Journal of Geography and Cartography (2019) Volume 2
doi:10.24294/jgc.v2i1.830

Genetic Optimization Mapping Applied to Medical Image Segmentation

Pierre-Richard, Jean Cornely

University of Massachusetts, Lowell

ABSTRACT

A number of important optimization problems have been classified as mapping applied towards segmentation of important features. The segmentation of important features can be formulated as configurational mapping problems by representing mapping configurations as solutions to problems of interest. One example of such configuration mapping is found in image segmentation where an image can be represented as unique subsets of a complete image and then evolved through mapping to become a segment of specific interest within an image. An effective segmentation mapping algorithm must determine the specific image subsets of an image field that best exhibit an a priori set of quantitative and qualitative characteristics. In this paper, a Genetic Optimization Mapping Algorithm is used to produce a population of sub-images, characteristic of specific image subsets of interest that were tested via a quantitative objective function, ranked using a linear fitness scheme, and modified using a genetic Crossover operator. The mapping algorithm is found to converge, within fifty to one hundred generations of maps, to a good fit to the targeted mapping configuration in a very robust and efficient manner.

Keywords: Genetic Mapping; Image processing; Medical image segmentation; Texture segmentation

1. Introduction

A number of important problems in medical image segmentation can be classified as segmentation problems. Some medical image segmentation problems of interest are: "tissue characterization via textural quantification using probability density function models", and "detection of breast cancer in mammograms". Image segmentation has been successfully performed in the past with a variety of approaches Blostein[2], and Clark[4]. The effectiveness of each approach depends on the type of segmentation problems to be solved. For example, if the objects we are attempting to detect have unique intensity ranges, a threshold approach may be used. The threshold approach is based on a discrimination of the pixel intensity values of the image based on whether a certain gray level intensity value is above or below a predetermined threshold, Coggins[5]. This approach will work only in cases where the gray level is sufficient to depict the desired characteristics within the image. If the information we want lies solely at the edges of the image, one approach that has been used is edge detection, Cross[6]. In this method, the concern is to locate the boundary of an object by locating its edges. This approach will only be valid in cases where the edges of the segments of interest are distinctly defined. Medical images in general, cannot be classified by their intensity values or a sharp distinction of edges, Beck[1]. A practical image segmentation problem of interest such as tumor detection, does not exhibit any of the above characteristics and as such, the approaches mentioned earlier will not be successful for this class of segmentation problems. Texture segmentation can be defined as a discrimination of the pixel values in an image based on their textural content. The ability to segment a textured image into separate regions continues to be a challenging problem in computer vision. A variety of the image segmentation methods that have been proposed are based on filter models, where the filters are derived from Gabor elementary functions. One such method transforms the texture differences in an image into detectable filter output discontinuities at the texture boundaries, Turner[15]. In most cases, the optimal solutions proposed by these techniques are obtained via gradient methods. However, finding optimal solutions using

doi: 10.24294/jgc.v2i1.830

gradient methods is very computationally intensive. This statement is true for Bayesian optimization techniques such as: Maximum a Posteriori (MAP), Maximum Likelihood (ML) and Expected Maximization Maximum Likelihood (EMML) and even when iterative algorithms are used, Byrne[3]. Another class of optimization algorithms finds an optimal solution by using random search patterns based on evolutional maps driven by analogies to the behavior of physical systems. For example, a class of optimization algorithms called Simulated Annealing Optimization Algorithms, use the idea that when a metal is cooled slowly from its liquid to solid state, its final lattice structure tends to become more organized and symmetrical. Moreover, the energy of the lattice energy transition states are lower than when the lattice is cooled quickly. This empirical observation is used as the basis for the search for optimal solutions that are associated with low energy "cooling schedules" through a Boltzman energy distribution function, Lawrence[13]. However, despite the fact that simulated annealing algorithms do not use gradient methods, they are serial algorithms (i.e., one step of the optimization requires the previous step), and as such, they may not be suited for applications to real time optimization problems. In this paper, we view the segmentation problem as a configurational mapping optimization problem, where the objective is to find a mapping configuration that best exhibits an a priori set of statistical and textural characteristics, using a Genetic Optimization technique. Genetic Optimization Mapping Algorithms mimic some of the optimization properties observed in natural evolution. Holland[11] believed that, appropriately incorporated in a computer algorithm, these mapping techniques might yield an innovative method for solving difficult optimization problems in a way similar to natural systems, through evolution. His proposed algorithms manipulated strings of binary digits that he called chromosomes. Like nature, his proposed algorithms solved the problem of finding good chromosomes by manipulating the material in the chromosomes "blindly". Like nature, they knew nothing about the type of problems they were solving. The only information they were given was an evaluation of the fitness of each individual chromosome they produced. The only use of that evaluation was to bias the selection of chromosomes so that those with the best evaluations tended to reproduce more often than those with lower evaluations. These mapping algorithms, using some simple encoding and reproduction mechanisms, displayed complicated behaviors, and turned out to solve some extremely difficult optimization problems, Dejong[8-9]. Like nature, they did so without knowledge of the decoded world they were encoding for, an interesting and puzzling phenomenon. They were simple manipulations of relatively simple chromosomes. Yet, when we use the descendants of these mapping algorithms, we find that they can evolve better designs, find better schedules, develop better mappings and produce solutions to a variety of important optimization problems, Vignaux[16] and Leether[14], that we cannot solve as well, or at all using the other well known optimization techniques discussed earlier. In this paper, the Genetic Optimization Mapping Algorithms developed in this paper uses the basic framework in[7],[8], and[11] with some added features and applies the technique to Medical Image Segmentation.

