1998

Image segmentation using genetic algorithm and morphological operations

Ming Yu
Iowa State University

Image segmentation using genetic algorithm and morphological operations

by

Ming Yu

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Electrical Engineering

Major Professor: Lalita Udpa

Iowa State University

Ames, Iowa

1998

Image segmentation using genetic algorithm and morphological operations

Ming Yu

Major Professor: Lalita Udpa

Iowa State University

Image segmentation is a fundamental component of picture processing and image analysis. Segmentation of an image entails the division or separation of the image into regions of similar attributes. The most basic attribute for segmentation is the image intensity (luminance for a monochromatic image). Several classical methods for image segmentation exist and it is well known that these methods are more or less heuristic and specific to a particular application.

Genetic Algorithms (GA) are stochastic search methods, the functioning of which is inspired by laws of genetics, natural selection and evolution of organisms. Their main attractive characteristic is the ability to deal with hard combinatorial search problems efficiently, where parallel exploration of the search space, eliminates to a large extent the possibility of getting stuck in the local extrema. The basis of the theory is that individuals tend to pass on their traits to their offspring and the fittest of the individuals tend to have more offsprings. In effect, the tendency is to drive the population towards favorable traits. Over long periods of time, entirely new species are produced which are better adapted to a particular ecological condition.

This thesis proposes a simple and robust method for image segmentation that is based on the application of Genetic Algorithm and Mathematical Morphology. The image

is divided into nonoverlapping subimages and the genetic algorithm is applied to each subimage, starting with initial random populations. Each individual of the population is evaluated using an appropriate fitness function. The best-fit individuals are selected and mated to produce offsprings to form the next generation. Morphological operations are used to produce the next generation along with the crossover and mutation operators. The algorithm converges to yield the final segmented subimage. These segmented subimages then are combined to form the final result. The feasibility of applying genetic algorithm and morphological operations to an image segmentation problem is evaluated and results are presented and discussed.

Graduate College
Iowa State University

This is to certify that the Master's thesis of

Ming Yu

has met the thesis requirements of Iowa State University

Signature redacted for privacy

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This thesis presents an image segmentation procedure that uses genetic algorithm and mathematical morphology for optimizing a criterion function. The image is divided into subimages and segmentation algorithm is applied to each subimage, starting with initial random populations. Each individual of the population is evaluated using an a fitness function. The best-fit individuals are selected and mated to produce offsprings that form the next generation. The morphological operation is used to produce the next generation along with the crossover and mutation operators. The algorithm converges to yield the segmented subimages. These segmented subimages then are combined to form the final result. The performance of genetic algorithm and morphological operations to an image segmentation problem is evaluated with respect to various parameters and the results are presented and discussed.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my major professor Dr. Lalita Udpa for her guidance, encourage and kindness during my graduate study. She provided new ideas and suggestions that help me with my study. Her motivation and enthusiasm are mainly responsible for my achievements.

I would like to sincerely thank Dr. Satish Udpa, Dr. Kai-Ming Ho and Dr. Gerald Sheble for being on my committee and being supportive and encouraging.

I am proud of my family for your love, support and encouragement.

Special thanks to my colleagues and friends for their cooperation and help.

CHAPTER 1. INTRODUCTION

A digital image is represented by two-dimensional array or matrix of numbers. Digital image processing is the manipulation of images by computers. It has many applications in diverse areas such as telecommunication, medical imaging, graphic arts, remote sensing etc. In the past several years, many new digital image processing and analysis techniques have been developed. Image processing system plays a very important role in scientific, industrial, medical and space applications. Present trends indicate a continuation of the explosive growth of digital image processing applications well into the next century.

At its most basic level, digital image processing requires a computer upon which to process images and two pieces of special input/output devices: an image digitizer and an image display device. A typical image analysis system performs the following operations: (1) acquisition, (2) storage, (3) processing, (4) communication, and (5) display. The basic image processing operations consist of (1) formation, (2) restoration, (3) enhancement, (4) coding, (5) compression and (6) analysis.

Digital image processing comprises a broad range of hardware, software, and components ranging from simple image enhancement to more complex processing and classification of image data [1]. A schematic of the overall system is given in Figure 1.1.

As shown in Figure 1.1, the first step in image processing is image acquisition: the sensor system specially designed to view a scene and provide a digital representation; or conversion of image data from an existing medium into a digital format (A-D conversion) e.g. scan and

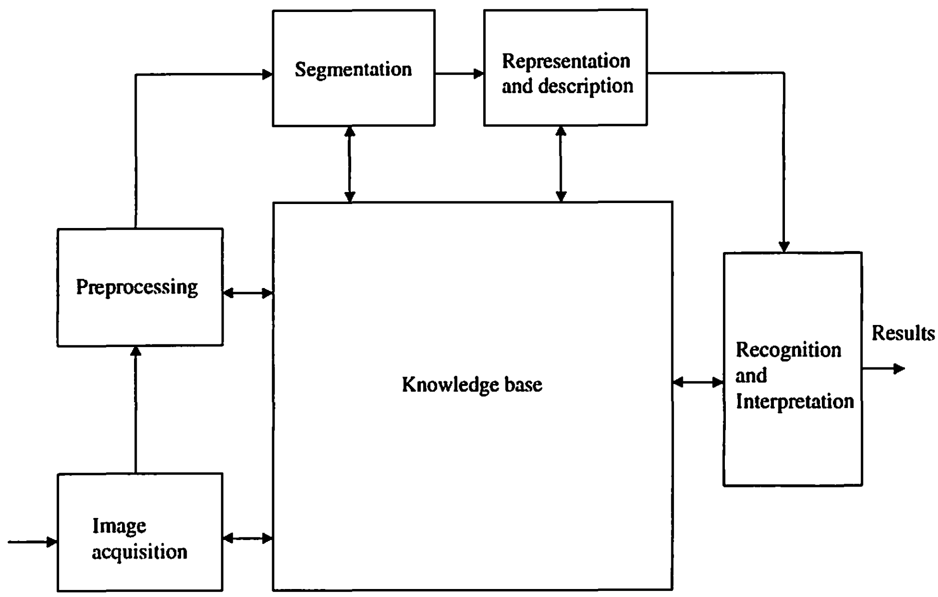


Figure1.1 Fundamental steps in digital image processing

digitize an aerial photograph. Each digital image formation system introduces a geometrical distortion, noise, and nonlinear transformations.

The next step deals with preprocessing of the image which typically involves procedures for image restoration and image enhancement. Digital image restoration is commonly defined as the processing of the measured image data to compensate for artifacts introduced by the image acquisition system. Digital image enhancement tries to improve the image quality by enhancing contrast, removing noise etc. Due to the fact that digital images usually require a very large amount of memory for their storage, it is very important to reduce the memory requirements for image storage and transmission. Digital image compression and coding reduce and compress the image information content by taking the advantage of the information redundancy existing in an image.

The third step deals with image analysis. Image analysis is the interpretation of the information content in an image data. It usually consists of image segmentation, image description (feature extraction), and image recognition (classification). Image segmentation partitions an input image into its constituent parts or objects. It entails division of an image into regions of similar attributes. Several methods have been developed for image segmentation including thresholding, clustering, region method, boundary detection, texture method and adaptive method. Image description, also called feature selection, involves extraction features that are basic for differentiating one class of objects from another. These features include spatial features, edges and boundaries, textures etc. Image recognition is the process that labels an object in the image based on information provided by the features. In this step, objects with identical features are grouped together under a certain class. Techniques for recognition include clustering, neural networks, decision trees, and spanning trees.

Due to the variety of application of digital image analysis, a multitude of algorithms has been presented in literature in all the areas mentioned above. Among these, algorithms for object identification and image segmentation are of prime interest. In medical diagnostic imaging, image segmentation is mainly used to process different images obtained using multiple projection methods such as CT, MRI, PET. These images are used to detect tumors and other disorders. Several classical methods for image segmentation such as thresholding, clustering, edge detection and thinning have been developed for segmenting different types of images.

This thesis proposes a new and simple approach for image segmentation that is based on genetic algorithm and mathematical morphology. Although GA (Genetic Algorithm) techniques have been applied extensively in optimization problems including some areas in image processing. The application of genetic algorithm in combination with morphological operation in image segmentation is investigated and the results are very encouraging.

The rest of the thesis is organized as follows. Chapter 2 describes the general methods of image segmentation. The basic morphological operations are explained in chapter 3. In chapter 4, an introduction to genetic algorithms has been provided and their important elements are explained. The new image segmentation algorithm based on genetic algorithms and morphological operations is discussed in chapter 5. The results of implementation of the algorithm developed in this thesis on simulated images are described and discussed in chapter 6. These results help in choosing the parameters of the algorithm as well as improving the performance of the algorithm. A discussion of the performance of the proposed approach and conclusive remarks are finally presented in chapter 7. The scope for future work is also pointed out in this chapter.

CHAPTER 2. IMAGE SEGMENTATION

2.1 Introduction

Image segmentation is an important and a low-level task in digital image processing. Segmentation of an image entails the division or separation of the image into regions of similar attributes. The most basic attribute for image segmentation is image amplitude-luminance for monochrome images and color component for color images. All subsequent interpretation tasks — feature extraction, object recognition and classification — depend heavily on the quality of the segmentation process.

There is no fundamental theory of image segmentation. In other words, no general methods have been found that perform adequately across a diverse set of imagery. As a consequence, no single standard method of image segmentation has emerged. Rather, there are collections of methods that have received some degree of popularity. It would be useful to have some means of assessing their performance. Qualitative criteria for good image segmentation are [2][3]: 1) the segmented regions should be uniform and homogeneous with respect to some characteristic, such as gray level or texture, 2) region interiors should be free of holes and region boundaries should be smooth and spatially accurate, and 3) adjacent regions should be differing significantly based on the characteristic on which they are uniform. If one represents this criteria set in terms of a hypothetical function, then the problem of good segmentation is one of optimizing this objective function by selecting appropriate segmentation parameters.

Generally speaking, there are three stages in image segmentation. The first is *image preprocessing*. In this stage, the image is visually improved. The second stage is *initial object discrimination*, where objects are grossly separated into groups with similar attributes. The third stage is *object boundary cleanup*, where object boundaries are reduced to single-pixel widths. In this final stage, noise clutter and other artifacts in the image are removed.

There are several generic methods for the first and second stages of the image segmentation process. These techniques generally come from the image-enhancement class of digital image processing operations. There are also techniques of image morphological processing that are frequently used for the third stage boundary cleanup operations. Although it is not feasible to describe all the details of all the methods here, the fundamentals of some of the methods are discussed.

2.2 Amplitude Segmentation Methods

Several image segmentation methods are based upon the thresholding of luminance or color components of an image.

2.2.1 Bilevel luminance thresholding

Many images can be characterized as containing some object of interest of reasonably uniform brightness placed against a background of differing brightness. A distinguish feature that can be utilized to segment the object from its background. If an object of interest is white against a black background, or vice versa, it is an easy task to segment the object from the background. But when the observed image is subject to noise, especially when the object and the background assume some broad range of gray scales, it is a little more difficult. If the background is nonuniform, then alternate approaches to segmentation have to be developed..

There are several analytic approaches for selecting a luminance threshold. One method is to set the gray scale threshold at a level such that the cumulative gray scale count matches an *a priori* assumption of the gray scale probability distribution. Another method is to set the threshold at the minimum point of the histogram between its bimodal peaks.

If the background of the image is nonuniform, it is necessary to adapt the luminance threshold to the mean luminance level. This can be achieved by subdividing the image into small blocks and determining the best threshold level for each block.

2.2.2 Multilevel luminance thresholding

A recursive multilevel thresholding method can be used to achieve effective segmentation in some classes of image. In the first step of the process the image is thresholded to separate brighter regions from darker regions by locating a minimum between luminance modes of the histogram. Then histograms are formed of each of the segmented parts. If these histograms are not unimodal, the parts are thresholded again. This process continues until the histogram of a part becomes unimodal.

2.3 Clustering Segmentation Methods

Consider a vector of measurement at each pixel in an image. The measurement could be the neighborhood feature such as the moving window mean, standard deviation, or it could be point color components. If the measurement set is to be effective for image segmentation, data collected at various pixels within a segment of common attribute should be similar. In this way, the data are tightly clustered in an N-dimensional measurement space. The N-dimensional measurement space is then subdivided into mutually exclusive compartments where each compartment envelopes typical data cluster for each image segment. The clustering segmentation concept is simple but the computational effort is intensive.

2.4 Region Segmentation Methods

2.4.1 Region growing

Region growing is one of the conceptually simplest approaches to image segmentation; neighboring pixels of similar amplitude are grouped together to form a segmented region. In practice, constraints must be placed on the growth pattern to achieve acceptable results.

2.4.2 Split and merge

Split and merge image segmentation method is based on the quad tree data representation whereby a square image segment is broken (split) into four quadrants if the original image segment is nonuniform in attribute. If four neighboring squares are found to be uniform, then they are replaced (merged) by a single square composed of the four adjacent squares.

The basic split and merge process tends to produce rather "blocky" segments because of the rule that square blocks are either split or merged. A modification of this process has been proposed by Horowitz and Pavlidis [4] where adjacent pairs of regions are merged if they are sufficiently uniform.

2.5 Boundary Detection

One way to segment the image is to detect the boundary of each region where there is a significant change in attribute across the boundary. Edge detection is one of the methods to detect boundaries. After the edge map has been found, morphological operations can be used to thin the edge.

A detected boundary may often be broken if the image is noisy or if the region attributes are similar between regions. In this case, edge linking techniques are very useful to bridge short gaps in such a region boundary.

There are several edge linking methods: (1) curve fitting edge linking, (2) heuristic (Roberts) edge linking, (3) Hough transform edge linking, etc. All these methods have their advantages and disadvantages. Depending on the properties of the image under consideration, one may be better than the others.

2.6 Texture Segmentation

One approach to texture segmentation is to compute some texture coarseness measure at all image pixels and then detect changes in the coarseness of the texture measure [5]. Another approach is to detect the transition between regions of different texture. The basic concept of texture edge detection is identical to that of luminance edge detection. A histogram thresholding method of texture segmentation also has been proposed [6].

2.7 Adaptive Segmentation

Selecting the right method and the appropriate set of algorithm parameters is the key to efficiently segmenting the image. But no segmentation method can automatically generate an "ideal" segmentation result in one pass in a range of different images encountered in real world applications. If the algorithm can not adapt to the variations in unstructured scenes, it will eventually yield poor results [7].

There are several factors that make the parameter adaptation process very difficult:

1) The number of parameters present in a typical segmentation algorithm is very large
2) The complex and nonlinear interactions among these parameters make it impossible to model their behavior.
3) The objective function that represents the segmentation quality varies from image to image since variations between images cause changes in the segmentation results.
4) The definition of the objective function can be a subject of debate because there is no single, universally accepted measure of segmentation quality.

There is a need of an adaptive segmentation technique that can efficiently search the complex space of parameter combinations and locate the optimal results. There are adaptive threshold selection techniques for segmentation but these techniques do not accomplish any learning.

Genetic algorithms are designed to efficiently locate an approximate global maximum in a search space. They have the attributes described above and show great promise in solving the parameter selection problem in the image segmentation.

The key elements of an adaptive image segmentation system are:

1) A closed-loop feedback control technique, which provides an adaptive capability.
2) A learning subsystem that optimizes segmentation performance and accumulates segmentation experience over time to reduce the effort needed to optimize subsequent images.
3) Image characteristics are represented and manipulated using genetic structure.
4) Image segmentation performance is evaluated using multiple measures of segmentation quality. Genetic Algorithm is an appropriate technique for the image segmentation. The combination of Genetic Algorithm with other search techniques will result in an efficient hybrid segmentation algorithm. These techniques include hill climbing (HC), morphological operation, etc.
5) The learning subsystem is very fundamental and is independent of segmentation algorithms and sensor data (visible, infrared, ultrasonic, laser, etc.). The performance of the overall system is limited by the capabilities of the segmentation algorithm, but the results of a given image are optimal based on the evaluation criteria.

CHAPTER 3. MATHEMATICAL MORPHOLOGY

3.1 Introduction

In the context of mathematics and signal processing, "morphology" is used to describe a branch of non-linear transformations and filtering methodologies. Image morphology pertains to the study of the structure of the objects within an image. Morphological operations work to clarify the underlying structure of objects. This is done by further simplifying object boundaries to their most rudimentary single-pixel-wide outlines or skeletons. These outline and skeletal forms yield an object's most primitive essence. The beauty and utilities of mathematical morphology are in its set theory based formulation, which directly deals with shape and structure.

It is very easy to apply mathematical morphology to image processing, especially to binary images. This is because that the black and white pixels naturally form two sets, the object and the background. The basic morphological operations are dilation and erosion. These operations enlarge or reduce an object in an image, based on another object called structure element. Although the basic operations are simple, they and their variants can be concatenated to produce much more complex effects, i.e. a wide range of morphological operations can be achieved by combining dilation and erosion with various structural elements.

Mathematical morphology can also be applied to gray scale image processing with 3-dimensional gray scale image surface presented as sets. One way to represent the sets of a gray scale image is via a multilevel function, which is treated as a stack of binary corrections processed individually by morphological operations [8]. Another set representation is umbra of a gray scale image [9]. The umbra can be intuitively thought as the 3-D infinite space "under" the image surface.

3.2 Binary Morphology

The binary image is assumed to be composed of pixels that have one of two brightness values, either black (0) or white (255). Binary image morphological process works much like the spatial convolution group process. The morphological process moves across the input image, pixel by pixel, placing resulting pixels in the output image. At each input pixel location, the pixel and its neighbors are logically compared against a *structuring element* to determine the output pixel's logical value.

The structuring element is an array of logical values. It is generally composed of square dimension of size 3×3, 5×5, and sometimes larger, depending on the application. Each logical value can take on the value 0 or 1 (off or on), or a third state of X which is the "don't-care" state.

The generalized implementation of the binary morphological operation is commonly referred to as the *hit and miss transform.* When the structuring element values match their respective input pixel values, we call the evaluation a "hit". Otherwise, it is a "miss". The hit and miss transform provides a convenient way to define numerous morphological operations [10].

The two fundamental morphological operations are erosion and dilation. The erosion operation uniformly reduces the size of objects in relation to their background. The dilation operation — the inverse of the erosion operation — uniformly expands the size of the objects. Erosion and dilation operations are used to eliminate small-image object features, such as noise spikes and ragged edges. Various combinations of these operations provide the basis for many additional operations.

3.2.1 Erosion

Simple erosion is the process of eliminating all the boundary points from an object, leaving the object smaller in area by half width of the structuring element all around its perimeter. If the object narrows to less than three pixels thick at any point, it will be disconnected (into two objects) at that point. Mathematically, erosion is the morphological transformation that combines two sets using the vector subtraction of set element [11]. Erosion is useful for removing segmented image objects that are too small to be of interest.

There are a number of definitions for erosion, which can be proved to be equivalent. General erosion of image **B** by structuring element **S** is denoted by **B** ⊗ **S** and is defined by

$$\mathbf{E} = \mathbf{B} \otimes \mathbf{S} = \{x, y \mid \mathbf{S}_{xy} \subseteq \mathbf{B}\} \tag{3.1}$$

where $\mathbf{S}_{xy}$ is the set **S** translated to (x, y).

The binary image **E** that results from erosion is the set of points (x, y) such that if **S** is translated so that its origin is located at (x, y), then it is completely contained within **B**. The structuring element **S** may be visualized as a probe which slides across the image **B**, testing the spatial nature of **B** at each pixel. Figure 3.1 shows an example of erosion. Notice that the pixels in the second row are not present in the image after erosion because the one-pixel wide horizontal line can not contain the two-pixel wide structuring element **S**. A more practical example shown in Figure 3.2 illustrates that the erosion basically shrinks the object by the size of structuring element.

