogram

Figure (5) image1 and the corresponding histogram

−1	−1	0	1	1
−1	−1	0	1	1
0	−1	1	1	0
1	1	−1	−1	−1
1	1	0	−1	−1

Figure (6) the optimal kernel in 2-D



a- region1

b- region2

c- region3

Figure (7) separated regions of image1

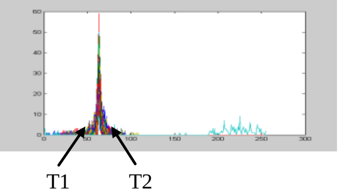


Figure (8) the histogram of a final result of the segmented image1

From the histogram of a convolution result in figure(8), the value of threshold is derived by using the triangle algorithm. This algorithm has been used to derive a threshold value from a segmented histogram. A line is constructed between the maximum of the histogram at the highest brightness and the lowest value in the image. The distance between the line and the histogram is computed for all values from the maximum and minimum of the histogram. The brightness value b0 where the distance between h[b0] and the line is maximal is the threshold value.

The segmented histogram in this thesis has three peak (two weak peak and one high peak) and two valley. Two theshold values have been derived from this histogram at a two valley, then these threshold values are used to separate the image into three regions as in the following equation:-

If result <= T1 then region1
If T1 < result <= T2 then region2
If result > T2 then region3

After the assurance of a perfect seperation, the mean and variance are calculated for each region in image1.

6-Conclusions

1. The first, and most important, consideration in creating a GA is defining a representation for the problem. The language used to specify candidate solutions must be robust; i.e., it must be able to tolerate random changes such that fatal errors or nonsense do not consistently result.

2. The problem of how to write the fitness function must be carefully considered so that higher fitness is attainable and actually does equate to a better solution for the given problem. If the fitness function is chosen poorly or defined imprecisely, the GA may be unable to find a solution to the problem, or may end up solving the wrong problem.

3. In addition to making a good choice of fitness function, the other parameters of a GA - the size of the population, the rate of mutation and crossover must be also chosen with care. The performance of GA's appear to be a nonlinear function of the these control parameters.

4. The filters that are used in this research are implemented with convolution masks. Because a convolution mask operation provides a result that is a weighted sum of the values of a pixel and its neighbors, they are called linear filters.

5. In Edge Detection operator that will be used in image segmentation, the summations of the convolution mask coefficients are equal zero. This results in regions of constant brightness being reduced to zero, and regions of abrupt changes in gray level returning large values, which indicates prominent edges.

6. Starting from a number of random convolution kernels, it has been shown that a kernel that meets a specific feature extraction problem,
which is equivalent to a linear transform, can be evolved before hand.

7. Due to the size limitations of the kernels evolved here, three to eleven row/column size with most being five has been used. This size is not very small and not very large. Because in the field of image processing small filters result in too many noise points and large filters tend to dislocate the edges. In the field of GAs, the smallest kernel will be developed regardless of the performance and large kernels should give good results, but require more computational time.

8. The changing process in a population size from 20 kernels to 50 kernels is helped in improving the values of fitness but this changing is required more computational time.

9. While convolution kernels can be either even or odd sized, odd is often more desirable from an implementation sense due to its symmetry and its clearly denoted center element.

7. References

[1] Gonzalez, R. C., and Woods, R.E., "*Digital Image Processing*", Reading, MA : Addison - Wesley Publishing Company, 1992.

[2] Umbaugh, S. E., "*Computer Vision and Image Processing*", Prentice – Hall PTR, 1998

[3] Niblack, W., "*An Introduction to Digital Image Processing*", Prentice - Hall International, (UK) LTD, 1986.

[4] Marr, D. , Hildreth, E. , "*Theory of Edge detection*", Proc. R. Soc. Lond, B. 207, 187-217, 1980.

[5] Schalkoff, R. J. , "*Digital Image Processing and Computer vision*", New York: Wiley, 1989.

[6] Holland, J. H., "*Genetic Algorithms in search, optimization, and machine learning*", university of Michigan Press, Ann Arbor, 1975.

[7] Buckles, B. P., and Perty, F. E, "*Genetic Algorithms*" , IEEE Computer Society Press Technology Seria, 1992.

[8] Whitely, D., "*A Genetic Algorithm Tutorial*", Technical Report CS- 93-103, Colorado State University, 1993.

[9] Arslan, T., and Horrocks, D.H., "*A GA for the Design of finite word length arbitrary response cascaded IIR digital filters"*, Proceedings of IEE / IEEE Int. Conf. on GAs in Engineering systems, Sheffield, UK, 1995.

[10] Koza, J.R.,"*Genetic Programming*", 2nd Edition, MIT Press, Cambridge, MA, 1993.

[11] Quagliarella, D.,Periaux, J., Poloni, C., and Winter,G., "*GAs and Evolution Strategy in Engineering And Computer Science*", John Wiley & Sons Ltd, England, 1998.

تقسيم الصورة باستخدام خوارزمية الجينات

د.جين جليل اسطيفان
عبير ضياء النقشبندي

الهيئة العراقية للحاسبات والمعلوماتية
معهد المعلوماتية للدراسات العليا

ملخص البحث

أغلب المهمات الصعبة في معالجة الصور هي معرفة الخصائص المناسبة التي تستخدم في تقسيم الصور. في هذا البحث، مقياس الأنحراف الذي يمثل احد صفات الصورة سوف يتم استخدامه في عملية تقسيمها وذلك باستخدامه كدليل لمعرفة عزل منطقة عن منطقة اخرى أو عزل منطقة عن الأرضية. هذه الصفة استخدمت من قبل خوارزمية الجينات كدالة مناسبة لها في عملية البحث لأيجاد الحل الأمثل. قيمة مقياس الأنحراف تكون عالية في حالة وجود اختلاف بين المناطق المختلفة وتكون قليلة في المنطقة الواحدة. باستخدام هذه الصفة في زيادة الأختلاف بين المناطق المختلفة وتقليل الأختلاف في المنطقة الواحدة، استخدمت خوارزمية الجينات في تطوير الـkernelالذي يلتف بالصورة لأنتاج kernelبأحسن الخصائص التي تمكنه في عملية تقسيم الصورة.

استخدمت الـ Space-filling curve approach في عملية تحويل الـkernel البعد الأول الى ثنائي الأبعاد. ففي عملية التطور في خوارزمية الجينات استخدم kernel بشكله احادي الأبعاد. بينما في عملية الألتفاف تم استخدام الـKernelبعد تحويله الى مصفوفة ذو بعدين.