2.1 Genetic optimization algorithm theory

Adaptation is defined as progressive modification of a set objects I, which is done by repeated actions of a chosen set of modifiers or operators O. The goal of adaptation is to ensure that the modifications produced by the operators create entities that belong to a particular subset of a defined search space S. A Genetic Optimization Algorithm A is an adaptation technique, which uses a set of factors controlling the way that the operators O are utilized to modify the candidate objects at each stage of the adaptation process. Texture image segmentation on the other hand, can be defined as the detection of a group of contiguous pixels in an image that have similar statistical, structural and textural characteristics. If we assume that the segmentations of interest are gray scale images of size $(M \times N)$, an exhaustive search, for example, of a (64×64) image field requires testing $2^{64\times64}$ distinct possible candidate solutions. These candidate solutions are binary image maps that, when masked over an input image, can be tested for their fitness relative to the segmentation problem of interest. An exhaustive search for the best image map solution is clearly inefficient and becomes even more difficult as the size of the image increases. The Genetic Optimization Mapping technique allows to efficiently search the space for the best image map subsets of the image field, that best exhibits an a priori set of quantitative and qualitative characteristics. The proposed Genetic Optimization Mapping Algorithm achieves this result by producing population of sub-image maps that are tested via a quantitative objective function, ranked using a linear fitness

and decrement scheme and modified using a random Crossover operator. The algorithm is found to converge within fifty to one hundred generations to a good fit to the targeted mapping segmentation.

2.2 The proposed genetic optimization algorithm

The proposed Genetic Optimization mapping Algorithm A is a configurational optimization algorithm that uses genetic operators to modify a population $I(t)$ of candidate segmentation masks at time t into $I(t+1)$. A candidate segmentation mask from population $I(t)$ is tested against a set of characteristics known a priori to be representative of the segmentation of interest. The population $I(t)$ is made of L candidate segmentation masks at iteration t

$$I(t) = \{I_1, I_2, I_3, ..., I_L\} \quad (2.2.1)$$

Each candidate segmentation mask is made of a set of k chromosomes

$$I_j = \{c_{j1}, c_{j2}, c_{j3}, ..., c_{jk}\} \quad (2.2.2)$$

Each chromosome is made of a set of m genes

$$c_{ji} = \{g_1, g_2, g_3, ..., g_m\} \quad (2.2.3)$$

An encoding scheme is used, that maps a string of bits to a binary image mask. Each bit string is mapped to a binary image or candidate segmentation mask by representing the bit strings as the union of a random number of smaller bit strings, each of which can in turn be mapped to binary parallelograms. Each binary parallelogram is determined from six components: a center (x_c, y_c), two radii (r_1, r_3), and two angles (θ_1, θ_3). Also, each binary parallelogram is referred to as a chromosome and the components that formed it are referred to as its genes. In this mapping therefore, a chromosome is made out of six genes and a candidate image mask is made out of the union of a random number of chromosomes

$$I_j = \{(g_1, g_2, g_3, g_4, g_5, g_6) \cup ..... \cup (g_1, g_2, g_3, g_4, g_5, g_6)\} \quad (2.2.4)$$

Where:

$$g_m = \{r_1, r_3, \theta_1, \theta_3, x_c, y_c\} \quad (2.2.5)$$

The parameters (x_c, y_c) in (2.2.5) are the Cartesian coordinates of the center of the parallelogram, (r_1, r_3) are the diagonals and (θ_1, θ_3), the angles that the diagonals make with the x-axis as shown in **Figure 2.2.1**. A particular case where the binary image mask is made of just two chromosomes is also shown in **Figure 2.2.1**. The adaptation process begins with the creation of an initial population of binary candidate image segmentation as in (2.2.1). In any given population, candidate segmentation maps will have a random number of chromosomes, and each chromosome is constructed as described by (2.2.4) and (2.2.5). The candidate segment is accepted as a valid candidate only if its area is such that it obeys the following relationship:

$$S_t\_I_i \le S_c\_I_i \le 2S_t\_I_i \quad (2.2.6)$$

Where $S_c - I_i$ is the area of the candidate and $S_t - I_i$, the area of the target. The chromosomes, which make a candidate map are therefore chosen such that the total area covered by their union, as shown in Figure 2.2.1, satisfies (2.2.6). Then, the proposed Genetic Optimization Mapping Algorithm A takes the set of candidate segments of a given population $I(t)$ from the search space and produce another set of candidate segments $I(t+1)$ from the search space such as:

$$A(I(t)) = I(t+1) \quad (2.2.7)$$

In addition, the proposed Genetic Optimization Mapping Algorithm A contains a reproduction process, which applies the Crossover operator C to pairs of candidate segments, usually called the parents, to produce another pair of candidate segments, usually called the children. In addition, a random bit Mutation operator M, is applied to the entire population of children candidate segments. Both the Crossover and Mutation operators perform modifications such as to produce children candidate segments of the search space whose characteristics match those of the segmentation of interest as:

$$A(I(t)) = M(C(I(t))) = I(t+1) \quad (2.2.8)$$

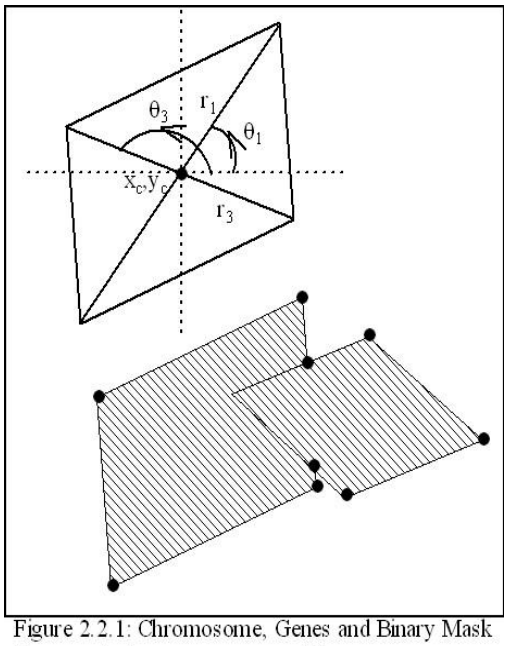

Figure 2.2.1: Chromosome, Genes and Binary Mask
Top: A chromosome c_{ji}, and its genes g_m
Bottom: A binary mask I_{ji} of two chromosomes