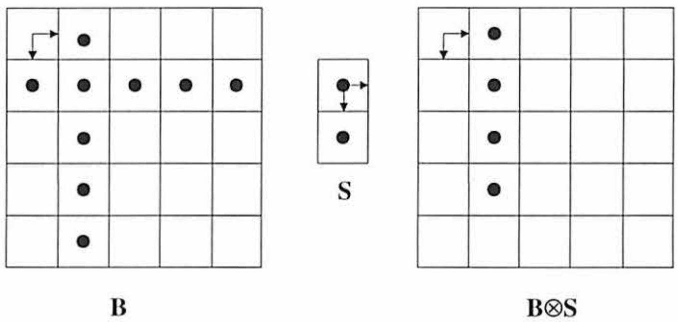


Figure 3.1 A basic example of erosion

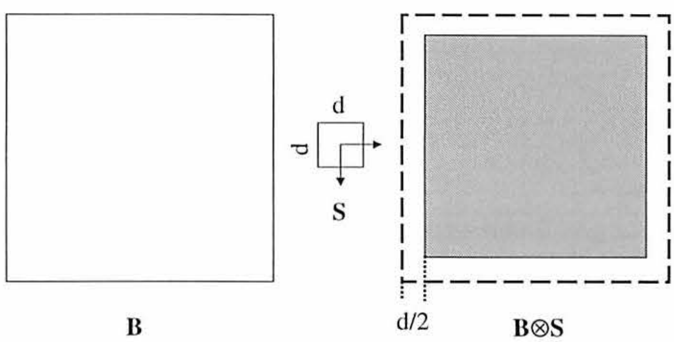


Figure 3.2 A practical example of erosion

3.2.2 Dilation

Simple dilation is the process of incorporating into the object all the background points that touch it, leaving it larger in area by that amount. If the object is circular, its diameter increases by half width of the structuring element with each dilation. If two objects are separated by less than three pixels at any point, they will become connected (merge into one object) at that point. Mathematically, dilation is a morphological transformation that combines two sets using vector addition of set element [11]. Dilation is useful for filling holes in segmented objects.

General dilation of image **B** by structuring element **S** is denoted by $\mathbf{B} \oplus \mathbf{S}$ and is defined by

$$\mathbf{D} = \mathbf{B} \oplus \mathbf{S} = \{x, y \mid \mathbf{S}_{x,y} \cap \mathbf{B} \neq 0\} \tag{3.2}$$

where $\mathbf{S}_{xy}$ is the set **S** translated to (x, y).

That is, the binary image **D** that results from dilation is the set of points (x, y) such that if **S** is translated so that its origin is located at (x, y), then its intersection with **B** is not empty. Figure 3.3 illustrates an example of dilation. Figure 3.4 shows a more practical example.

3.2.3 Properties of dilation and erosion

In morphological dilation, the roles of the sets **B** and **S** are symmetric, i.e., the dilation operation is commutative

$$\mathbf{B} \oplus \mathbf{S} = \mathbf{S} \oplus \mathbf{B} \tag{3.3}$$

But, in general, erosion is not commutative

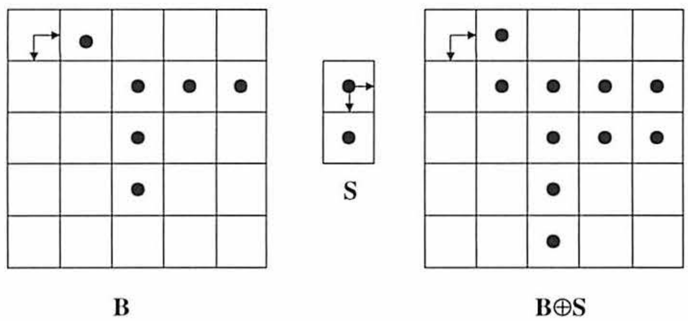


Figure 3.3 A basic example of dilation

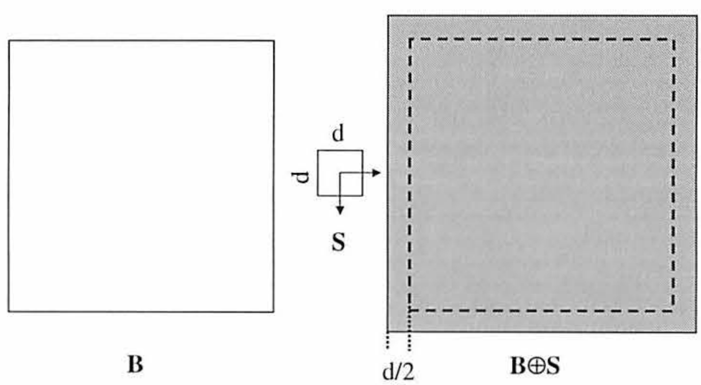


Figure 3.4 A practical example of dilation

$$\mathbf{B} \otimes \mathbf{S} \neq \mathbf{S} \otimes \mathbf{B} \tag{3.4}$$

Dilation and erosion are increasing operations in the sense that, if set $\mathbf{A} \subseteq \mathbf{B}$, and set $\mathbf{C}$ is the structuring element, then

$$\mathbf{A} \oplus \mathbf{C} \subseteq \mathbf{B} \oplus \mathbf{C} \tag{3.5a}$$

$$\mathbf{A} \otimes \mathbf{C} \subseteq \mathbf{B} \otimes \mathbf{C} \tag{3.5b}$$

Dilation and erosion are opposite in effect: dilation of the background of an object behaves like erosion of the object. This statement can be qualified by the duality relationship.

$$\overline{\mathbf{A} \otimes \mathbf{B}} = \overline{\mathbf{A}} \oplus \mathbf{B} \tag{3.6}$$

Dilation and erosion of the intersection and union of the sets obey the following relations:

$$(\mathbf{A} \cap \mathbf{B}) \oplus \mathbf{C} \subseteq (\mathbf{A} \oplus \mathbf{C}) \cap (\mathbf{B} \oplus \mathbf{C}) \tag{3.7a}$$

$$(\mathbf{A} \cap \mathbf{B}) \otimes \mathbf{C} = (\mathbf{A} \otimes \mathbf{C}) \cap (\mathbf{B} \otimes \mathbf{C}) \tag{3.7b}$$

$$(\mathbf{A} \cup \mathbf{B}) \oplus \mathbf{C} = (\mathbf{A} \oplus \mathbf{C}) \cup (\mathbf{B} \oplus \mathbf{C}) \tag{3.7c}$$

$$(\mathbf{A} \cup \mathbf{B}) \otimes \mathbf{C} \supseteq (\mathbf{A} \otimes \mathbf{C}) \cup (\mathbf{B} \otimes \mathbf{C}) \tag{3.7d}$$

Dilation and erosion of a set by the intersection of two other sets satisfy these containment relations:

$$\mathbf{A} \oplus (\mathbf{B} \cap \mathbf{C}) \subseteq (\mathbf{A} \oplus \mathbf{B}) \cap (\mathbf{A} \oplus \mathbf{C}) \tag{3.8a}$$

$$\mathbf{A} \otimes (\mathbf{B} \cap \mathbf{C}) \supseteq (\mathbf{A} \otimes \mathbf{B}) \cup (\mathbf{A} \otimes \mathbf{C}) \tag{3.8b}$$

On the other hand, dilation and erosion of a set by the union of a pair of sets is governed by the equality relation:

$$\mathbf{A} \oplus (\mathbf{B} \cup \mathbf{C}) = (\mathbf{A} \oplus \mathbf{B}) \cup (\mathbf{A} \oplus \mathbf{C}) \tag{3.9a}$$

$$\mathbf{A} \otimes (\mathbf{B} \cup \mathbf{C}) = (\mathbf{A} \otimes \mathbf{B}) \cup (\mathbf{A} \otimes \mathbf{C}) \tag{3.9b}$$

The following chain rules also hold for dilation and erosion because of the associativity.

$$\mathbf{A} \oplus (\mathbf{B} \oplus \mathbf{C}) = (\mathbf{A} \oplus \mathbf{B}) \oplus \mathbf{C} \tag{3.10a}$$

$$\mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C}) = (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} \tag{3.10b}$$

3.2.4 Opening and closing

Dilation and erosion are often applied to an image in concatenation to form two other morphological operations, namely *opening* and *closing*. The process of erosion followed by dilation is called opening. The opening of image **B** by structuring element **S** is denoted by $\mathbf{B} \circ \mathbf{S}$ and is defined as

$$\mathbf{B} \circ \mathbf{S} = (\mathbf{B} \otimes \mathbf{S}) \oplus \mathbf{S} \tag{3.10}$$

The opening operation intends to eliminate small and thin objects, break objects at thin points, and generally smooth the boundaries of larger objects without significantly changing their area. The reason for this is that erosion removes small features in the image that can not be recovered by the successive dilation. At the mean time, big regions shrunk by the erosion are dilated back in the dilation step without changing their area. Therefore the overall effect is the elimination of "small" features (such as noise) and the conservation of the interested objects. This is shown in Figure 3.5.

The process of dilation followed by erosion is called *closing*. The closing of image **B** by structuring element **S** is denoted by **B•S** and is defined as

$$\mathbf{B} \bullet \mathbf{S} = (\mathbf{B} \oplus \mathbf{S}) \otimes \mathbf{S} \tag{3.11}$$

The closing operation intends to fill small and thin holes in objects, connect nearby objects, and generally smooth the boundaries of objects without significantly changing their area. The reason for this is that dilation fills out small holes and small gaps between objects to form one big region, which can not be broken by the following erosion operation as shown in Figure 3.6.

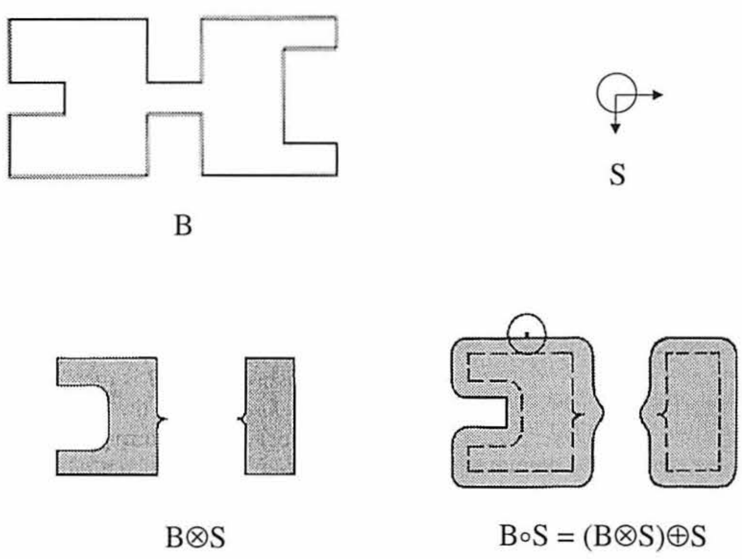


Figure 3.5 An example of opening

The opening operation satisfies the following properties.

1) $\mathbf{B} \circ \mathbf{S}$ is a subset (subimage) of **B**.

2) If **C** is a subset of **D**, then $\mathbf{C} \circ \mathbf{S}$ is a subset of $\mathbf{D} \circ \mathbf{S}$.

3) $(\mathbf{B} \circ \mathbf{S}) \circ \mathbf{S} = \mathbf{B} \circ \mathbf{S}$.

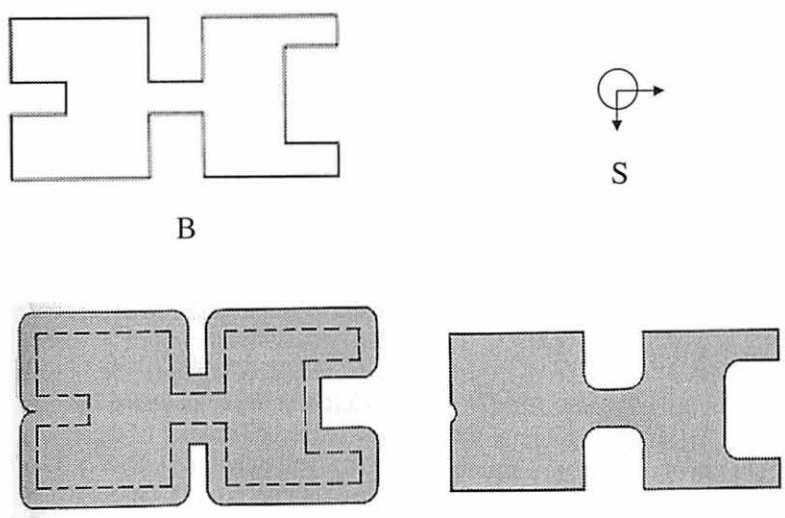


Figure 3.6 An example of closing

Similarly, the closing operation has the following properties.

1) **B** is a subset (subimage) of $\mathbf{B} \bullet \mathbf{S}$.

2) If **C** is s subset of **D**, then $\mathbf{C} \bullet \mathbf{S}$ is a subset of $\mathbf{D} \bullet \mathbf{S}$.

3) $(\mathbf{B} \bullet \mathbf{S}) \bullet \mathbf{S} = \mathbf{B} \bullet \mathbf{S}$.

The last property is called idempotence, which means that the resulting image by opening or closing is invariant to further applying of the same operation. The practical importance of idempotent transformations is that they comprise complete and closed stages of image analysis algorithms.

3.3 Gray Scale Image Morphology

Morphological concepts can be extended to gray scale image [1]. In binary morphological operation, the application of set theory is straight forward, while in the case of gray scale images, additional methods have to be provided. To generalize these concepts to a gray scale image, it is assumed that the image contains visually distinct gray scale object set against a gray background. Also, it is assumed that the objects and background are both relatively spatially smooth.

In the following discussion, the input digital image function has the form $f(x, y)$ and the structuring element function has the form $b(x, y)$.

3.3.1 Gray scale image dilation and erosion

Like the similar binary image operations, gray scale erosion and gray scale dilation operations are two most fundamental gray scale morphological operations.

Gray scale dilation of f by b, denoted $f \oplus b$, is defined as

$$(f \oplus b)(s,t) = \max\{f(s-x,t-y)+b(x,y) \mid (s-x,t-y) \in D_f ;(x,y) \in D_b\} \qquad (3.12)$$

where D_f and D_b are the domain of f and b, respectively. Now b is a function instead of a set in binary case.

Dilation is commutative. Interchange f and b in Equation 3.12 can be used to compute $b \oplus f$. The result is the same, and b now is the function translated. Because dilation is based on choosing the maximum value of $(f + b)$ in a neighborhood defined by the shape of the structuring element, there are two basic effects of dilation operation on the gray scale image: (1) if all the values of the structuring element are positive, the output image tends to be brighter than the input image; and (2) dark details are either reduced or eliminated.

Gray scale erosion, denoted $f \otimes b$, is defined as

$$(f \otimes b)(s,t) = \min\{f(s+x,t+y) - b(x,y) \mid (s+x),(t+y) \in D_f; (x,y) \in D_b\} \qquad (3.13)$$

where D_f and D_b are the domain of f and b, respectively.

Erosion is based on choosing the minimum value of $(f - b)$ in a neighborhood defined by the shape of the structuring element. There are two basic effects of erosion operation on the gray scale image: (1) if all the values of the structuring element are positive, the output image tends to be darker than the input image; and (2) bright details that are smaller in "area" than the structuring element are either reduced or eliminated.

Dilation and erosion are duals with respect to function complementation and reflection, just like in binary case. That is:

$$(f \otimes b)^c(x,y) = (f^c \oplus \hat{b})(x,y) \qquad (3.14)$$

where $f^c = -f(x,y)$ and $\hat{b} = b(-x,-y)$.

3.3.2 Gray scale image opening and closing

The closing and opening operations introduced in binary image case can be easily extended to gray scale images. Gray scale closing is achieved by first performing gray scale dilation with a gray scale structuring element, and then by performing gray scale erosion with the same structuring element. The closing of image f by image (structuring element) b, denoted $f \bullet b$, is

$$f \bullet b = (f \oplus b) \otimes b \tag{3.15}$$

Similarly, gray scale opening is accomplished by gray scale erosion followed by gray scale dilation. The opening of image f by image (structuring element) b, denoted $f \circ b$, is

$$f \circ b = (f \otimes b) \oplus b \tag{3.16}$$

The opening and closing for gray scale images are duals with respect to complementation and reflection. That is

$$(f \bullet b)^c = f^c \circ \hat{b} \tag{3.17}$$

where $f^c = -f(x,y)$, $\hat{b} = b(-x,-y)$.

In practical application, opening operation is usually applied to remove small (with respect to the size of the structuring element) light details, while leaving the overall gray levels and larger brighter features relatively undisturbed. The closing operation is generally used to remove dark details from an image, while leaving bright features relatively undisturbed.

CHAPTER 4. GENETIC ALGORITHM

4.1 Introduction

The Darwinian theory of natural evolution, especially the "survival of the fittest" principle, and the mechanisms of natural genetics are the basis for the Genetic Algorithm optimization techniques. The Genetic Algorithm imitates natural selection to identify the maximum/minimum of some objective function in a search space. Genetic algorithms are theoretically and empirically proven to provide a robust search in the complex solution space. They are computationally simple yet powerful in their search for improvement. Their main attractive characteristic is their ability to deal efficiently with hard combination search problems, where the parallel exploration of the search space eliminates to a large extent the possibility of getting stuck in local extrema.

The Genetic Algorithm is based on the mechanism exhibited by nature incorporating the robustness, the efficiency, and the flexibility of biological systems. Genetic algorithms have been used to solve difficult problems with objective functions that do not possess "nice" properties such as continuity, differentiability, satisfaction of the Lipschitz Condition, etc. [12], [13], [14], [15]. These algorithms maintain and manipulate a family, or population, of solutions and implement a "survival of the fittest" strategy in their search for better solutions. This approach is based on the fact that individuals tend to pass on their traits to their offspring. In general, the fittest individuals of any population tend to reproduce and survive to the next generation, thus improving successive generation under particular ecological conditions.

Having been established as a valid approach to complex problems requiring efficient and effective search, genetic algorithms now have more widespread applications in business, scientific and engineering fields. This thesis presents the application of GA to the problem of Image Segmentation.

4.2 Genetic Algorithm

Genetic algorithms search the solution space of a function through the use of simulated evolution, i.e., the survival of the fittest strategy. Genetic algorithms have been shown to solve linear and nonlinear problems by exploring all regions of the state space and exponentially exploiting promising areas through *mutation*, *crossover*, and *selection* operations applied to individuals in the *population*, which is individual solutions (analogous to chromosomes) of the state space [15]. These operators, which rely on probability rules, are applied to the population, and successive generations are produced. In general, the starting search for an optimal solution begins with a randomly generated population of chromosomes. Each generation will have a new set of chromosomes obtained from the application of the operators. A *fitness*, or objective function, is defined according to the problem. The parent selection process ensures that the fittest members of the population have highest probability of becoming parents, in the hope that their offspring will combine desirable features, and have superior fitness, to both. The algorithm terminates either when a set of generation number is reached, or the fitness has reached a "satisfactory" level. The use of a genetic algorithm requires the determination of six fundamental issues: (1) chromosome representation, (2) selection function, (3) creation of the initial population, (4) genetic operators making up the reproduction function, (5) fitness function, and (6) termination criteria. The Genetic Algorithm consists of the following steps:

1. Generate the initial population.
2. Evaluate the fitness of the each individual according to a fitness function.
3. Select the fittest individual for mating.
4. Apply reproductive operators (e.g. crossover, mutation) to create offspring.
5. Evaluate the fitness of the offspring and select the fit individuals from the current generation and the offspring. They form the population of the next generation.
6. Stop if stopping criterion is met, else go to step 3.

A genetic algorithm is summarized in Figure 4.1, and each of the major components is discussed in detail below.