The proposed Genetic Optimization Mapping Algorithm A also uses an information function that consists of a set of goodness evaluations and ranking measures. From the information function, the algorithm is able to determine the pairs of parent candidate segmentations with the highest rankings and consequently biases the Crossover operator process such that it will take place more often with those segmentations of high-ranking evaluations and high fitness measurements. The biasing is done in such a way that does not entirely preclude candidate segmentations of poor fitness from taking part in the process. As such, candidate segmentations with poor fitness, but that may potentially contain genetic information, which may help the adaptation process are still allowed to participate in the reproduction process.

2.3 The crossover module

The operation performed by the cross over module for the proposed Genetic Optimization Mapping Algorithm can be summarized as follows:

Two parent candidate segmentations $\{I_1, I_2\}$ are randomly selected from a given population $I(t)$ as:

$$I_1 = \{c_{11}, c_{12}, ...c_{1x}, c_{1x+1}, c_{1x+2}, c_{1x+3} ....., c_{1x+n}\} \quad (2.3.1)$$

$$I_2 = \{c_{21}, c_{22}, ...c_{2x}, c_{2x+1}, c_{2x+2}, c_{2x+3} ....., c_{2x+n}\} \quad (2.3.2)$$

with the highest ranked segmentation maps having a higher probability of being selected. A random Crossover point x is chosen such that it belongs to both chromosomes $\{I_1, I_2\}$. Two new candidate segmentations, the children candidate segmentations are formed from $\{I_1, I_2\}$ by first, exchanging the set of attributes to the right of a chosen Crossover position x of parent candidate I_1 with those to the right of position x of parent candidate I_2 yielding the child candidate I_1^c. Another child candidate I_2^c is created by reversing the process above finally resulting into two children $\{I_1^c, I_2^c\}$ described by the following set of chromosomes

$$I_1^c = \{c_{11}, c_{12}, ...c_{1x}, c_{2x+1}, c_{2x+2}, c_{2x+3} ....., c_{2x+n}\} \quad (2.3.3)$$

$$I_2^c = \{c_{21}, c_{22}, ...c_{2x}, c_{1x+1}, c_{1x+2}, c_{1x+3} ....., c_{1x+n}\} \quad (2.3.4)$$

The Crossover operator when applied to pairs of parent segmentations potentially generates offspring segmentation maps not in the original population. It also potentially creates candidate segmentation maps with features not contained or seen in the original population. The simulations conducted in this paper, using the proposed Genetic Optimization Mapping algorithm shows that, as was predicted in Holland[11], the algorithm

tends to produce candidate segments with increased complexity.

2.4 The mutation module

A mutation feature is introduced in this mapping technique to guard against rapid homogeneity within the gene pool and as a result to prevent premature convergence. Mutation randomly alters the bits of a gene from a 0 value to a 1 or vice versa with a predefined probability P_m. The primary purpose of mutation is to introduce occasional perturbations in the estimated parameters of Equations (2.2.4) and (2.2.5) to ensure that all points in the mapping space can potentially have a chance at participating in the adaptation process. The bit Mutation operator allows reaching points within the mapping space by performing affine transformations such as scaling, shifting, rotation on individual candidate binary image masks. Generally if P_m is large, the convergence is faster but larger errors will result. This is somewhat analogous to the step size parameter in gradient algorithms; larger step sizes usually imply faster convergence with higher errors whereas smaller step sizes imply slower convergence but smaller errors. In this paper, a conventional bit mutation scheme is used. Upon creating a population of candidate segmentation, the candidate bit string counterparts are gathered together into a single bit string. We perform the bit mutations by finding the number of bits to be mutated based on a predefined bit mutation rate or probability. We also randomly determine the position of the bits to be mutated within the population string. For the simulations used in this paper, a bit mutation probability of 0.001 is used that is, one in every thousand bits is mutated in the entire population. Dejong[8] showed that Genetic Optimization Mapping Algorithms of population sizes from 50 to 100, taken through twenty five to fifty generations with bit mutation probabilities of 0.001, converge to near optimum solutions.

2.5 The ranking module

A linear fitness and decrement scheme is used to rank the candidate segmentations of a given generation. The ranking module receives a set of "goodness of fit" measurements from the goodness of fit module. The highest ranking is associated to the candidate segment with the highest goodness of fit measurement and successive candidates are linearly ranked. For example, let's assume that the linear ranking module receives a set of goodness of fit measurements from a population of 8 candidate segments: 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, and 0.2. First, the goodness of fit measurements are sorted in descending order and subsequently linearly ranked from 8 to 1: 8, 7, 6, 5, 4, 3, 2, 1. Then, new goodness of fit measurements are assigned to the candidate segmentations according to the formula:

$$SP \qquad (2.5.1)$$

where, R is the ranking function and SP the size of the population. The following set of goodness of fit measurements is produced: 1.0, 0.875, 0.75, 0.625, 0.5, 0.375, 0.25, and 0.125. The reader may notice that the linear ranking scheme rearranges the goodness of fit measurements such that in general, the candidates with the lowest evaluations would end up with slightly higher evaluations. Davis[7] discusses how this scheme plays an important role in ensuring that segmentations solutions other than the fittest are selected by the Crossover module. By ranking all individuals linearly, the linear fitness scheme ensures that candidate segmentations of low goodness of fits will have a higher probability of being processed by the Crossover operator, but candidates of poor fitness containing important genetic information can still participate in the adaptation process.

2.6 The fitness module

Fitness viewed as a measure of the adaptive influence upon the future, introduces a concept useful through the whole spectrum of adaptation. The sample space of all possible candidate segmentation maps I_i, that can be generated from the desired encoding mechanism, is examined through an objective function $G(I_i)$, which compares the elements of the mapping space to an a priori set of characteristics representative of the segmentation problem of interest. In this paper, the objective function evaluates three characteristics of the candidate segmentation map. The first evaluation concerns the candidate's statistical content or Texture $T(I_i)$. This evaluation is achieved through a Chi-square test. The Chi-square test compares all the moments of the probability

distribution of a candidate segmentation to an a priori determined probability distribution function based on our prior knowledge of the probabilistic features of the segmentation features of interest. The Chi-Square comparison is done by first finding the Pearson's statistics χ^2 as

$$\chi^2 = \sum_{i=1}^{I} \frac{(N_i - n_i)^2}{n_i} \quad (2.6.1)$$

Where N_i is the histogram of the candidate and n_i that of a target. The Pearson's statistics χ^2 is then normalized using a variable estimate V, which is the sum of the squares of p, zero mean, unit variance, Gaussian random variables x_i as:

$$V = \sum_{i}^{p} x_i^2 \quad (2.6.2)$$

The normalized result V is then used as an argument to the Q-function as:

$$Q\left(\frac{\chi^2}{V}\right) = \int_{\frac{\chi^2}{V}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \quad (2.6.3)$$

A large value of χ^2 in (2.6.2) produces a small probability in (2.6.4) indicating that the candidate corresponding to the histogram N_i is very unlikely. Then, a shape component of a candidate segment $Sh(I_i)$ contrasts the degree in circular irregularity between the candidate map and a predefined morphological template characteristic of our knowledge of some of the geometrical characteristics of the segmentation of interest. The function $Sh(I_i)$ performs the shape evaluation using an eight-radial template measurement approximation of the candidate segment solution as shown in **Figure 2.6.1**. The function $Sh(I_i)$ measures the approximated difference between the template of the candidate segments I_i, and an assumed template T_I_i, characteristic of the segmentation problem of interest, by evaluating the weighted square distances between these templates along the eight radial measurements as

$$Sh(I_i) = \frac{\sum_{i=1}^{8} (dI_i - dT\_I_i)^2}{\sum_{i=1}^{8} (dW\_I_i)^2} \quad (2.6.4)$$

Where, W_I_i is the worse candidate template with respect to the assumed template T_I_i. The final evaluation relates to the size $Sz(I_i)$ of the candidate segment, and a double exponential cost function is used that penalizes candidate segmentation maps of sizes much greater than that of a predefined target segmentation map from being accepted such as:

$$S_t\_I_i \le Sz(I_i) < 2S_t\_I_i \quad (2.6.5)$$

Where, $S_t\_I_i$ is the predefined area of the segmentation map of interest. These three evaluations are then weighted and combined to form the goodness of fit measurement as:

$$G(I_i) = \alpha_1 T(I_i) + \alpha_2 Sh(I_i) + \alpha_3 Sz(I_i) \quad (2.6.6)$$

The overall goodness of any candidate segment is consequently based on our basic prior knowledge of the textural, and geometrical characteristics of the segmentation candidates of interest. The goodness of fit measure limits the subsets of sub-images of interest in their size or the number of pixels they contain, their shape or the way in which the pixels are arranged, and their texture or the statistical features we expect the set of pixels, which belong to the subsets of interest to have. The $\alpha_i's$ are chosen in a manner which reflects the amount and quality of prior information about the texture, morphological and geometrical characteristics of the segmentation problems of interest. When we have average information in these areas, an equal weighting is used. A block diagram implementation of the Genetic Optimization Mapping Algorithm, proposed in this paper, is given in the

following section. The individuals as given by are

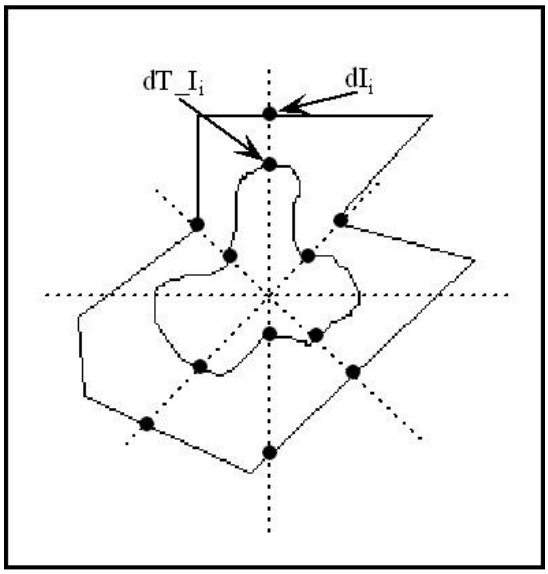

Figure 2.6.1: Shape component of Goodness of fit measure. dT_I_i is the distance from the assumed template, and dI_i the distance from the candidate binary mask I_i

algorithm starts with a set of (2.2.1) and (2.2.2). The individuals

then evaluated and consequently ranked as described in sections 2.5 and 2.6. Fifty percent of the ranked individuals are consequently inputted to the cross over module and the other 50% is selected randomly from the best-fit individuals of the previous generation. A random bit mutation operator randomly mutates single bits from the entire population (the mutated offsprings are represented by cross lines). At this point, the process is repeated as this cycle continues until complete convergence or a single outstanding individual map is found.