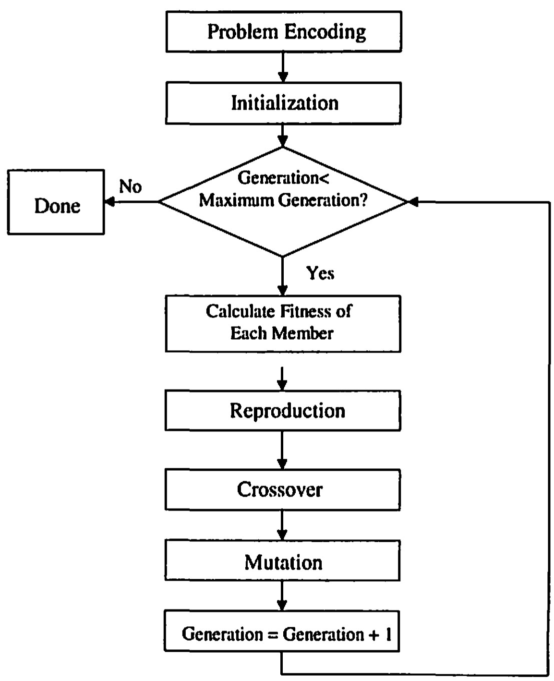


Figure 4.1 A simple Genetic Algorithm

4.2.1 Solution representation (problem encoding)

For any GA, a chromosome representation is needed to describe each individual in the population of interest. The representation scheme determines how the problem is structured in the GA and also determines the genetic operators that are used. Each individual or chromosome is made up of a sequence of genes from a certain alphabet. An alphabet could consist of binary digits (0 and 1), floating point numbers, integers, symbols (i.e., *A, B, C, D*), matrices, etc. In Holland's original design, the alphabet was limited to binary digits. Since then, problem representation has been the subject of much investigation. It has been shown that more natural representations are more efficient and produce better solutions [15]. One useful representation of an individual or chromosome for function optimization involves genes or variables from an alphabet of floating point numbers with values within the variables upper and lower bounds. Michalewicz [15] has done extensive experimentation comparing real-valued and binary GAs and has shown that the real-valued GA is an order of magnitude more efficient in terms of CPU time. He also shows that a real-valued representation moves the problem closer to the problem representation that offers higher precision with more consistent results across replications [15]. The technique for the solution representation may vary from problem to problem. In most of the work done so far, representation is carried out using bit strings.

4.2.2 Selection function

The selection of individuals to produce successive generations plays an extremely important role in the genetic algorithm. A probabilistic selection is performed based on the individual's fitness such that the better individuals have an increased chance of being selected. An individual in the population can be selected more than once with all individuals in the population having a chance of being selected to reproduce into the next generation. There are several schemes for the selection process: roulette wheel selection and its extensions, scaling techniques, tournament, elitist models, and ranking methods [13], [15].

A common selection approach assigns a probability of selection, P_j, to each individual, j based on its fitness value. A series of N random numbers is generated and compared against the cumulative probability, $C_i = \sum_{j=1}^{i} P_j$, of the population. The appropriate individual, i, is selected and copied into the new population if $C_{i-1} < U(0,1) \leq C_i$, where $U(0,1)$ is uniformly distributed random number between 0 and 1. Various methods exist to assign probabilities to individuals such as roulette wheel, linear ranking and geometric ranking [13], [15].

Roulette wheel, developed by Holland [14], was the first selection method. The probability, P_i, for each individual is defined by:

$$P\,[\text{individual } i \text{ is chosen}] = \frac{F_i}{\sum_{j=1}^{PopSize} F_j}, \tag{4.1}$$

where F_i equals the fitness of individual i. The use of roulette wheel selection limits the genetic algorithm to maximization since the evaluation function must map the solutions to a fully ordered set of values on $\Re^+$. Extensions, such as windowing and scaling, have been proposed to allow for minimization and negativity.

Ranking methods only require the evaluation function to map the solutions to a partially ordered set, thus allowing for minimization and negativity. Ranking methods assign P_i based on the rank of solution i when all solutions are sorted. Normalized geometric ranking, [16], defines P_i for each individual by:

$$P \text{ [Selecting the } i\text{th individual]} = q'(1-q)^{r-1} \tag{4.2}$$

where:

q = the probability of selecting the best individual

r = the rank of the individual, where 1 is the best

P = the population size

$$q' = \frac{q}{1-(1-q)^P}$$

Tournament selection, like ranking methods, only requires the evaluation function to map solutions to a partially ordered set; however, it does not assign probabilities. Tournament selection works by selecting j individuals randomly, with replacement, from the population, and inserts the best of the j into the new population. This procedure is repeated until N individuals have been selected.

4.2.3 Initialization

The GA must be provided an initial population as indicated in step 2 of Figure 4.1. The initial population usually is randomly generated according to a uniform random distribution over the set of possible solutions. However, since GAs can iteratively improve existing solutions (i.e., solutions from other heuristics and/or current practices), the beginning population can be seeded with potentially good solutions, with the remainder of the population being randomly generated solutions. In other words, the solution set may be altered if background information indicates that the search should be biased toward certain areas of the solution space.

4.2.4 Genetic operators

Genetic operators provide the basic search mechanism of the GA. The operators are used to create new solutions based on existing solutions in the population. There are three basic types of operators: *reproduction, crossover* and *mutation*. Reproduction [13] is a process in which individual strings are copied according to their fitness function values. Crossover takes two individuals and produces two new individuals while mutation alters one individual to produce a single new solution. The application of these three basic types of operators and their derivatives depends on the chromosome representation used.

Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two m-dimensional row vectors denoting individuals (parents) from the population. For $\boldsymbol{X}$ and $\boldsymbol{Y}$ binary, the following operators are defined: binary mutation and simple crossover.

Binary mutation flips each bit in every individual in the population with probability p_m according to Equation 4.3.

$$x_i' = \begin{cases} 1 - x_i, & \text{if} \quad U(0,1) < p_m \\ x_i, & \text{otherwise} \end{cases} \tag{4.3}$$

Simple crossover generates a random number *r* from a uniform distribution from 1 to *m* and creates two new individuals (***X'*** and ***Y'***) according to Equations 4.4 and 4.5.

$$x_i' = \begin{cases} x_i, & if \quad i < r \\ y_i, & otherwise \end{cases} \tag{4.4}$$

$$y_i' = \begin{cases} y_i, & if \quad i < r \\ x_i, & otherwise \end{cases} \tag{4.5}$$

Michalewicz [15] developed operators for real-valued representations, i.e., an alphabet of floats. For real ***X*** and ***Y***, the following operators are defined: *uniform mutation, non-uniform mutation, multi-non-uniform mutation, boundary mutation, simple crossover, arithmetic crossover,* and *heuristic crossover*. Let a_i and b_i be the lower and upper bound, respectively, for each variable *i*.

Uniform mutation randomly selects one variable, *j*, and sets it equal to a uniform random number $U(a_i, b_i)$:

$$x_i' = \begin{cases} U(a_i, b_i), & if \quad i = j \\ x_i, & otherwise \end{cases} \tag{4.6}$$

Boundary mutation randomly selects one variable, *j*, and sets it equal to either its lower or upper bound, where $r = U(0, 1)$:

$$x_i' = \begin{cases} a_i, & if \quad i = j, \quad r < 0.5 \\ b_i, & if \quad i = j, \quad r \geq 0.5 \\ x_i, & otherwise \end{cases} \tag{4.7}$$

Non-uniform mutation randomly selects one variable, j, and sets it equal to a non-uniform random number:

$$x_i' = \begin{cases} x_i + (b_i - x_i) f(G) & if \quad r_1 < 0.5 \\ x_i - (x_i + a_i) f(G) & if \quad r_1 \geq 0.5 \\ x_i, & otherwise \end{cases} \tag{4.8}$$

where

$$f(G) = (r_2(1 - \frac{G}{G_{\max}}))^b, \tag{4.9}$$

r_1, r_2 = a uniform random number between (0,1),

G = the current generation,

G_{max} = the maximum number of generations,

b = a shape parameter.

The multi-non-uniform mutation operator applies the non-uniform operator to all of the variables in the parent $\boldsymbol{X}$.

Real-valued simple crossover is identical to the binary version presented above in Equations 4.4 and 4.5. Arithmetic crossover produces two complimentary linear combinations of the parents, where $r = U\,(0,\ 1)$:

$$X' = rX + (1 - r)\,Y \tag{4.10}$$

$$Y' = (1 - r)\,X + rY \tag{4.11}$$

Heuristic crossover produces a linear extrapolation of the two individuals. This is the only operator that utilizes fitness information. A new individual, X', is created using Equation 4.12, where $r = U\,(0,\ 1)$ and X is better than Y in terms of fitness. If X' is *infeasible*, i.e., feasibility equals 0 as given by Equation 4.14, then a new random number r is generated and a new solution is created using Equation 4.12; otherwise the crossover operation stops. To ensure halting, after t failures, let the children equal the parents and stop.

$$X' = X + \mathrm{r}\,(X - Y) \tag{4.12}$$

$$Y' = X \tag{4.13}$$

$$feasibility = \begin{cases} 1, & if \quad x_i' \ge a_i, x_i' \le b_i \quad \forall i \\ 0, & otherwise \end{cases} \tag{4.14}$$

The mechanics of reproduction, crossover and mutation are simple, involving random number generation, string copies, and some partial string exchanges. These processes in a GA search for best solutions.

4.2.5 Fitness functions

The fitness function is the link between the GA and problem to be solved. A fitness function returns a single numerical value that represents how fit the member is compared to the rest of the members in the population. Fitness functions of many forms can be used in a GA, subject to the minimal requirement that the function can map the population onto a partially ordered set.

4.2.6 Termination (stopping criterion)

The GA moves from generation to generation selecting and reproducing parents until a stopping criterion is met. The most frequently used stopping criterion is a specified maximum number of generations. Another termination strategy involves population convergence criteria. In general, GA will force much of the entire population to converge to a single solution. When the sum of the deviations among individuals becomes smaller than some specified threshold, the algorithm can be terminated. The algorithm can also be terminated due to a lack of improvement in the best solution over a specified number of generations. Alternatively, a target value for the evaluation measure can be established based on some arbitrarily "acceptable" threshold. Several strategies can be used in conjunction with each other.

The search involved in Genetic Algorithm is from a population, not from a single point. Although it is randomized, GA is not a simple random walk. GA does not require any auxiliary information. It only requires a function value associated with each individual. This characteristic makes GA a more canonical method than many other search schemes. The mechanics of GA involve nothing but copying strings and swapping partial strings. Simplicity of operations and effectiveness in finding the solution are two major attractions of GA.

4.2.7 Elitism and diversity

Elitism intends to maintain the diversity of the population. It keeps certain amount of the top fittest individuals. After certain generations, these individuals may not be the good one so they are replaced by another set of fittest individuals. These individuals are kept in several generations again until they are out of date. Elitism is not necessary the way to bring the diversity into the population, the selection function will be designed to bring the diversity into the population also. At the same time, reproduction process will generate some diverse individuals for the selection process.

CHAPTER 5. IMAGE SEGMENTATION BASED ON GENETIC ALGORITHM AND MATHEMATICAL MORPHOLOGY

Image segmentation is the fundamental process in image analysis. Segmentation algorithms basically identify homogeneous image regions, each corresponding to objects or to background. There are many segmentation methods as addressed in Chapter 2. In this chapter, we present a new Image Segmentation method based on Genetic Algorithm and Mathematical Morphology.

5.1 Problem Statement

In this thesis, the problem of detecting homogeneous regions in an image is addressed. We are concerned with identifying objects in a noisy image. As in a digitized image, the noisy image is represented by a matrix **X** with components, x_{ij}, whose value represents the intensity of the pixel (i, j). The noisy image is produced by superimposing a noise component on the original image as described by the following equation:

$$x_{ij} = f_{ij} + n_{ij} \tag{5.1}$$

Here, f_{ij} is the original image intensity and n_{ij} is the noise that is normally distributed with zero mean and variance σ^2. The original image in this study is composed of a single object intensity R_o embedded in a background of intensity R_b. The segmentation algorithm is performed on 16×16 subimages, and the resulting subimages are then combined to obtain the entire segmented image.

The 2-D 16×16 subimage is represented as a vector in 1-D as $\overline{x} = [x_{0,} x_{1,} ......, x_{255}]$, $0 \le x_i \le 255$, where x_i represents the intensity of the pixel in the ith position. Figure 5.1 shows the matrix of the 2D image and Figure 5.2 shows the 16×16 noisy image with embedded L shaped object.

$$
\begin{array}{cccccc}
x_0 & x_1 & x_2 & x_3 & \ldots\ldots & x_{15} \\
x_{16} & x_{17} & x_{18} & x_{19} & \ldots\ldots & x_{31} \\
x_{32} & x_{33} & x_{34} & x_{35} & \ldots\ldots & x_{47} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_{240} & x_{241} & x_{242} & x_{245} & \ldots\ldots & x_{255}
\end{array}
$$

Figure 5.1 The subimage matrix

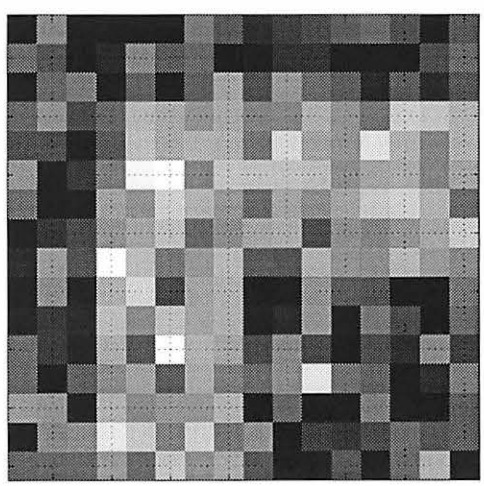


Figure 5.2 2D 16×16 noisy image

5.2 Approach

The noisy image to be segmented can be any size. An appropriate subimage size is chosen based on the size of the original image. This limits the length of the chromosome to a reasonable size. After the image is divided into desired subimages, the algorithm for image segmentation is applied to each subimage. The algorithm used to segment the noise image includes (i) generating initial population, (ii) evaluating every individual in the population, (iii) generating offspring by applying reproductive operators, (vi) moving to the next generation by allowing only the fittest individuals to survive. This process is iterated until the stopping criterion is met. Of all the populations in the final generation, the fittest individual is the segmented subimage. The resulting segmented subimages are then combined to form the entire segmented image. The steps mentioned above in the algorithm are explained in details next.

5.2.1 Initial population

A set of individuals has to be produced to create an initial population in GA. This set is the initial population of GA. Each individual in this set is a string with element value of R_o (object-intensity) or R_b (background-intensity). The initial population can be represented by the set,

$$\{Y_i^k\}, \quad i = 0, 1, \ldots, 255; \; k = 1, 2, \ldots, N$$

where N is the population size.

To generate candidates in the initial population, random method was used. That is, each element in the vector is randomly chosen to be R_o or R_b.

$$Y_i^k = R_0 \text{ or } R_b \text{ (randomly)}, \quad i = 0, 1, ..., 255; \; k = 1, 2, ..., N$$

Each vector represents an individual in the initial population. These vectors actually are string representation of the subimage. Figure 5.3 shows a typical randomly generated 16×16 subimage which is a candidate of the initial population.

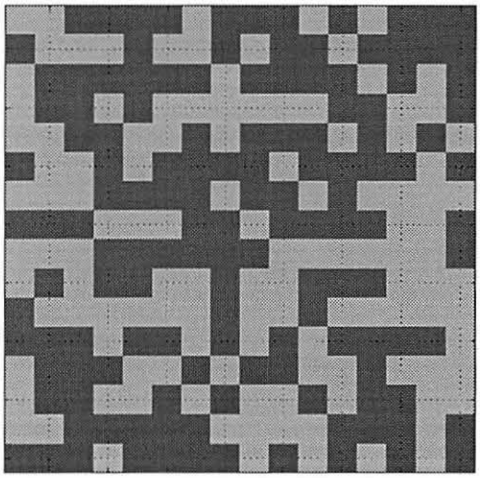


Figure 5.3 A candidate of initial population generated by random method

This method does not depend on the knowledge of noise variance. Since this method is totally random in generating the candidates, the initial population size is kept high so that a large enough solution space is used to achieve better results. Other methods generate the candidates for the initial population based on the knowledge of the noise variance. In this way the population size can be decreased compared to that used in the random method. But this requires the noise variance to be given by the user before generating an initial population.

5.2.2 Fitness evaluation

A fitness function must be defined in order to evaluate the fitness of the individuals in the population. The fitness function is very important in GA. It helps to decide whether the individuals are capable of producing offsprings, whether the offsprings are fit enough to survive, and whether the individuals in the current generation are capable of existing in the next generation.

If the original image is $\bar{x}$ and the initial population is

$$\{Y_i^k\},\ \ i = 0, 1, ..., 255;\ k = 1, 2, ..., N$$

The fitness of an individual $f(Y^k)$ of the population is calculated as:

$$f(Y^k) = E(Y^k) + \alpha \times T(Y^k) \qquad k = 1, 2, ..., N \tag{5.2}$$

where
$$E(Y^k) = \frac{1}{\sum_{i=0}^{255} |Y_i^k - x_i|} \tag{5.3}$$

$$T(Y^k) = \frac{1}{\sum_{i=0}^{255} |Y_i^k - w_i|} \qquad w_i \in neighboring\ pixel\ of\ Y_i^k \tag{5.4}$$

$E(Y^k)$ is the measure of similarity between the individual Y^k and the original noisy subimage. The goal here is to minimize the total difference between the noisy image of varying gray level intensity and the binary candidate subimage. The candidate subimage which has the least hamming distance to the noisy image has the highest $E(Y^k)$. The second term in the fitness, $T(Y^k)$ is the reciprocal of the transition count in the horizontal and vertical direction. This term is introduced in order to let the image with homogeneous regions have a better fitness than those with discontinuities do. In general, the fitter individuals will have a high $T(Y^k)$ since the sum of the transition count will be low. α is the weighting factor that normalizes the two quantities $E(Y^k)$ and $T(Y^k)$.

To calculate the transition count, the 4 neighbors of every pixel are observed. If the pixel under consideration has a value R_b and three out of the four neighbors have value R_o then the transition count of that pixel is $3\times|R_b - R_o|$. The corner pixels have only two 4-neighbors and the pixels on the edge of the image have only three 4-neighbors. Figure 5.4 is the noisy image with SNR = 2. Two candidates along with the similarity measure value $E(Y^k)$ and transition count measure $T(Y^k)$ are shown in Figures 5.5 and 5.6. It can be seen from these figures that the candidate in Figure 5.6 is homogeneous and has a higher $T(Y^k)$ than the one in Figure 5.5.

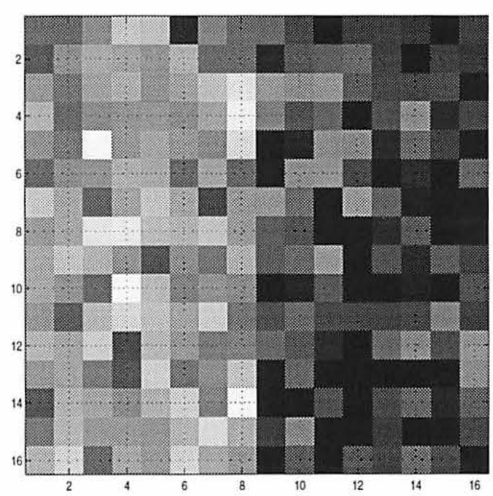


Figure 5.4 16×16 noisy image with SNR = 2

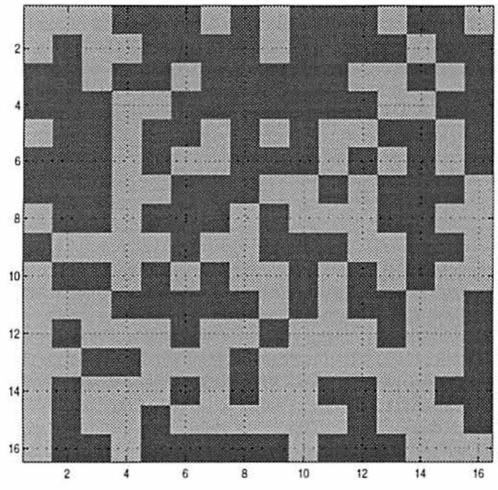


Figure 5.5 Similarity: $E(Y^k) = 5.6979\times10^{-5}$

Transition count: $T(Y^k) = 1.8904\times10^{-5}$

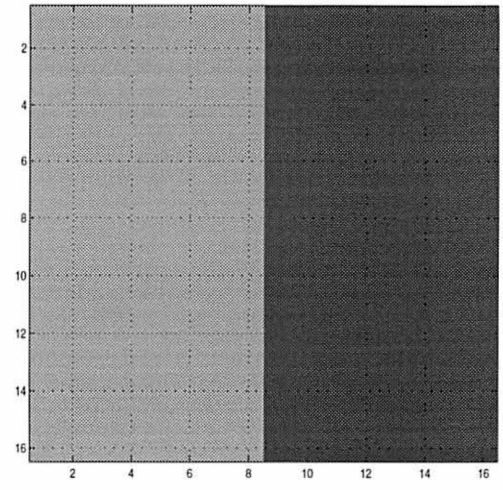


Figure 5.6 Similarity: $E(Y^k) = 9.9140\times10^{-5}$

Transition count: $T(Y^k) = 1.0417\times10^{-4}$

5.2.3 Selection of fittest individuals

Once the evaluation of the fitness of the individuals has been done, the fittest individuals must be selected so that they can be mated to produce offspring in the next generation. A very simple way to accomplish the selection used in this thesis is by defining a threshold Φ.

$$\Phi = \frac{Max(f) + Min(f)}{2} \tag{5.5}$$

Max(f) and *Min(f)* are the maximum and minimum values of the fitness f, respectively. The individuals with fitness greater than the threshold are selected to produce offspring. Sometimes when Φ is small, in order that the search space does not get limited, a minimum number of offspring, η, is defined. If the number of generated offsprings is less than η then the threshold Φ is decreased so that more individuals are selected for the reproduction and the number of offspring generated by the selected individuals is more than the minimum required.