2.7 The block diagram representation

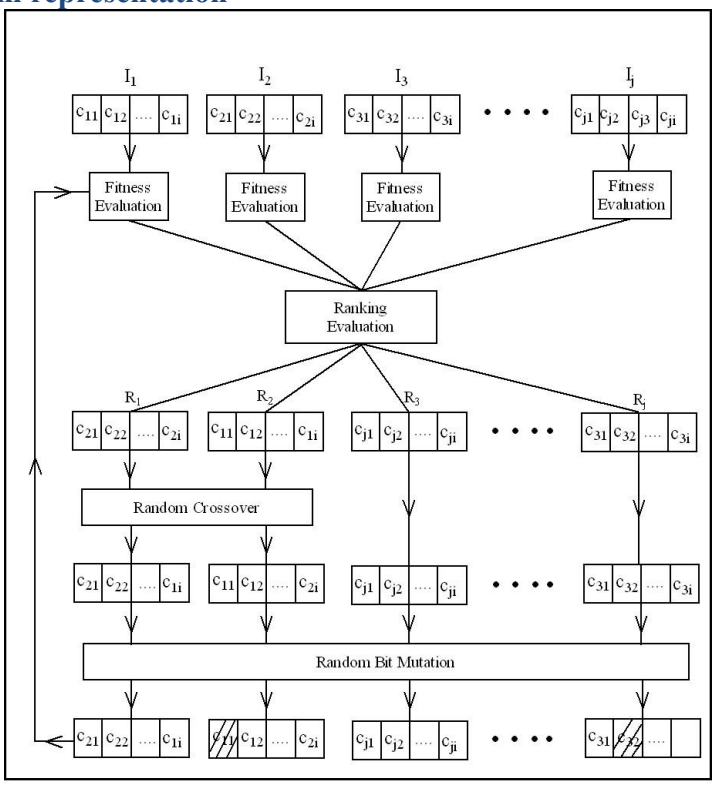

Figure 2.7.1: Block Diagram Implementation of the Genetic Optimization Process

3. Results, discussions and future recommendations

3.1 Results and discussions

Four simulations were conducted. In each simulation, a 128x128 gray scale input image was used. The input image that is used is made of two main components, a background and a target with different textural, morphological and size characteristics. The simulation results are presented in **Figure 3.1.1** to **Figure 3.1.8**. There are two Figures per simulation. The first Figure in each case shows a partial summary of the adaptation process by showing the best and worse four candidates of the first population, the best and worse four candidates of the last population, the best of the first and last population, the input image that is used and the input target that is sought. The second Figure in each case shows a set of goodness of fit histograms for the first, the fiftieth and the hundredth generations. The adaptation parameters; size of the population (1000), and bit mutation rate (0.001) are chosen based on the results of previous studies, Davis[7], and Dejong[8,9]. The linear ranking and decrement scheme as well as the random Crossover operator are unique to the set of Genetic Optimization Algorithms developed in this paper. The objective in the first three simulations is to demonstrate the ability to perform textural segmentations using the proposed Genetic Optimization Algorithm. In the first simulation, after 100 populations the algorithm found the following results: the mean and variance of the best candidate segmentation solution were 39.7986 and 49.7543 compared to 40 And 49 for the target. The area of the best candidate segmentation was 2266 pixels compared 2269 for the target. The Chi-square, Shape and Size evaluation were: 0.794, 0.9973, 0.9967 respectively. As a result, the overall goodness of fit evaluation for the best candidate segmentation is 0.9935, and the results are shown in **Figure 3.1.1**. In **Figure 3.1.2**, the convergence characteristics for this simulation are displayed. In **Figure 3.1.1**, we see that in the first population, candidate segmentations are created all over the image field. This is also verified in Figure 3.1.2 by observing that, while we have candidates with very poor goodness of fits (0.1 to 0.3), we also have a significant number of good average candidates (0.5), and other good candidates well above the average mark (0.6 to 0.7). By the hundreth generation, the goodness of fit histogram has evolved considerably towards the exponential profile depicted in **Figure 3.1.2**. The convergence characteristics shown in **Figure 3.1.2** suggest that in this simulation example, the Genetic Optimization process does find better and evolved candidates to the segmentation problem of interest. In addition, visual investigation of the results in **Figure 3.1.1** reveals that the best four candidate image masks proposed in the first population although positioned correctly within the image field do not have the morphological characteristics of the target being sought. This observation seems to indicate that the algorithm may be able to identify the textural characteristics of the target within a few generations while still lacking knowledge of other possible characteristics such as its morphology and size.