5.2.4 Reproduction

This step is very important in the genetic algorithm because it is responsible for the evolution. The quality of final results depends on the GA operators in this step. The basic operators of reproduction, namely crossover and mutation are discussed in chapter 4. However, the implementation of these operators depend on a particular problem. In this thesis, morphological operations are used as reproduction operators on selected individuals before the two-point crossover and mutation operators are applied. Figure 5.7 shows this procedure.

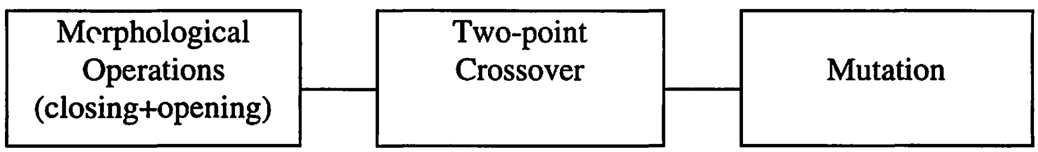


Figure 5.7 Reproduction Procedure

5.2.4.1 *Morphological operations*

Morphological operation used in the first step is “closing” followed by opening. The structuring elements of size 3×3, 5×5, and 7×7 were tried. Depending on the particular problem, different structuring element size are chosen. The element value was also varied by assigning 50, 100, and 200 to the entire element. These two operations tend to clear small holes or small blocks in the noisy image and render the object and background regions more homogeneous. Individuals pre-conditioned using morphological operation are then subjected to crossover and mutation operators to generate offspring.

5.2.4.2 Two-point crossover

From earlier discussion in section 5.2.3, we know that not all the individuals in the population will produce offspring. Based on Equation 5.5, only the individuals with fitness better than the threshold Φ are considered fit to be parents. Two parents are randomly selected among these qualified individuals to perform a two-point crossover and produce two offspring. A minimum number of parents, P, is defined maintain a minimum number of qualified individuals. If the number of qualified individuals are less than P then the threshold Φ is decreased in order to let more individuals to be qualified. The number of offspring, M, is also defined in order to maintain the population size in each generation. Assume that the total number of qualified individuals in the generation with population size N is K. Two of them are randomly selected for the crossover until the number of offsprings produced is greater than M. Let S denote the set of K individuals selected from the current generation.

$$S = \{Y_j^i\} \quad 1 \le i \le K, \; 0 \le j \le 255 \tag{5.6}$$

Every two individuals in the set S generate two new offspring as follows. For each randomly chosen individual Y^i, i=1, 2,..., K, two crossover points p_1 and p_2 ($0 < p_1 < p_2 < 255$) are randomly generated. If Y^j, j=1, 2,..., K, is another randomly chosen individual in $\{S\}$, then the p_2-p_1+1 elements of Y^j, $\{Y_{p_1}^j, Y_{p_1+1}^j, \ldots, Y_{p_2}^j\}$, are exchanged with the remaining 256-(p_2-p_1+1) elements from in Y^i. The two new offspring created from Y^i, Y^j in $\{S\}$ can be represented as,

$$\{Y_0^i, Y_1^i, \ldots, Y_{p_1-1}^i, Y_{p_1}^j, Y_{p_1+1}^j, \ldots, Y_{p_2}^j, Y_{p_2+1}^i, Y_{p_2+2}^i, \ldots, Y_{255}^i\}$$
$$i, j = 1, 2, \ldots, K, \quad Y^i, Y^j \in \{S\}$$

and

$$\{Y_0^j, Y_1^j, \ldots, Y_{p_1-1}^j, Y_{p_1}^i, Y_{p_1+1}^i, \ldots, Y_{p_2}^i, Y_{p_2+1}^j, Y_{p_2+2}^j, \ldots, Y_{255}^j\}$$
$$i, j = 1, 2, \ldots, K, \quad Y^i, Y^j \in \{S\}$$

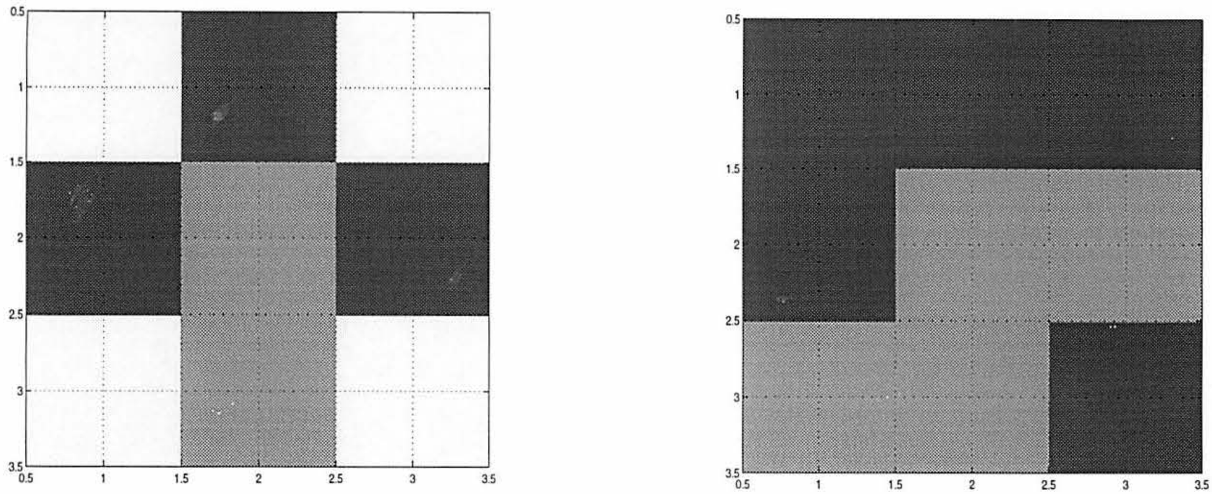
Among all this individuals, only fittest N (population number) of them is selected to survive.

5.2.4.3 *Mutation*

The reason that the mutation operator is introduced is because that, in nature, a living being undergoes some changes in its characteristics due to the influence of the surroundings. The mutation operations is intended to adapt the candidate better to its surroundings. The mutation operator is applied to those existing individuals in our study in a similar way in order to develop traits that make them fitter for survival. The mutation operator does not produce new offspring but it improves the fitness of the individual. The procedure to apply the mutation operator to an individual is as follows:

Each bit (pixel value) in an individual is modified depending on the value possessed by the majority of its neighbors. For each bit x_i in the individual Y_i^k the neighborhood C is defined and considered. It is similar to the 8-pixel neighborhood or 4-pixel neighborhood. If N_b is the number of background pixels and N_o is the number of object pixels in C then

$$Y_i^k = \begin{cases} R_b & \text{if } N_b > N_o \\ R_o & \text{if } N_b < N_o \end{cases} \quad i = 0,1,\ldots,255, \quad k = 1,2,\ldots,N \tag{5.7}$$



(a) 4-pixel neighborhood (b) 8-pixel neighborhood

Figure 5.8 Different neighborhoods

Figure 5.8(a) shows the typical 4-pixel neighborhood and figure 5.8(b) shows the 8-pixel neighborhood. In case (a), the center pixel is the pixel under consideration with $N_b = 3$ and $N_o = 1$. Since $N_b > N_o$, the center pixel is changed from R_o to R_b. In case (b), $N_b = 5$ which is greater than $N_o = 3$, hence the center pixel is also changed from R_o to R_b.

5.2.5 Next generation

After the reproduction operators have been applied to the selected individuals, there are total $N + K(K\text{-}1)$ individuals in the current population. The fitness of each individual is then evaluated using the fitness function in Equation (5.2). The fittest N individuals are selected to form the next generation. Generally, if there are more than one individual with the same fitness value and the same element value, only one of them is randomly selected to prevent GA getting stuck in the local extrema.

5.2.6 Stopping criterion

The genetic algorithm evolves from one generation to the next generation and eventually, all individuals in the population will have the same (or very close) fitness after a certain number of generations, indicating that convergence is reached. Typically the algorithm will stop at this point because further processing will not improve the quality of the individuals. In this study, the algorithm is stopped either after a fixed number of generations or the maximum and minimum fitness of individuals are in a desired range.

5.3 Parameters that Affect the Performance of the Algorithm

The various parameters of the algorithm have to be chosen carefully in order to solve the problem effectively. The next section provides a brief discussion of parameters that affect the performance of the algorithm.

5.3.1 Population size

Population size is the number of individuals at a given time in the population. In this study, a fixed population size is used through all the generations. The size of the population is directly proportional to the run time of the algorithm. A small population size speeds up the search. But on the other hand, the smaller the population size, the more generations the algorithm needs to find the best results. Hence there is a trade off between the population size and number of generations.

5.3.2 Fitness functions

For a given problem, the fitness function can be chosen in a number of ways which reflect the flexibility of the genetic algorithm. The choice of a proper fitness function is very important because the performance of the algorithm depends critically on the fitness function chosen. If the fitness function is not appropriate, it is possible the average fitness increases from one generation to the next generation, but the results obtained are far away from optimal. Hence care must be taken while choosing the fitness function since the wrong choice of fitness function may lead to total failure of the algorithm.

5.3.3 Structuring element size

The size of the structuring element used in the morphological operation also depends on the nature of the image. The ideal size of the structuring element is chosen such that the morphological operation eliminates the small holes and speckles or noise pixels in the background. Different size of the structuring element has been studied and the results are presented in the next chapter. So far, there are no theoretical considerations on the optimality of the size of structuring element.

5.3.4 Crossover techniques

There are various methods for crossover. Normally, individuals in the mating pool produce an offspring with every other individual. A probability concept can also be introduced so that each individual mates with the remaining individuals with a certain probability. Based on specific properties of digital images, other crossover techniques have also been studied.

These complicated techniques may affect the simplicity of the algorithm and there is therefore a tradeoff between complexity of computation and accuracy of segmentation results.

5.3.5 Mutation techniques

The mutation technique used in this study is based on the neighborhood of each bit (pixel value) in the candidate, where each bit is trapped if the majority of the neighbors is different from the pixel under consideration. Another way to do the mutation is to define a mutation rate and randomly flip bits. Using a 8-pixel neighborhood bits can be flipped with some induced probability instead of just count the object or background intensity pixel number. This can modify the mutation operator used in this study.

5.3.6 Selection function

A selection function is necessary to decide the pool of individuals in the population that can be used for producing the offspring for the next generation. One way is to simply decide a fixed number of individuals to be selected based on their fitness evaluations. Another way is to define a threshold as discussed in section 5.3.2. The latter method brings more flexibility because we can select the best-fit individuals with no limitation on the quantity. If the number of the qualified individuals is less than the minimum number requirement, the algorithm will decrease the threshold value to let more individuals to be selected. By doing this, every time a different number of individuals is selected to produce offspring and the diversity of the population has been maintained.

5.3.7 Stopping criterion

In this study, the stopping criteria used are (1) the average fitness does not change from one generation to the next, and (2) a maximum number of iterations are performed. There are some other stopping criteria which are all related to the convergence of the average fitness. Convergence is also defined in different ways. Sometimes the average fitness oscillates within a certain range. In this case, a range is defined so that if the difference between the maximum and the minimum of the average fitness during several generations is in this range, the algorithm is converged.

In the next chapter, some simple simulated images are used to investigate the effects of above parameters on the performance of the algorithm on images of varying signal-to-noise ratio (*SNR*) and different objects.

CHAPTER 6. RESULTS AND DISCUSSION

The application of genetic algorithm to image segmentation in combination with morphological operations as the reproduction operators is studied. The role of different parameters of the algorithm is also studied to get better understanding of this algorithm. These results are presented and discussed in the following sections.

6.1 Population Size

The initial population was generated randomly. The population size was chosen to be 50, 100, 150 and 200. The maximum number of generation was selected to be 100. The study was conducted on images of size of 16×16 with $R_o = 150$ and $R_b = 50$. The noisy image was produced by superimposing a random gaussian noise with zero mean and with certain variance on the clear images. The noise variance σ^2 was 2500 and 4000, and the corresponding signal to noise ratio (*SNR*) equals to 2 and 1.58, respectively. The result for population size 50 is shown in Figure 6.1: (a) the original image, (b) the noisy image with *SNR* = 2, (c) one candidate of the initial population and finally (d) the segmented image. Figure 6.2, shows the comparison of segmented images using the same parameters for the population size 50, 100, 150 and 200, respectively. For all these different initial population sizes, the candidates are selected from the same population, i.e., the initial population was generated once and then selected to form different initial population size.

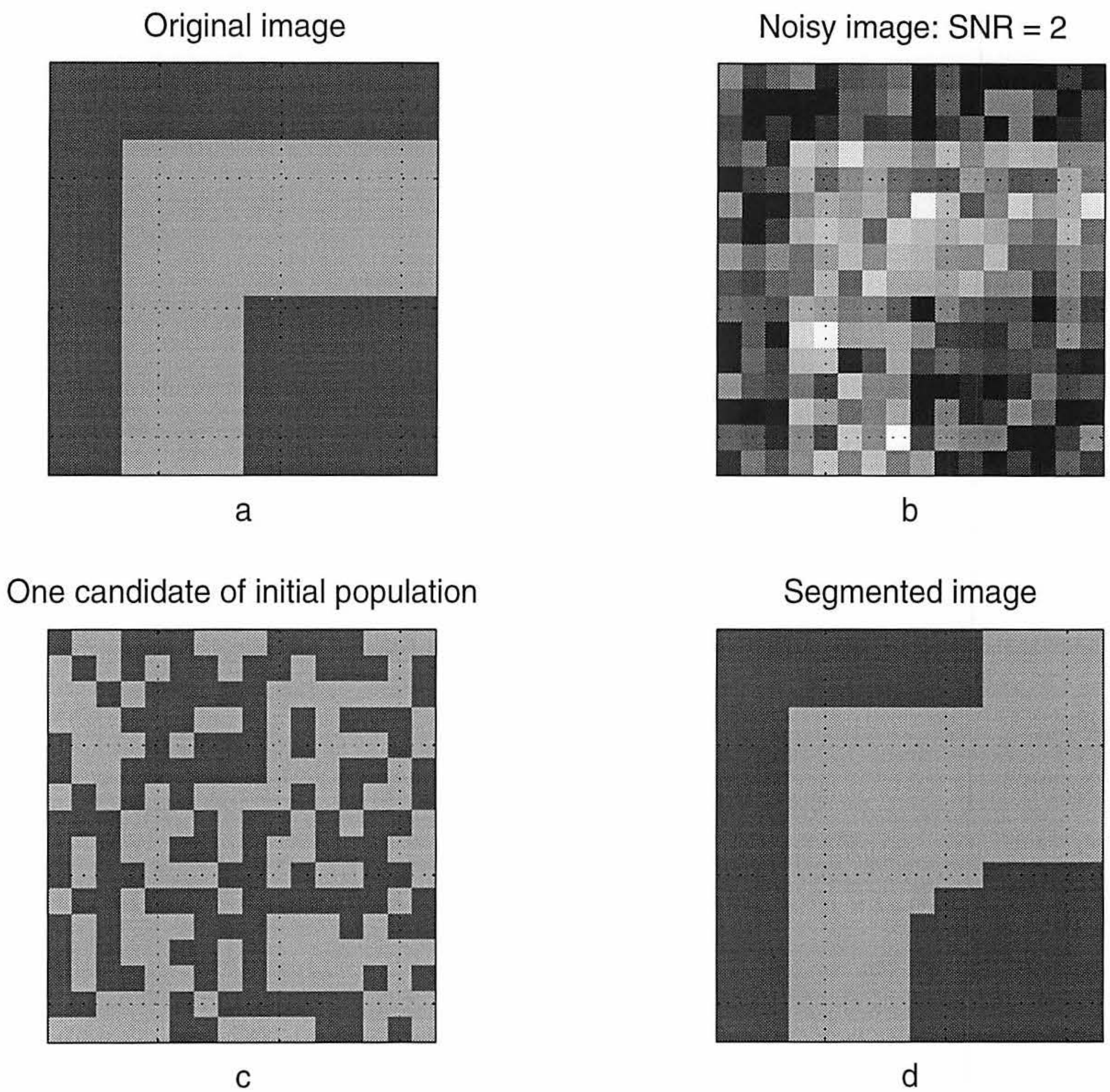


Figure 6.1 Results for 16×16 image with L Shape Object

(SNR = 2, population size = 50)

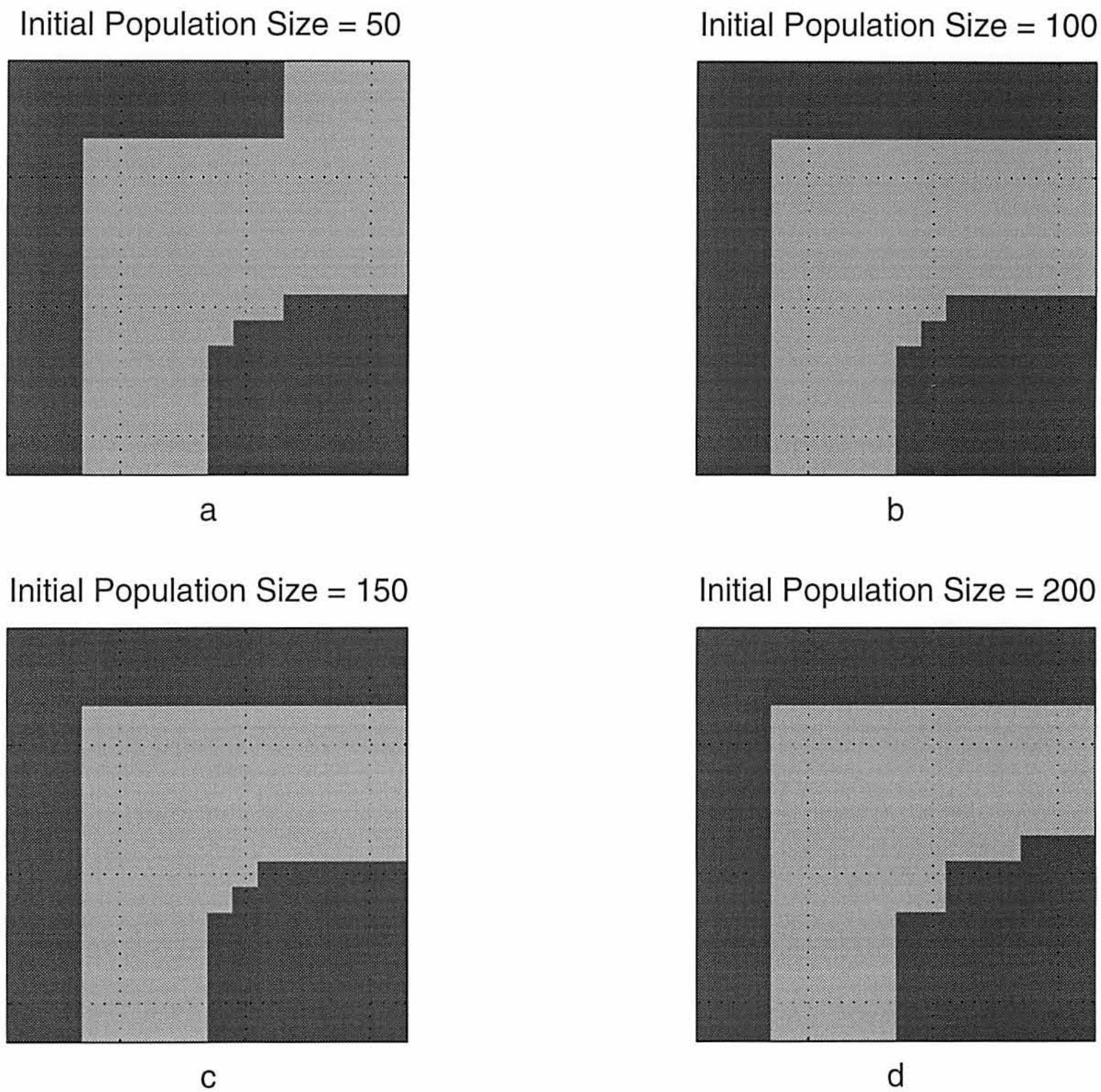


Figure 6.2 Comparison of segmented image using different sizes of initial population ($SNR = 2$)

The same study was also conducted on images of *SNR* = 1.58. The results presented in Figure 6.3 show (a) the original image, (b) the noisy image with *SNR* = 1.58, (c) one candidate of the initial population and finally (d) the segmented image. Figure 6.4 shows the comparison of segmented images for the population size 50, 100, 150 and 200. The plots of the reciprocal of the average fitness with respect to the number of generations are also presented in Figure 6.5 and Figure 6.6 for *SNR* = 2 and *SN*R = 1.58, respectively.