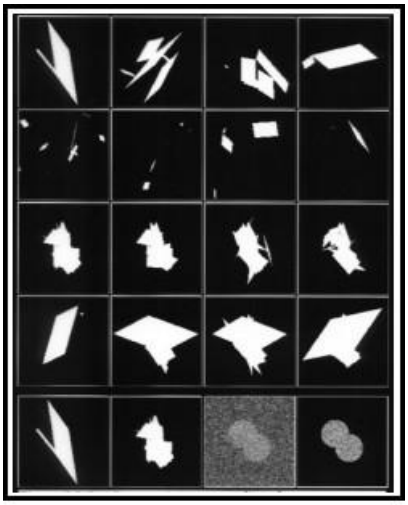

Figure 3.1.1: Simulation number one
Segmentation results of experiment one

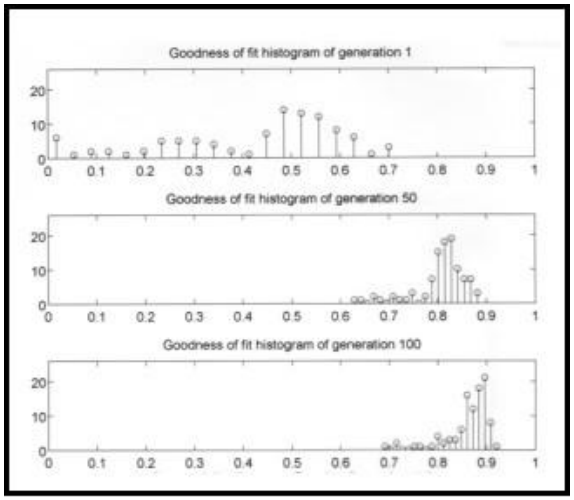

Figure 3.1.2: Simulation One
Convergence Characteristics

However, the best four candidate image masks proposed in the last population, in addition to their position within the image field do have morphological, size and position characteristics within the image field similar to that of the target. These features of the Genetic Optimization technique developed in this paper are significant because, in the case of early detection of breast, and teeth tumors for instance, the textural, morphological and size characteristics may be built a priori as part of the optimization process through a training procedure. The training can be performed on an ensemble of images from a random process model characteristic of the segmentation problem of interest. In this case, the algorithm would be able to identify within the image field, correlations between a priori sample features and a given candidate, even when these correlations could not be identified with the naked eye. Late detection of the cancer types mentioned earlier could be prevented with such technique. In the second simulation, we take away our knowledge of the morphological characteristics of the target sought in order to investigate the algorithm's ability to find a good candidate segment with sole knowledge of its textural characteristics. The results are shown in **Figure 3.1.3** and **Figure 3.1.4**. The convergence characteristics in this example are almost similar as in the previous example, as seen in **Figure 3.1.4**.

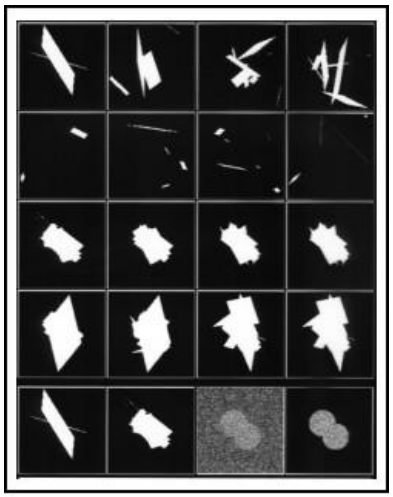

Figure 3.1.3: Simulation Two
Segmentation Results

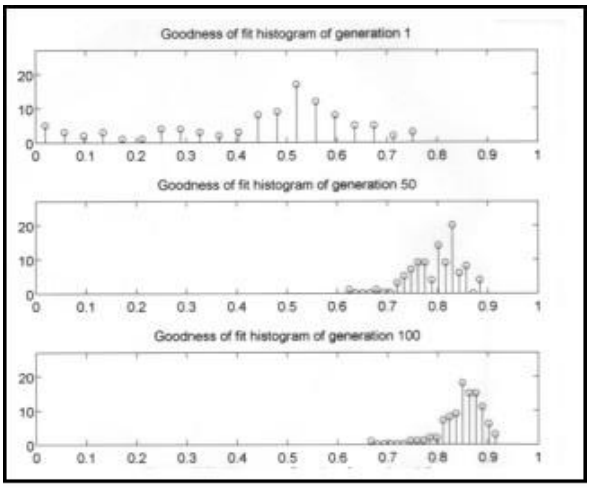

Figure 3.1.4: Simulation Two
Convergence Characteristics

However, the progression of the goodness of fit histograms from a fairly average profile to an exponential profile is not as pronounced as in the first case. The convergence characteristics shown in **Figure 3.1.4** do suggest that the adaptation is successfully producing evolved candidate segments but not in as a smooth a manner as in the previous example, as shown by the results in **Figure 3.1.3**. This behavior may be due to the fact that without further knowledge of the characteristics of the target, the Genetic Optimization process may be limited in its ability to produce an outstanding candidate. The behavior may also be hinting at the possibility that the Genetic Optimization process may require a lot of a priori approximate information about the class of segmentation problems of interest. Visual observation of the results displayed in **Figure 3.1.3** suggests that despite limited a priori knowledge of the morphological characteristics of the target, the algorithm chooses an optimum candidate segment, which has some knowledge of the morphological characteristics of the target sought, although not as defined as in the first example. This simulation example gives us a way to predict how we may be able to use the Genetic Optimization technique in cases where we may not know how large and of what shape we expect a tumor to be. The reader may note that in tumor diagnostic cases for instance, we may not know if "tissue malignancy" is present within the image field. In addition, when "tissue malignancy" is present, we may not know where it may reside within the image field. All we may know before hand are the inherent geometrical and textural

characteristics of "tissue malignancy" based on the class of segmentation problems of interest. The ability for the Genetic Optimization technique to suggest the coordinate or position of a group of "malignant tissue" within the image field is significant as it directly addresses the potential for this technique to be used for "positional detection" even when no "malignant tissue" can be identified with the naked eye through a mammogram, or a radiograph. In the third simulation, we allow the target to change in its morphological characteristics. The reader will note from **Figure 3.1.5** that a piece of the target is taken away, and the objective is to investigate whether the Genetic Optimization process takes this change into account and attempts to factor the change in the candidate solutions it proposes. Visual observation of the results in **Figure 3.1.5** and **Figure 3.1.6** suggests that the Genetic Optimization process found it difficult to completely include the added knowledge in morphological characteristics of the segmentation being sought. Although careful observation of the optimum solution proposed reveals that it does contain some knowledge of the added morphological characteristics (the best solution proposed has a small cut on the top right). In thinking about possible reasons why the morphological characteristics may not be included as desired, we look at the encoding mechanism that is used to go from strings of binary digits to binary parallelograms, and finally to binary image masks. We realize that the nature of the solutions we obtain would inherently be limited in flexibility by the nature of the encoding mechanism we use. The encoding mechanism therefore plays an important role both in the smoothness and the geometrical nature of the solutions we obtain. This observation hints at the fact that in order to create flexible Genetic Optimization techniques that may be applied to image segmentation, it may be worth incorporating encoding mechanisms that are characteristic of the segmentation problems of interest. In other words, the nature of the segmentation problems of interest may be intimately tied to the nature of the encoding mechanism we use. Finally, we look at the behavior and performance of the Genetic Optimization process developed in this paper in a real case. In this simulation example, we consider realizations of a random process characteristic of human teeth and we use sample realizations of this process as a training set to create a priori information about the textural nature of human teeth versus human gums. The objective in this simulation is to investigate the potential utility of the Genetic Optimization process in differentiating between teeth and gums, healthy teeth and gums versus possibly diseased ones. In this particular example, we investigate the simpler case of segmenting a tooth from its gum.

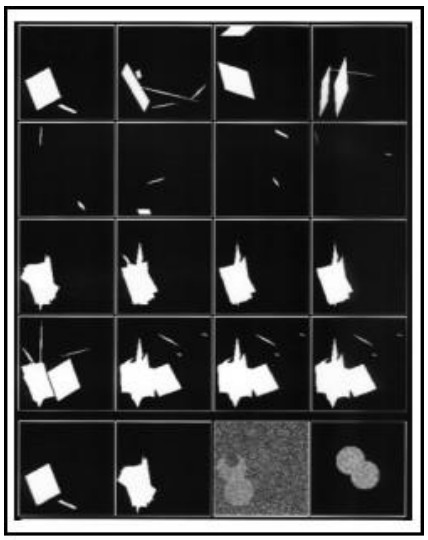

Figure 3.1.5: Simulation Three
Segmentation Results

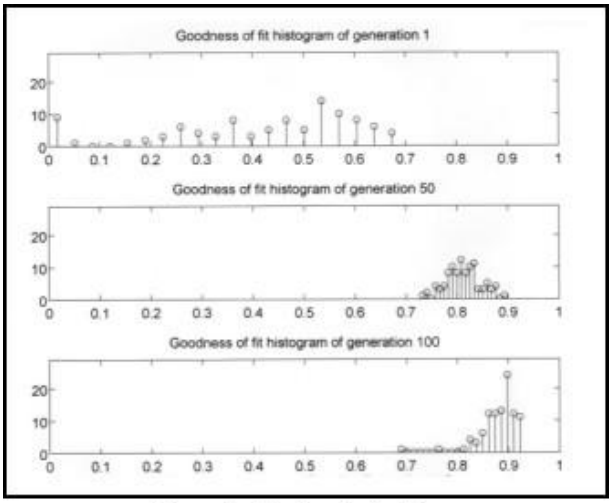

Figure 3.1.6: Simulation Three
Convergence Characteristics

Visual observation of the results of **Figure 3.1.7** and **Figure 3.1.8** reveals that the segmentations results proposed after 100 generations were significantly better than that of the first generation. In particular, the reader

may note that the position of the upper section of the tooth was correctly identified. In addition, the overall characteristics of the upper section of the tooth were also found. However, the algorithm could not correctly describe the shape and size of the upper tooth section. This is a potential difficulty since in most real segmentation cases, we may know very little about the shape and size of the segmentation being sought. In most cases, it may even be difficult to correctly and accurately describe the textural characteristics of the segmentation being sought. This simulation example allows investigating the potential behavior of the Genetic Optimization process in those cases. The results in **Figure 3.1.7** suggest that the Genetic Optimization Algorithm developed in this paper could benefit from better morphological and textural representations of the segmentation problems of interest. The results also suggest that in this specific example, the flexibility in encoding mechanism significantly affects the nature and smoothness of the segmentation solutions.