From these results, we could draw the conclusion that the initial population size should be around 100. Even though the size of 150 and 200 sometimes may give better results, it is still computationally expensive. Although this study is only concentrated on the L shape object image, some other images with different objects are studied later. Those results also confirm the conclusion that for 16×16 images, population size less than 50 will not optimal.

Considering the reciprocal of the average fitness, it can be seen from Figure 6.5 and 6.6 that with increasing initial population size, the convergence of the algorithm will get faster but it is not so obvious when the population size is larger than 100. Also, the reciprocal of average fitness shows that the quality of the segmented image does not improve with initial population size larger than 100 while the computation time gets longer and longer. On the other hand, the algorithm converges to a certain point after 20 generations with the population size between 100 and 200.

Studies in this section focus on the image with *SNR* = 2 and *SNR* = 1.58. Later in this chapter, the application of this algorithm on low contrast images with *SNR* = 1 will be discussed.

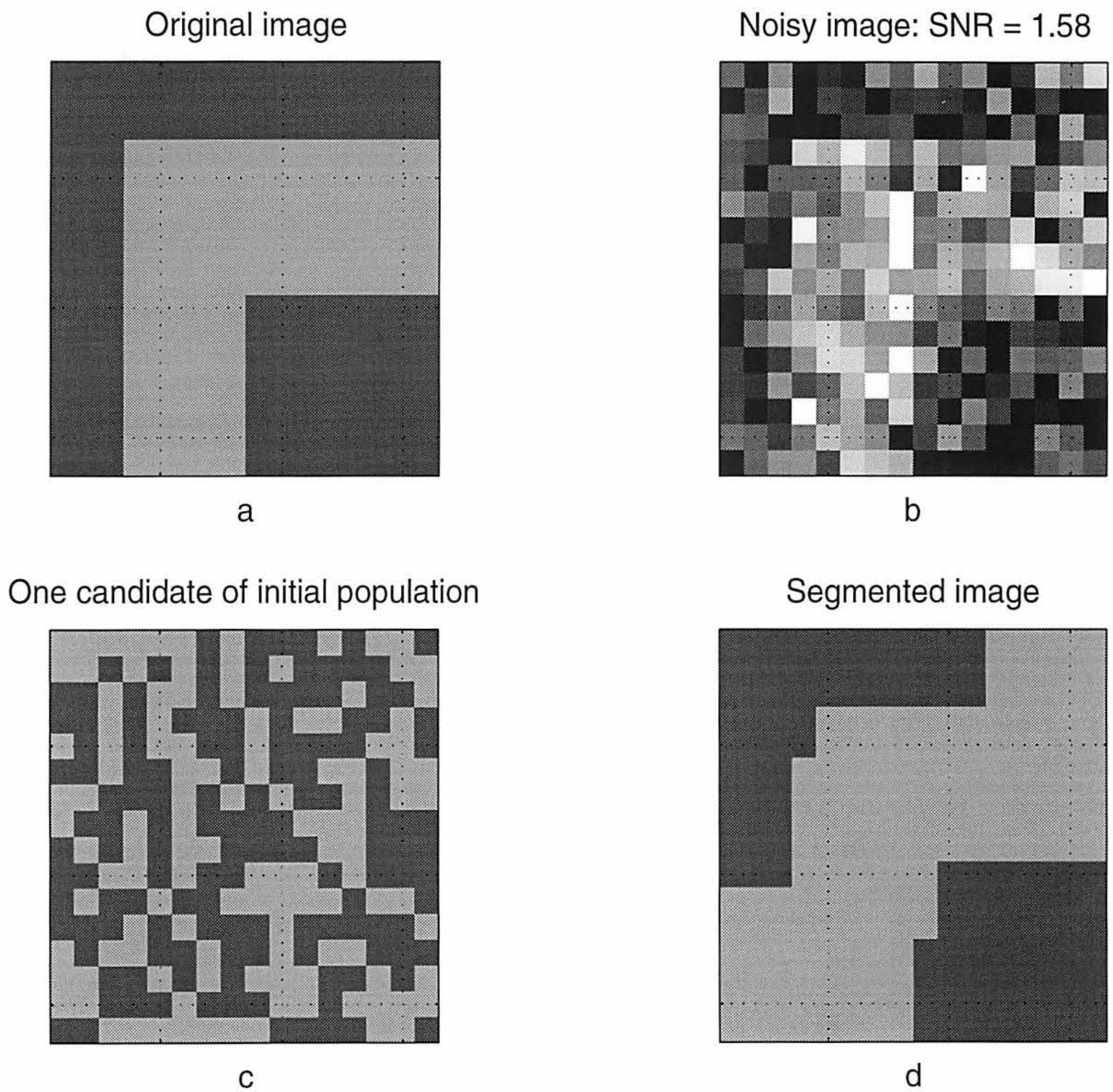


Figure 6.3 Results for 16×16 image with L shape object

(SNR = 1.58, population size = 50)

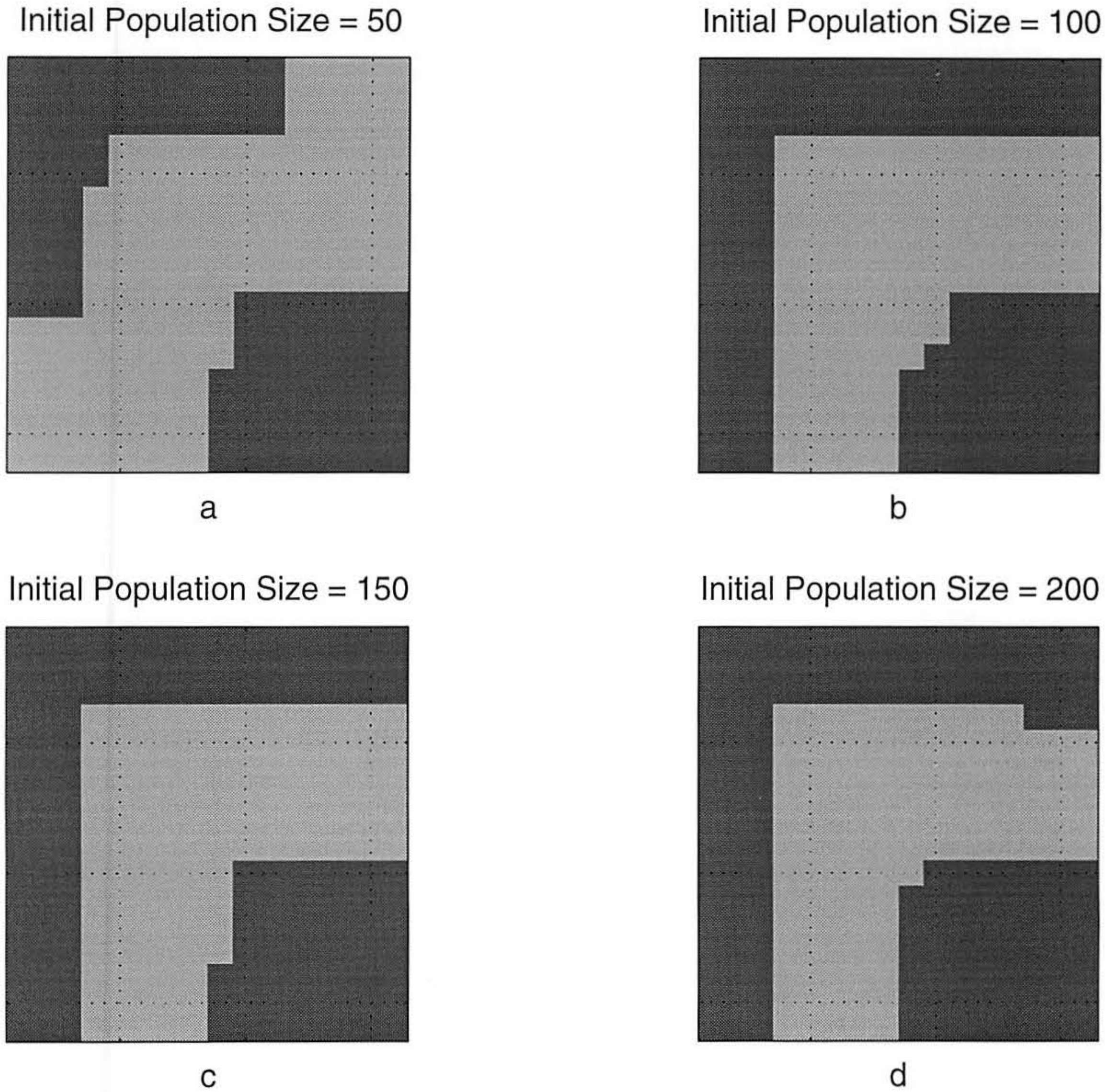
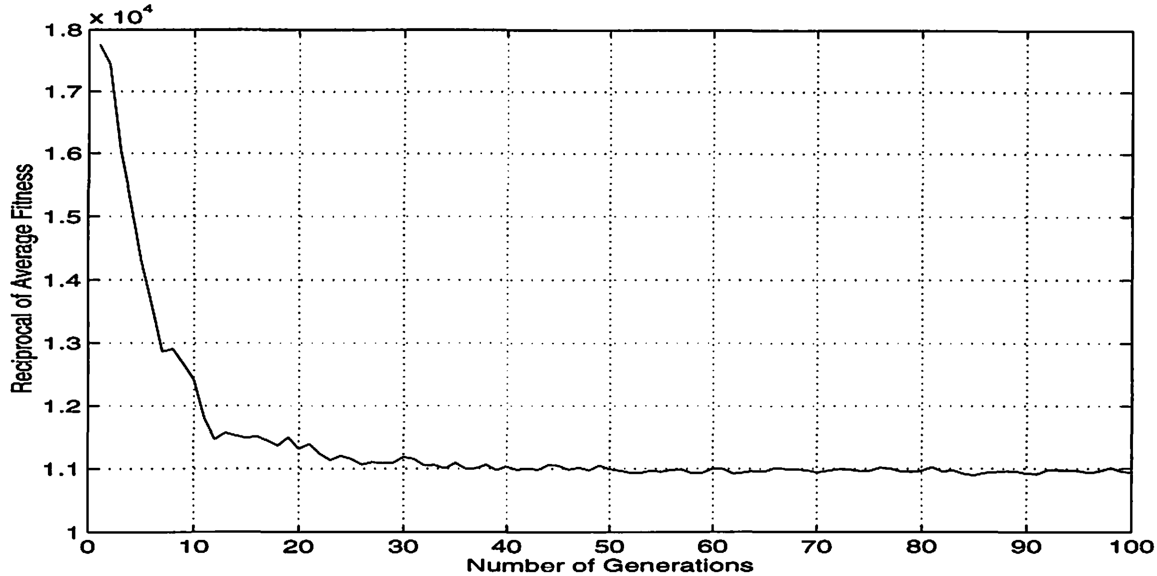
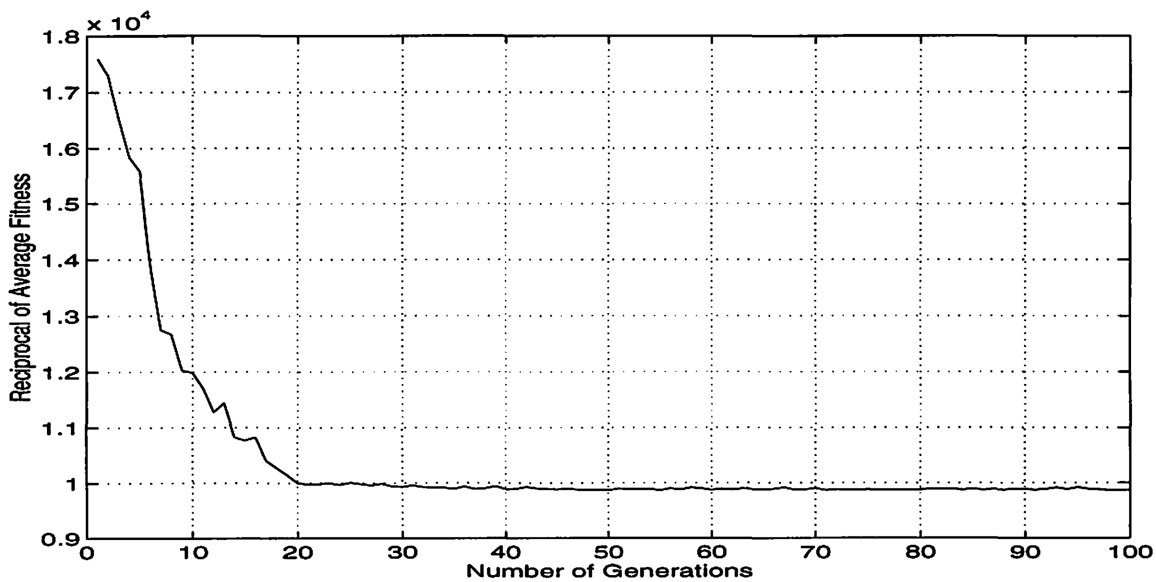


Figure 6.4 Comparison of segmented image using different sizes of initial population (SNR = 1.58)



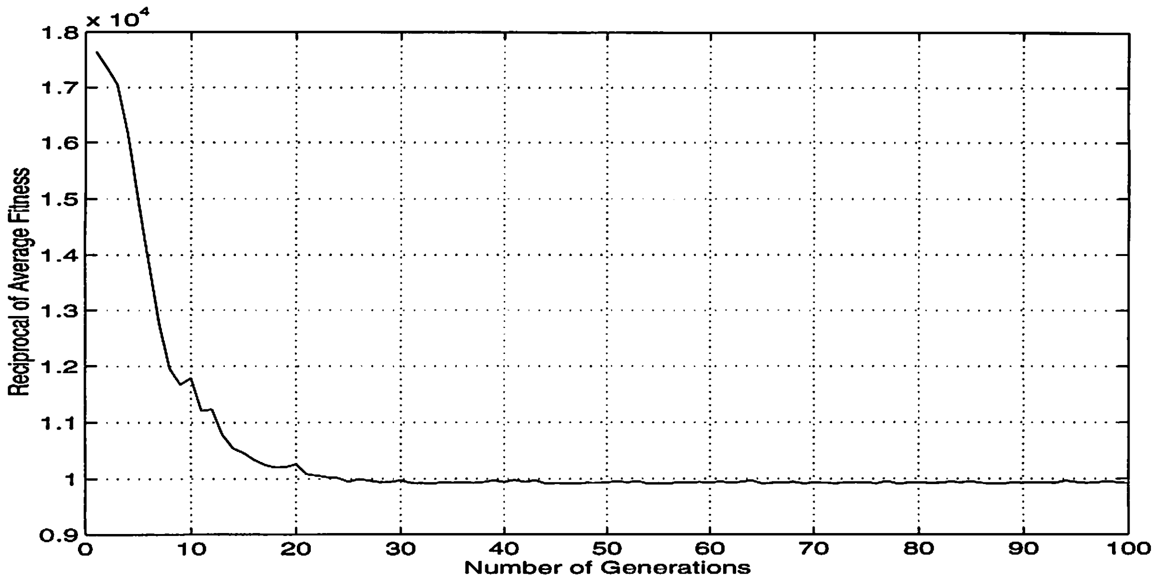
(a)



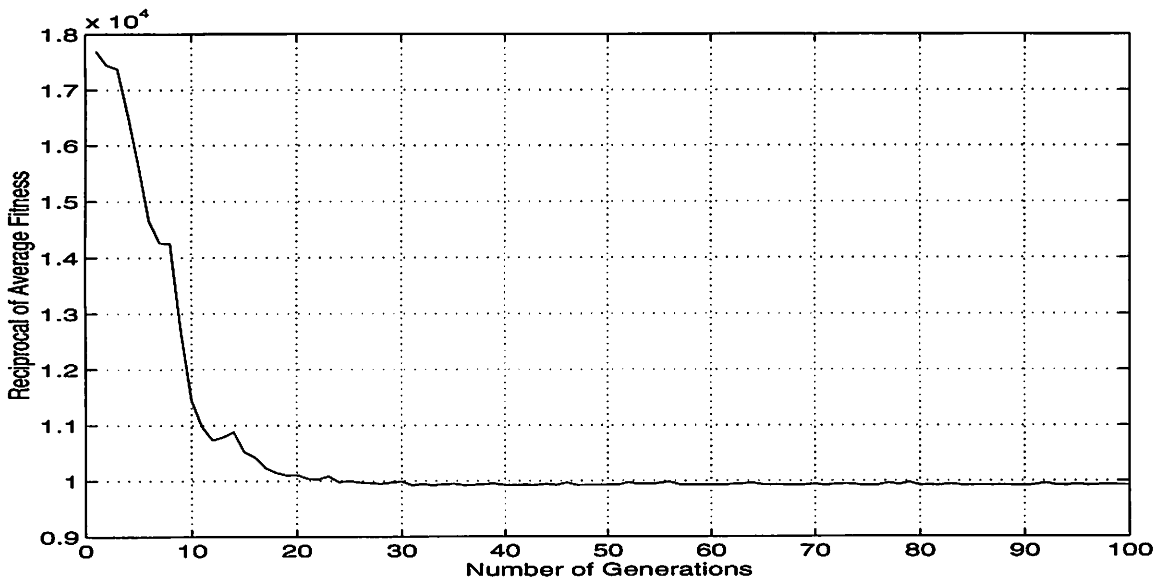
(b)

Figure 6.5 Reciprocal of average fitness for initial population:

(a) 50, (b) 100, (c) 150, (d) 200 with *SNR* = 2

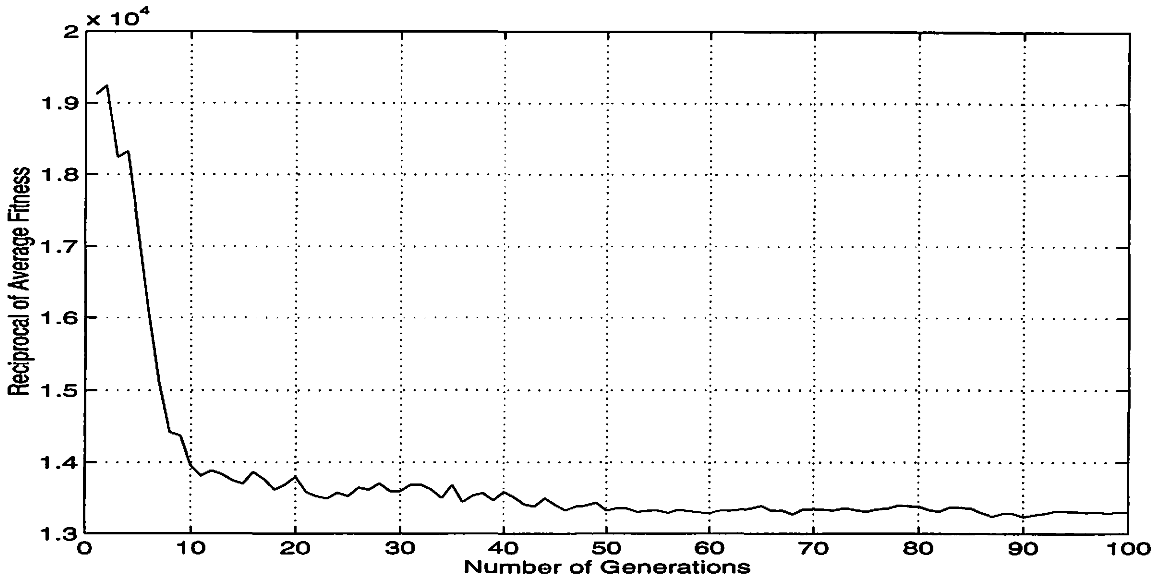


(c)

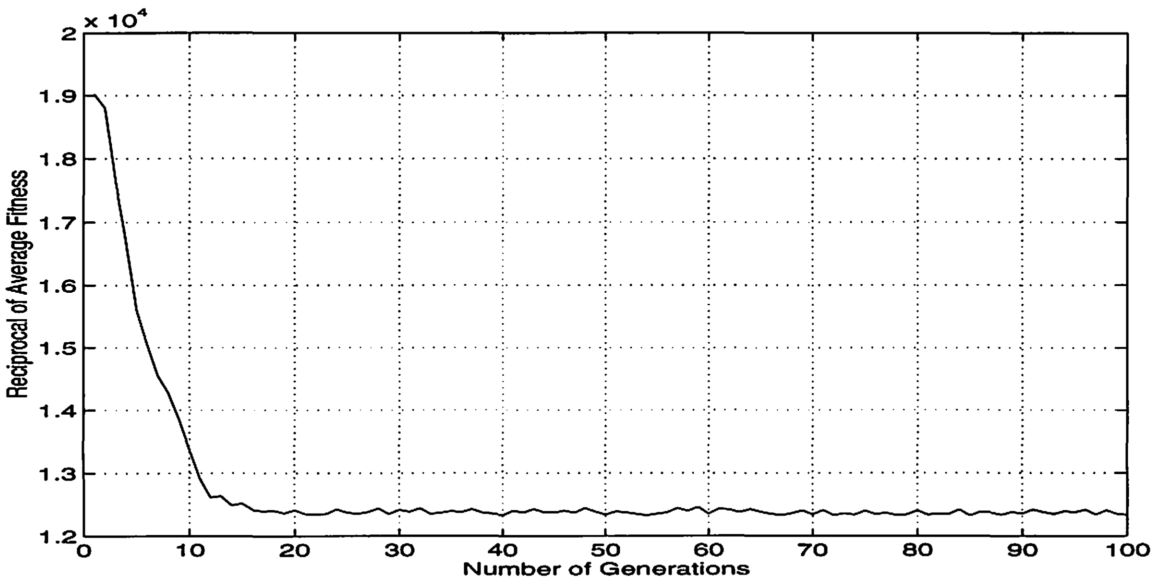


(d)

Figure 6.5 (continued)



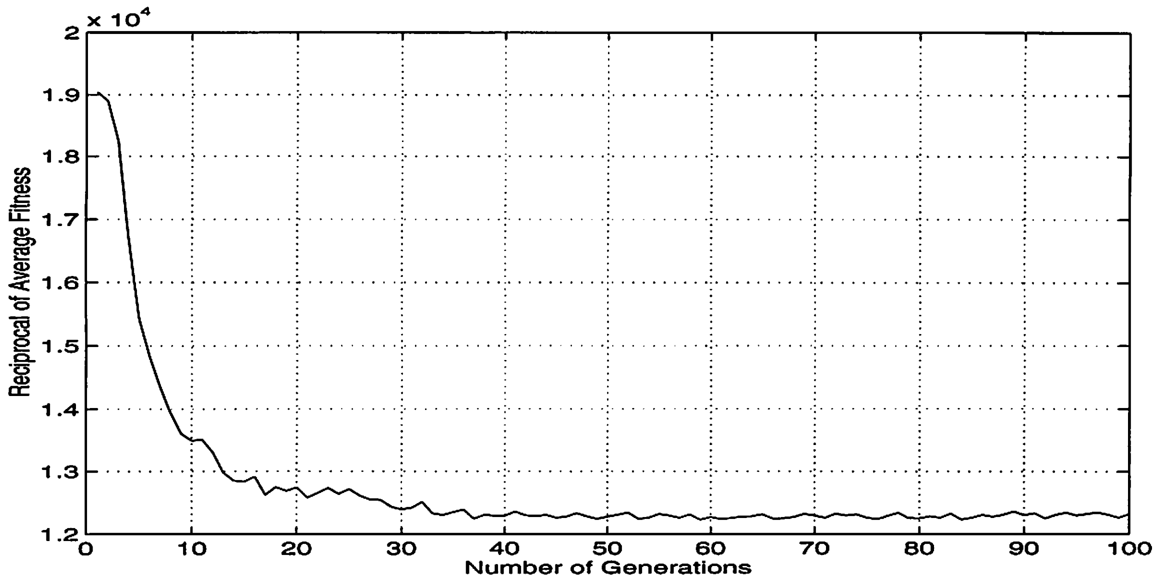
(a)



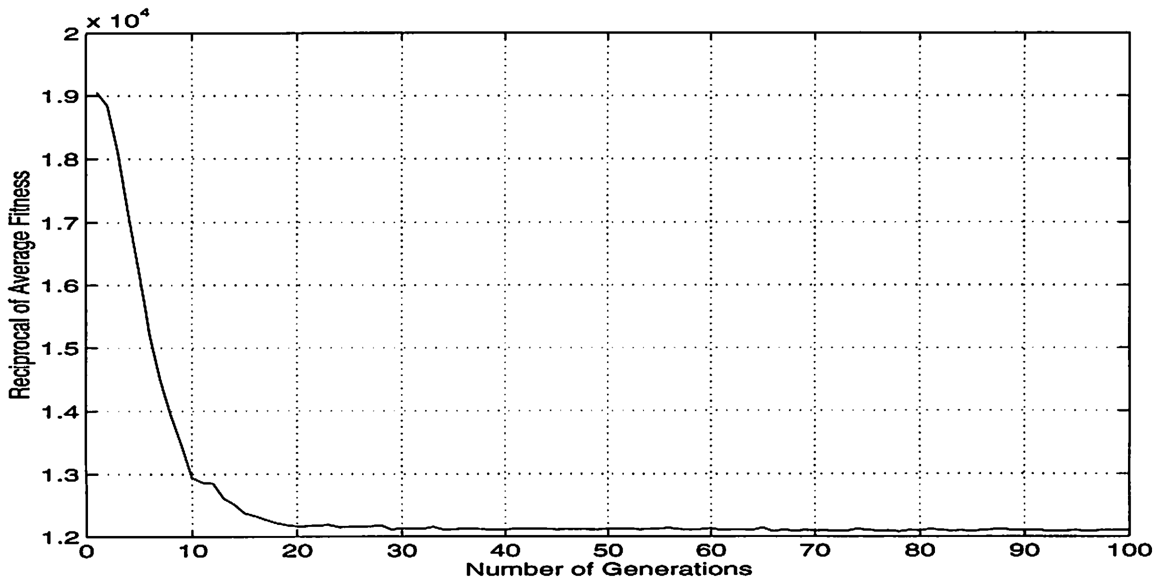
(b)

Figure 6.6 Reciprocal of average fitness for initial population:

(a) 50, (b) 100, (c) 150, (d) 200 with $SNR = 1.58$



(c)



(d)

Figure 6.6 (continued)

6.2 Number of Generations

The study was first conducted on 16×16 noisy image with *SNR* = 2. The initial population size was chosen to be 50, 100, 150 and 200. The algorithm was made to stop after 100 generations. Figure 6.5 shows the plots of reciprocal of the average fitness versus the number of generations. It can be seen from the plots, that the reciprocal of the average fitness decreases very steeply in the first 10 to 20 generations. After that the function remains almost constant. The study was then conducted on 16×16 noisy image with *SNR* = 1.58. The results are shown in Figure 6.6 with population size chosen to be 50, 100, 150 and 200. It is safe to say that the maximum fitness was attained after the fiftieth or sixtieth generation. The stopping criterion was therefore changed and the algorithm was stopped after 60 generations.

6.3 Reproduction

In the reproduction stage of the algorithm, all the individuals in the population are subjected to morphological processing (closing followed by opening). Then the fitness of all the new candidates is evaluated and those with the fitness better than the average fitness are selected for the next step: crossover. The original individuals and the offsprings generated by the crossover operation are then selected to maintain the same size of population based on the fitness evaluation. In this section, we first discuss the effect of structuring element size and value on the algorithm and then a new crossover technique is presented.

6.3.1 Structuring element size and element value

The size of structuring element used in the morphological operation is studied using the L shape image to find the best size of structuring element. The initial population size here is 100. For the noisy image with SNR = 2 and SNR = 1.58, the structuring element of size 3×3, 5×5 and 7×7 are tested in the algorithm. Figure 6.7 shows the result using structuring element size of 3×3 and element value of 100 for noisy image with SNR = 2. Figure 6.8 shows similar result on a noisy image with SNR = 1.58. Figure 6.9 shows the results for element size of 5×5 and 7×7 with element value 100 for the noisy image with SNR = 2. Figure 6.10 shows the results for element size of 5×5 and 7×7 with element value 100 for the noisy image with SNR = 1.58.

From these results, it is clearly indicated that for this image, the optimal size of structuring element is 5×5. However, it is very important to note that the optimal size of the structuring element depends strongly object size in the image.

To study the effect of structuring element size on this algorithm, different object noisy images were tested using size of 3×3 and 5×5 structuring elements with element value 100. All the other parameters are kept constant. In Figure 6.11 and 6.12, a two-block objects are presented in the image. Structuring element size of 3×3 and 5×5 were tried. As we can see in Figure 6.12, the quality of the segmented image is much better if the structuring element size reduced to 3×3. A curved object image is also studied using structuring element of size 3×3 and 5×5. The results are shown in Figures 6.13 and 6.14. Again, the quality of the segmented image using a 3×3 structuring element is much better, than that obtained using larger structuring elements.

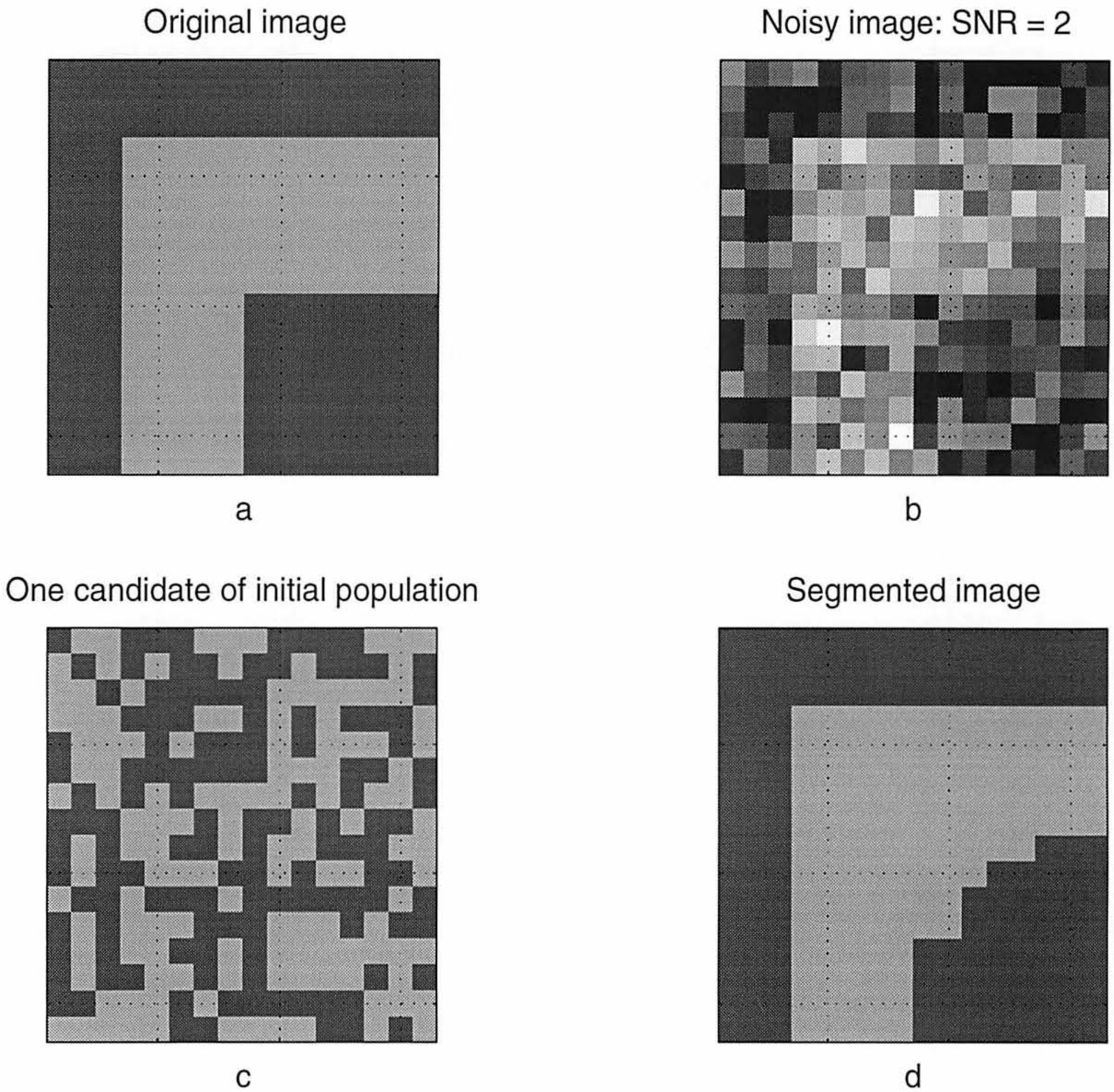


Figure 6.7 Results of 16×16 noisy image ($SNR = 2$) with structuring element of size 3×3.

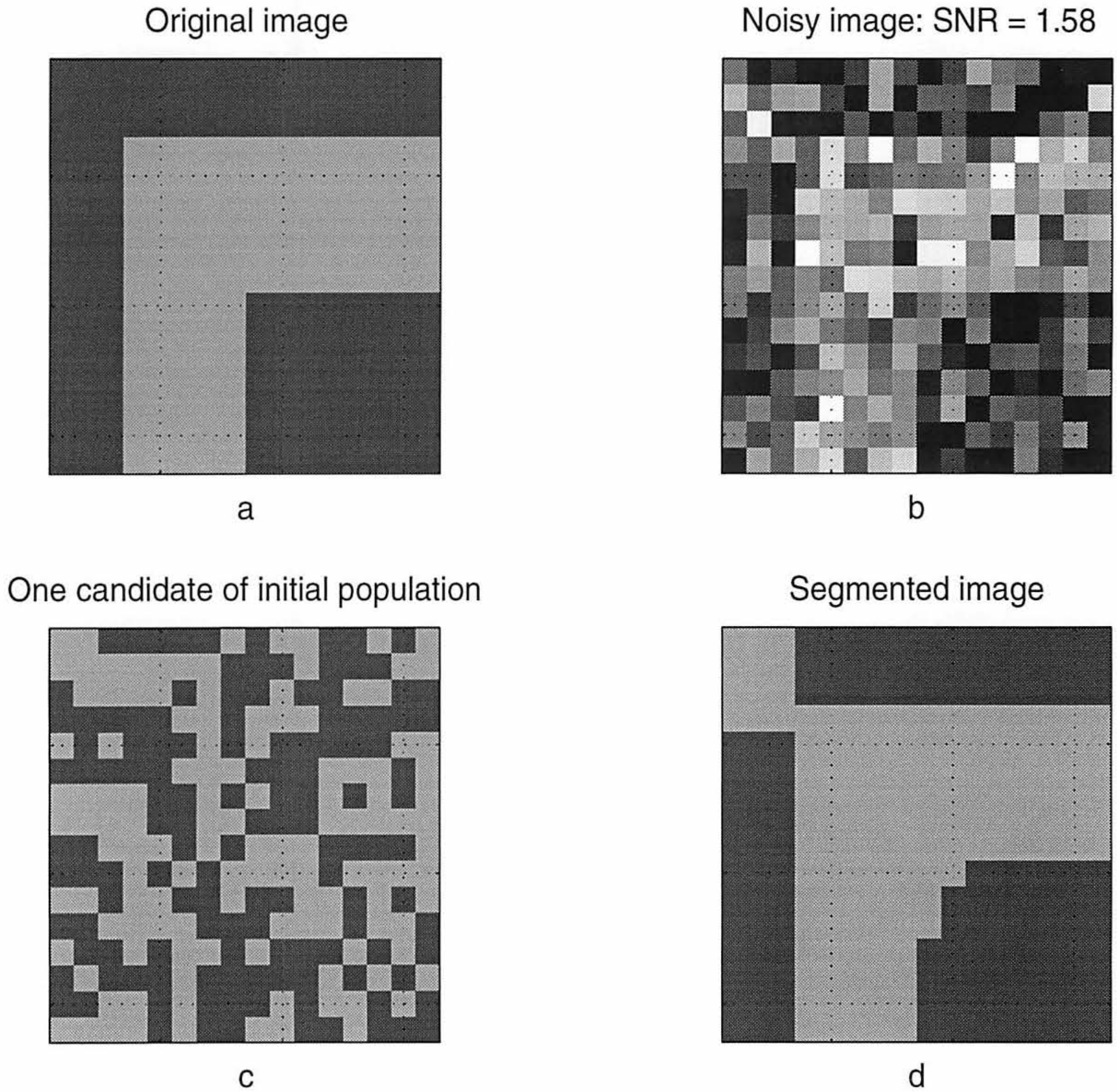


Figure 6.8 Results of 16×16 noisy image (SNR = 1.58)
with structuring element of size 3×3

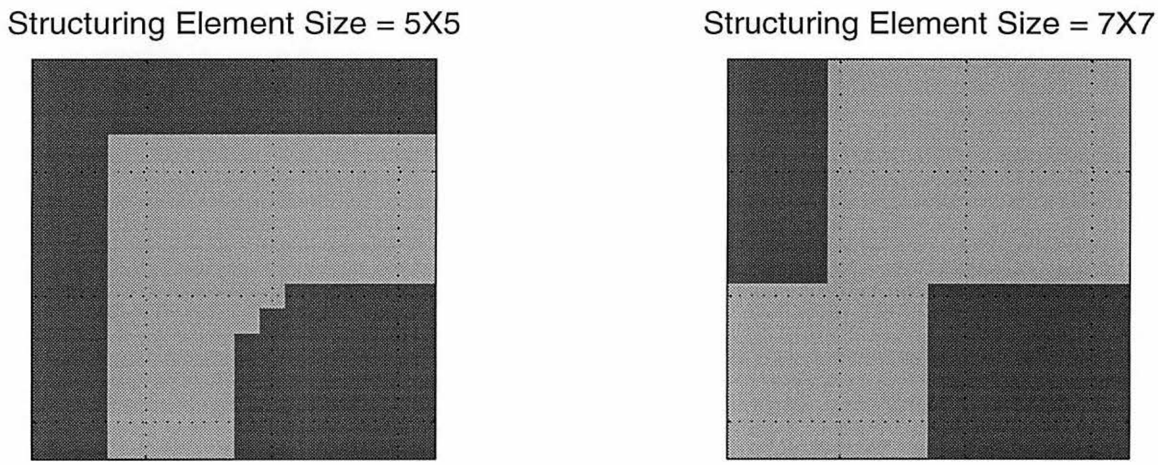


Figure 6.9 Results of 16×16 noisy image (SNR = 2) with structuring element of size 5×5 and 7×7