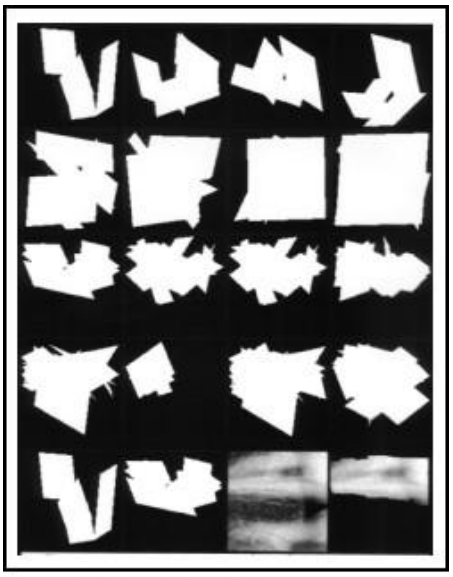

Figure 3.1.7: Simulation Four Segmentation Results

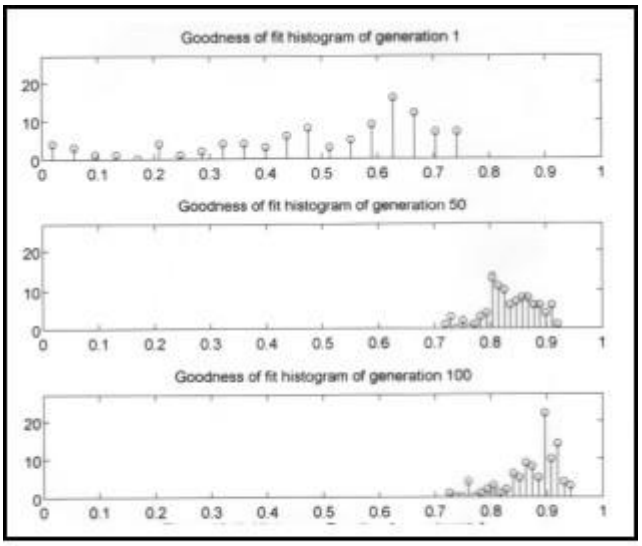

Figure 3.1.8: Simulation Four Convergence Characteristics

3.2 Conclusions

The utility of the Genetic Optimization process was investigated. The results described above illustrate the potential usefulness of such technique in configurational optimization problems. The algorithm's effectiveness in Medical Image segmentation was also investigated. In this case, the results suggest that the nature of the solutions we obtain may be intimately tied to the nature of our prior knowledge about the medical image segmentation of interest. The type of encoding mechanisms used also limits the nature of the solutions. The encoding mechanism that is currently being used in the Genetic Optimization process is not as flexible as desired to obtain smooth solutions when applied to the class of problems of interest. Issues of flexibility in the encoding mechanism become extremely important when only minor changes are needed in order to effect significant improvements in the genetic make up of a particular individual segmentation solution. In addition, a bit mutation module, which allows to controllably affect the size, shape and texture of individual candidates within any given population may be more suited for the class of segmentation problems considered in this paper. The Genetic Optimization technique developed in this paper introduces some added parameters to the optimization process when compared to its predecessors. A random rather than a fixed Crossover operator was introduced as part of the Genetic Optimization process. Random Crossover introduces more flexibility in the mating of candidate segments and affects the nature of the resulting solutions. Also a linear fitness and decrement scheme was introduced which biases the parent selection process so as to allow individual segmentation solutions with

poor goodness of fit but potentially good genetic information to make it to the adaptation process. These features although they depart from the basic Genetic Algorithms proposed in Davis[7], Dejong[8,9], Holland[11] and Lawrence[12], do not allow us to understand how the overall convergence characteristics of the process is affected.

Acknowledgments

The author wishes to thank Dr. Walter S. Kuklinski from The University of Massachusetts at Lowell, currently at The Mitre Corporation, for his mentoring and useful insights in developing the Genetic Optimization Algorithm presented in this paper. The author also wishes to thank Dr. Alfred G. Noel, from the Mathematics Department at the University of Massachusetts, Boston for his many useful suggestions while developing this algorithm.

References

1. Beck J, Prazdny L, Rosenfeld A. A theory of textural segmentation, Human and Machine Vision. New-York Academic Press, 1983.
2. Blostein D, Ahuja N. Representation and three-dimensional interpretation of image texture: An integrated approach, In Proc. First Int. Conf. Computer Vision 1987, 444-449.
3. Byrne LC. Iterative image reconstruction based on cross entropy minimization. IEEE Transactions on image processing 1993; 2(1): 96-103.
4. Clark M, Bovik AC. Texture Segmentation using Gabor modulation/demodulation. Pattern Recognition Lett 1987; 6, 261-267.
5. Coggins JM., Jain AK. A spatial filtering approach to texture analysis. Pattern Recognition Lett 1985; 3, 195-203.
6. Cross GC., Jain AK. Markov random field texture model. IEEE Trans. Pattern Anal. Machine Intell., PAMI 1983; 5, 25-39.
7. Davis EG. Genetic Algorithms, Reading Massachusetts. Addison-Wesley, 1989.
8. Dejong K. Learning with Genetic algorithms: An overview, Machine Learning. 1988; 3, 212-138.
9. Dejong K. A 10 year prospective, in Proc. Int. Conf. Genetic Algorithms, Their Applications, 1985, 169-177.
10. Keren D, Kooper D. Describing complicated objects using polynomials. IEEE Trans. on Pattern Analysis and machine Intelligence 1994, 16.
11. Holland JH. Adaptation in natural and artificial systems, Ann Arbor, University of Michigan Press, 1975.
12. Lawrence D. Handbook on Genetic Algorithms, New-York, Van Nostrand Reinhold, 1991.
13. Lawrence D, Steenstrup M. Genetic Optimization and Simulated annealing, In Genetic Algorithms and Simulated Annealing, L. Davis, Ed. London, Pitman, 1987, 1-11.
14. Leehter Y, Sethares WA. Non linear parameter estimation via Genetic Algorithm. IEEE Trans, Signal Processing 1994; 42, 4.
15. Turner MR. Texture discrimination by Gabor functions. Biol Cybern 1986; 55, 71-82, 1986.
16. Vignaux GA., Michalewicz Z. A Genetic Algorithm for the linear transportation problem. IEEE Trans. Syst. Man Cybern 1991; 21, 445-452.