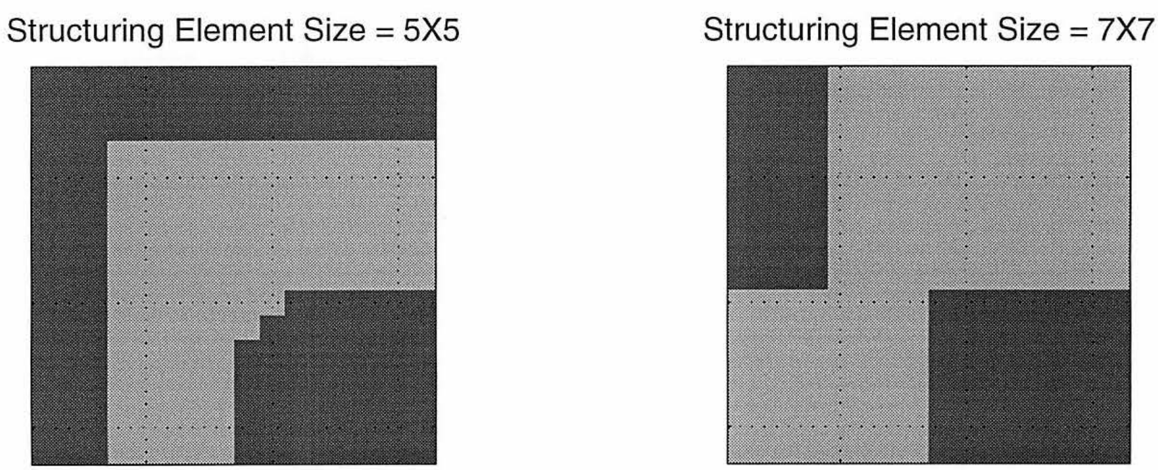


Figure 6.10 Results of 16×16 noisy image (SNR = 1.58) with structuring element of size 5×5 and 7×7

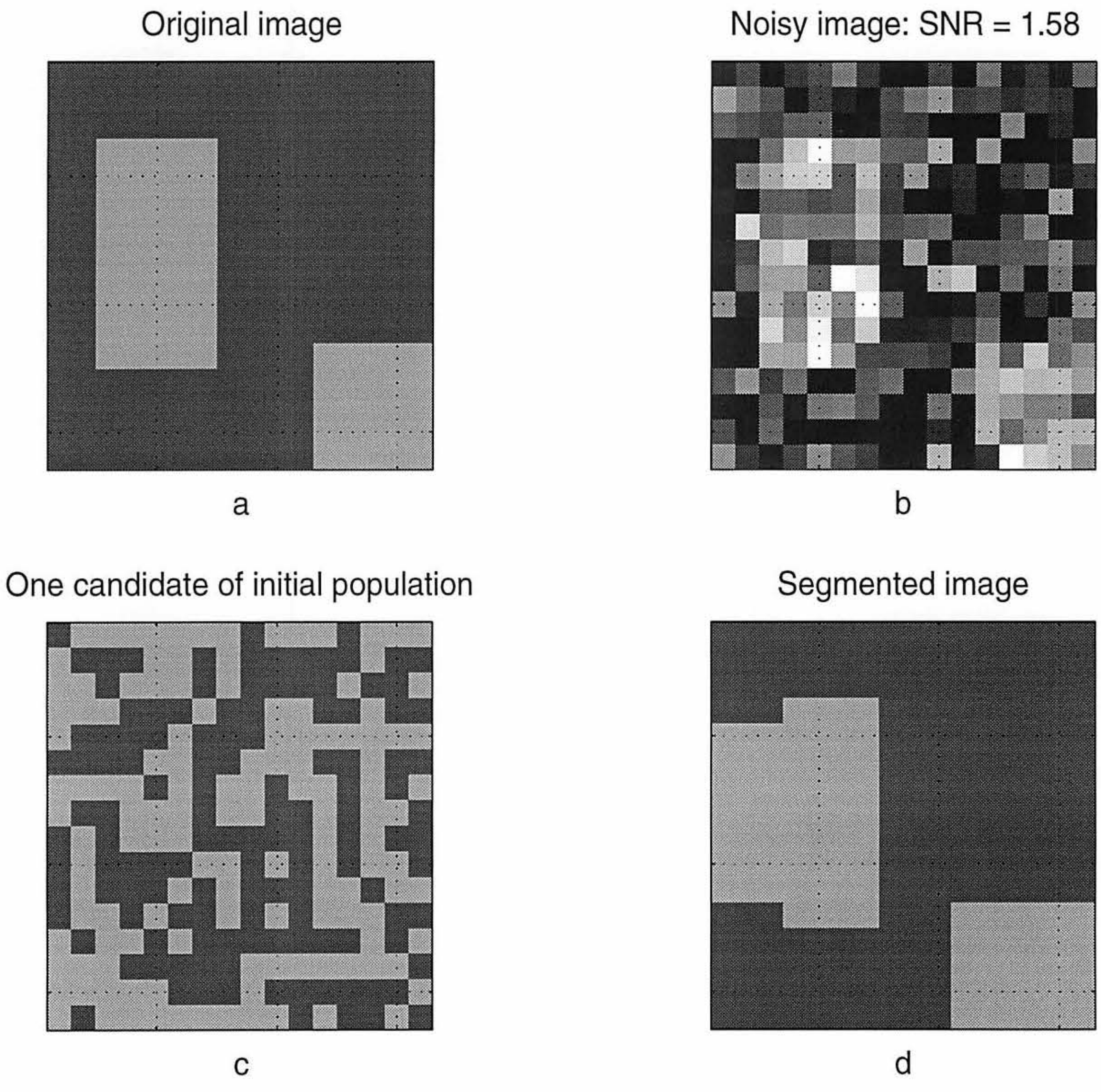


Figure 6.11 Results of 16×16 Two-Block object noisy image (SNR = 1.58) with structuring element of size 5×5

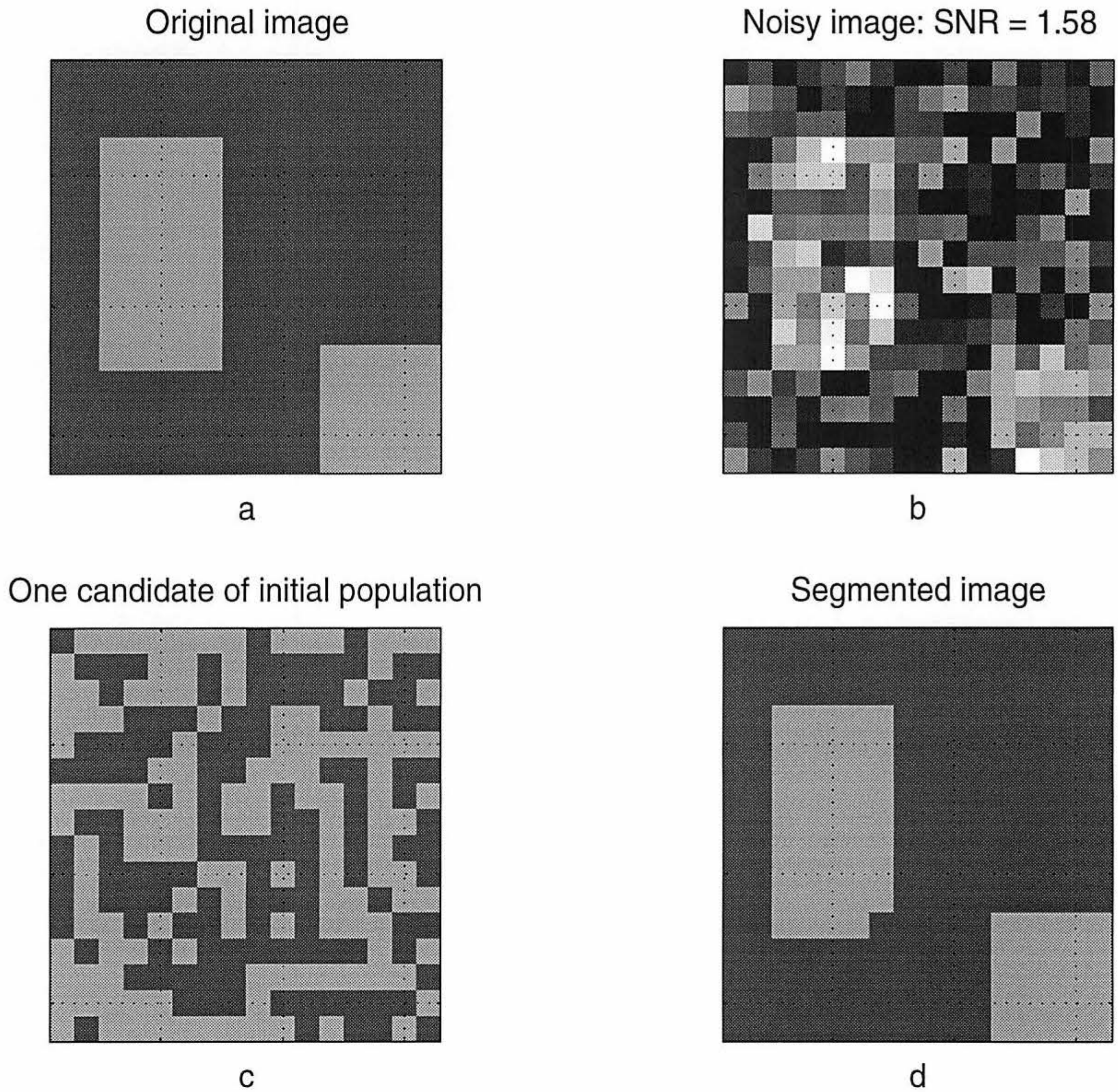


Figure 6.12 Results of 16×16 Two-Block object noisy image (SNR = 1.58) with structuring element of size 3×3

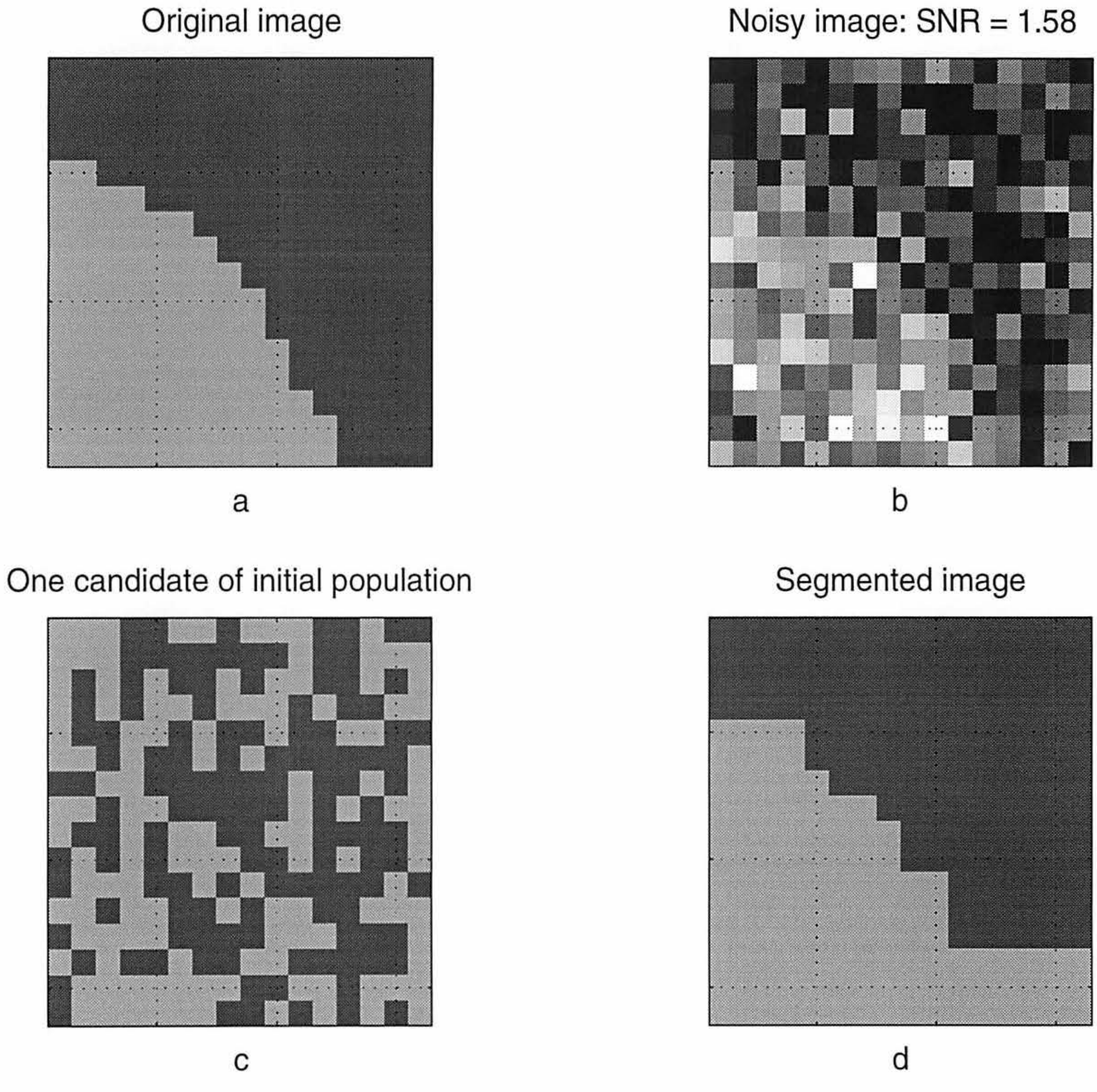


Figure 6.13 Results of 16×16 Curved object noisy image (SNR = 1.58) with structuring element of size 5×5

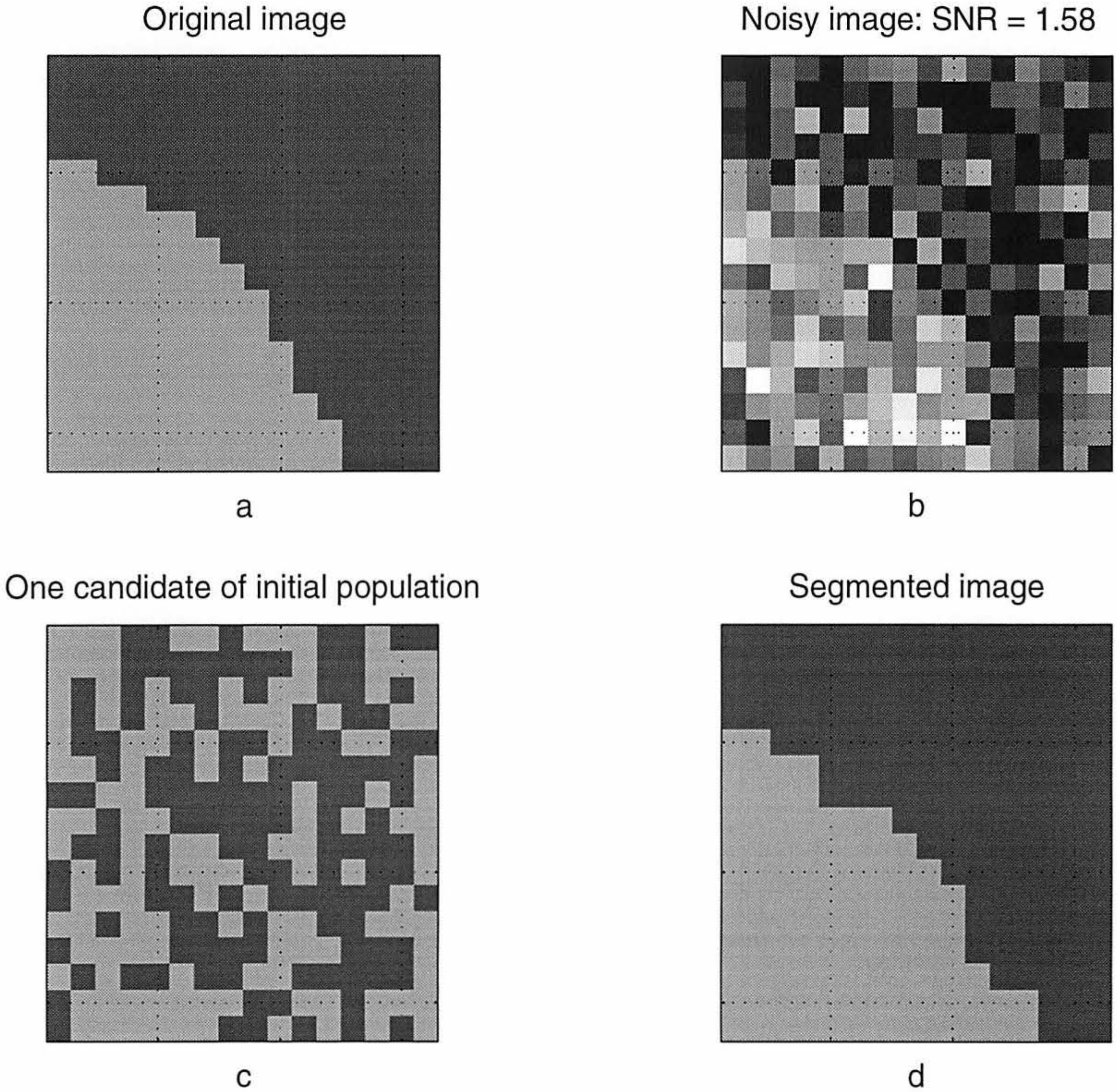


Figure 6.14 Results of 16×16 Curved object noisy image (SNR = 1.58) with structuring element of size 3×3

We also used threshold method with closing operation to segment the noisy image. The threshold value is 100, i.e., the pixel value is set to 150 for noisy image if it is great than 100, otherwise, it is set to 50. After the thresholding, the closing operation is applied to the thresholded image. The results are shown in Figure 6.15. It clearly indicate that threshold method with closing operation is not suitable for this problem.

Another important parameter is the value of the structuring element. The results for a 5×5 structuring element with element value 50 are shown in Figure 6.16. The results for the same structuring element size but with element value 500 applied to the same noisy image with the same initial candidates are shown in Figure 6.17. From these two results, it can be seen that a higher element value does not necessarily give a better result. Further investigation on image of Figure 6.18 was done with structuring element of size 3×3 and element value of 100, 0 and –50. Figure 6.18 shows the results with no morphological operation. Figure 6.19 shows the results after adding morphological operation with the 3×3 structuring element with element value 100. Figure 6.20 shows similar results obtained using two different element values namely, (a) element value equal to -50 and (b) element value equal to 0. From these results, one can see that the morphological operation does improve the quality of the segmented image. But so far, there is no definite procedure for selecting the optimal values of structuring element size as well as the element value.

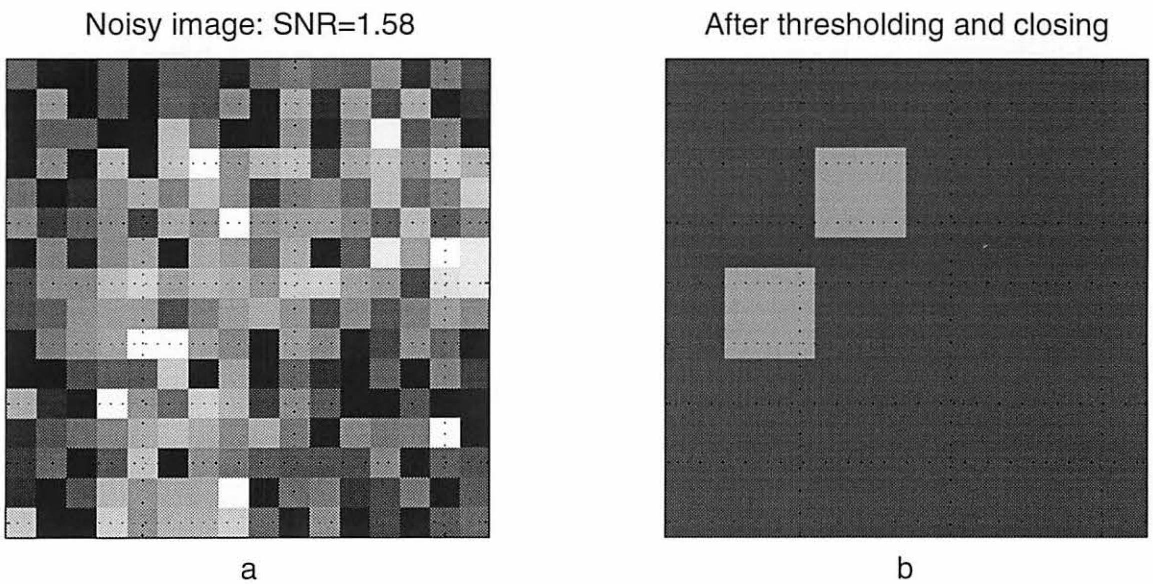


Figure 6.15 Results of thresholding method with closing operation

6.3.2 Crossover

The algorithm has been applied to 2D image data instead of the conventional 1D signal. Due to the complexity of the 2D image, the conventional two-point crossover is replaced by a new multi-point crossover technique developed to exploit the properties of 2D image.

Since the fitness of all the individuals are evaluated after the morphological operation, those with fitness better than the average fitness are selected for the crossover operation. After the parents are selected, two random numbers are generated to select two individuals

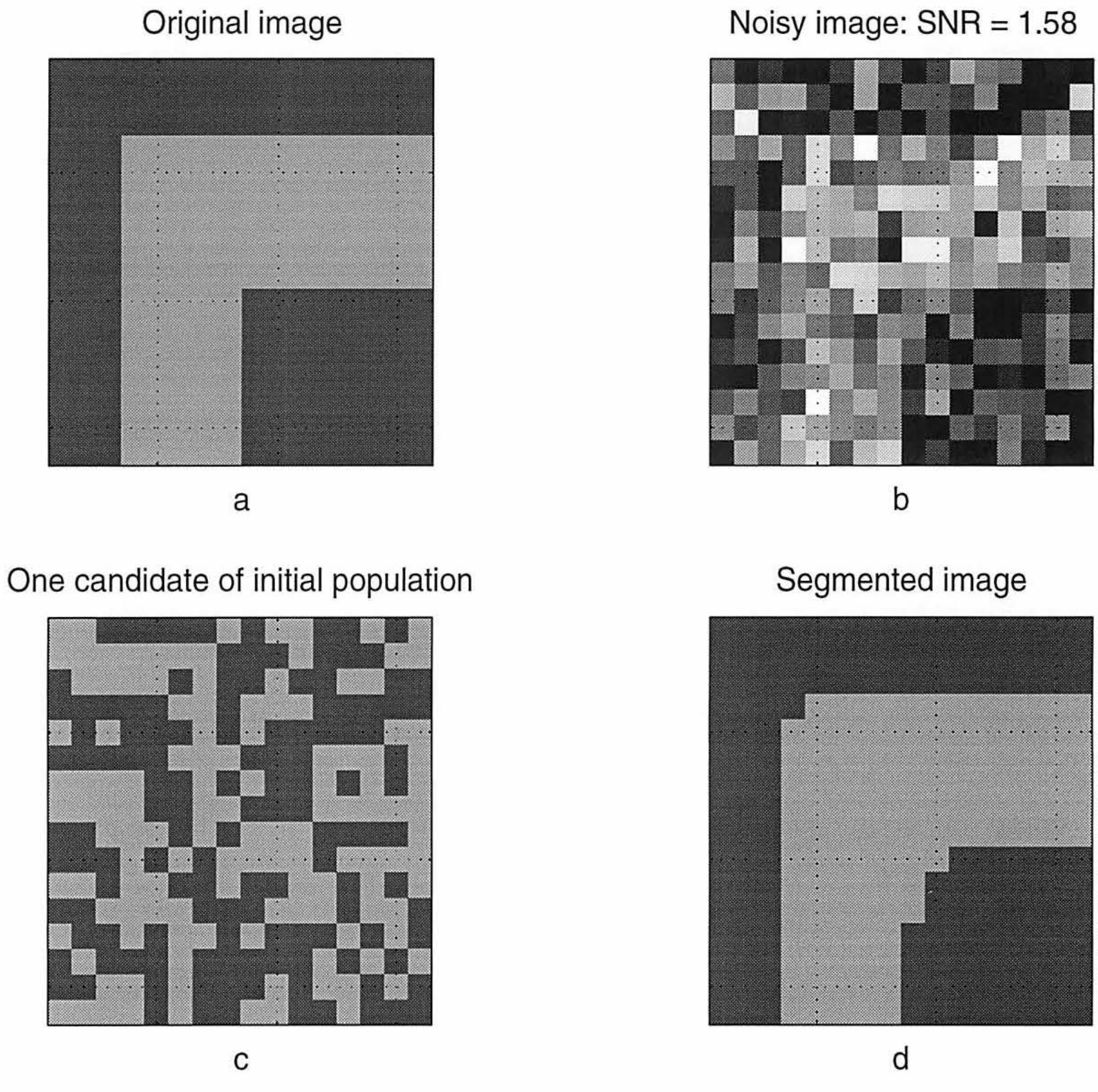


Figure 6.16 Results for structuring element of size 5×5 with element value 50

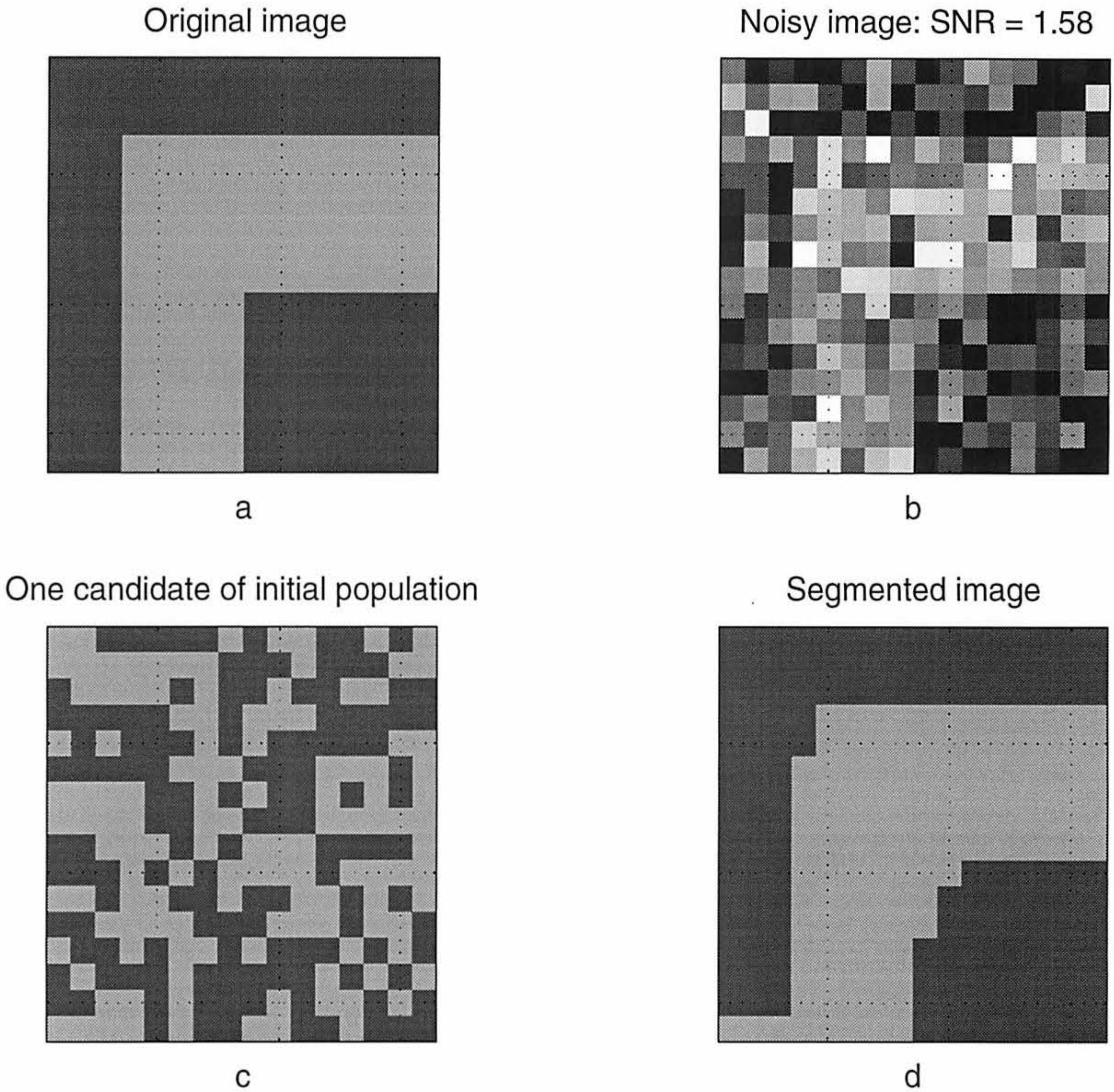


Figure 6.17 Results for structuring element of size 5×5 with element value 500

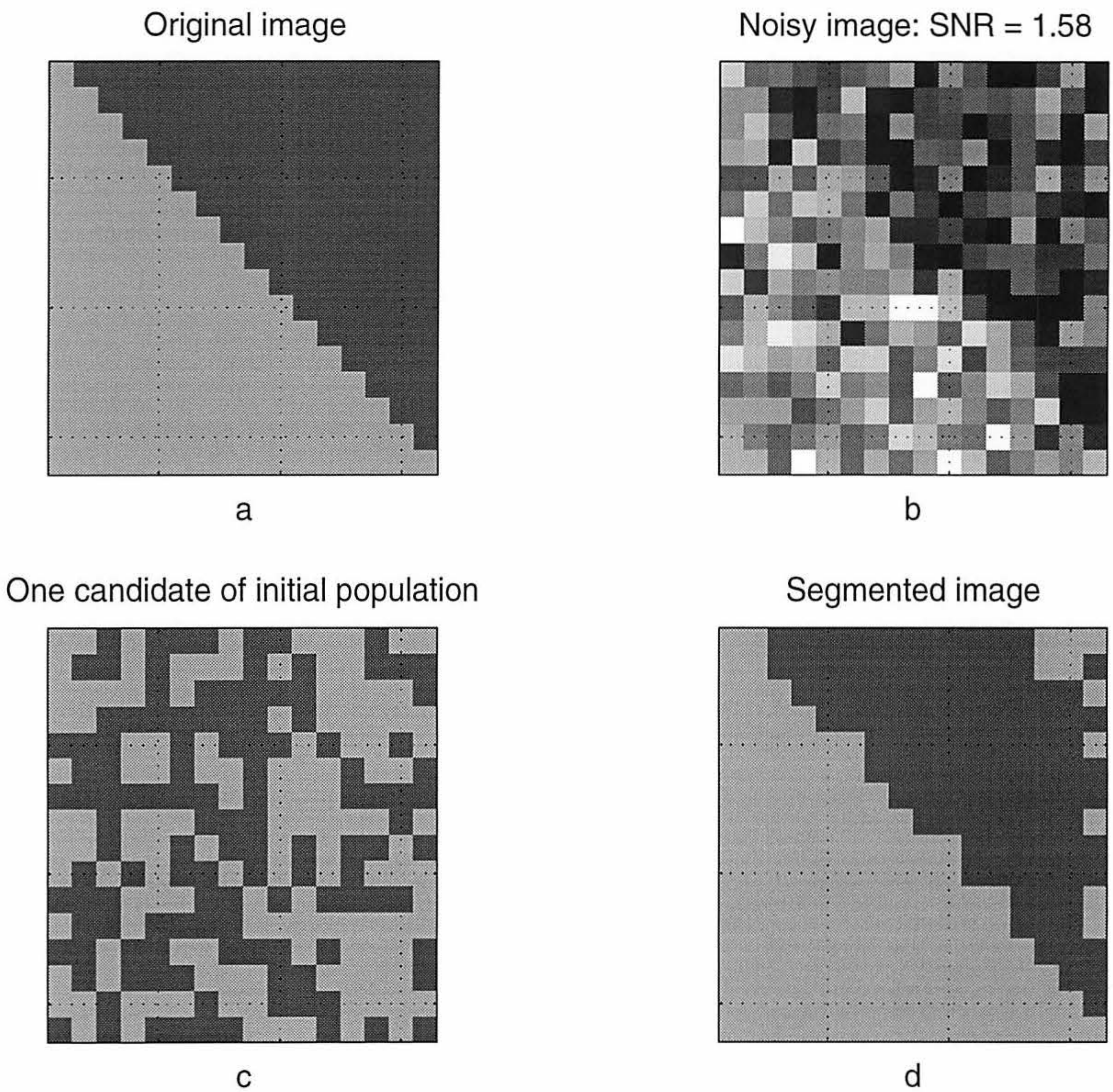


Figure 6.18 Results of Slope shape object with no morphological operation

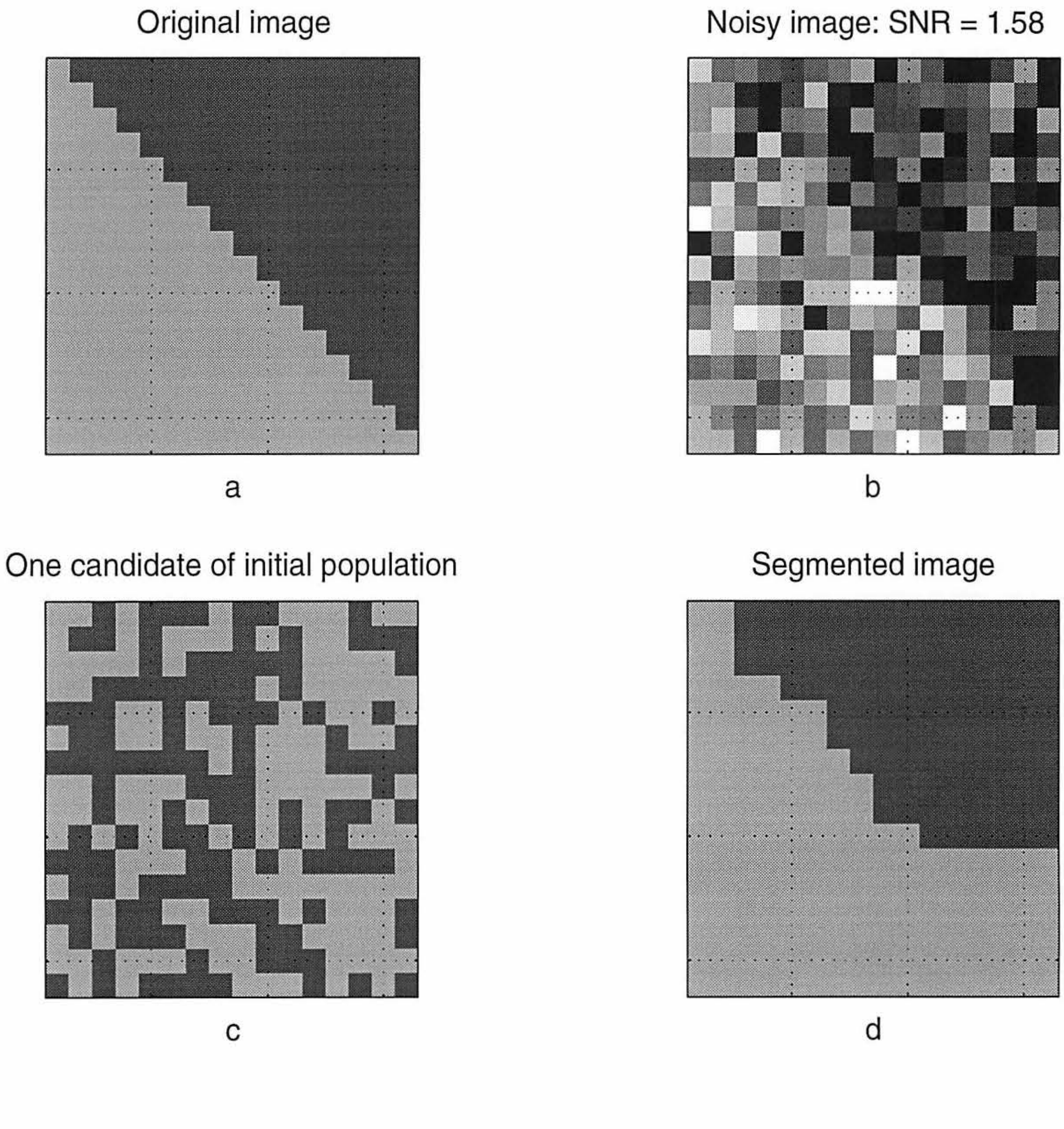


Figure 6.19 Results for structuring element of size 3×3 with element value 100

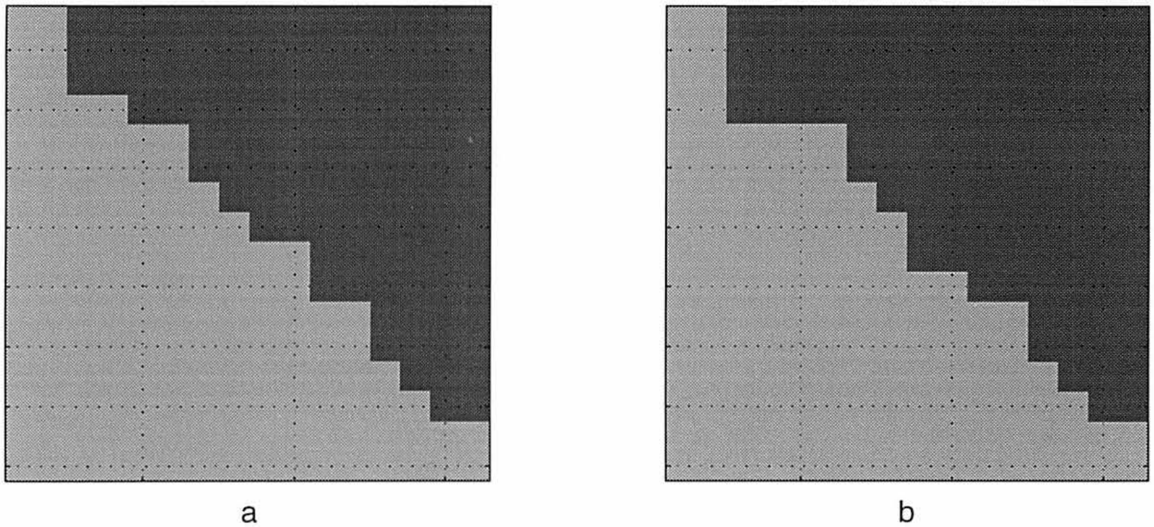


Figure 6.20 Results of Slope shape object with structuring element of size 3×3
(a) element value: -50 (b) element value: 0 (SNR = 1.58)

from these parents.

Assume the noisy image is $\bar{x}$, the two randomly selected candidates are Y^l and Y^m, where $1 \le l,m \le N$. N is the population size. The pixel-wise fitness is defined by finding the absolute difference between the noisy image pixel and the candidate image pixel as follows:

$$f(i) = |\bar{x}(i) - Y_i^k| \quad where \quad 0 \le i \le 255, \quad 1 \le k \le N \tag{6.1}$$

The average fitness is defined as:

$$f_{av} = \frac{1}{256}\sum_{i=0}^{255} f(i) \tag{6.2}$$

If the pixel wise fitness of the first candidate image is better than the average fitness of this candidate, the pixel value is kept. Otherwise, the pixel value is exchanged with the corresponding pixel value of the second candidate. By doing this, two offspring with multi-point crossover are produced. In some cases, the new multi-point crossover technique improves the quality of segmented image. Figure 6.21 and 6.22 show the results of these two crossover techniques applied on small object image. All the other parameters in the two algorithms are the same.

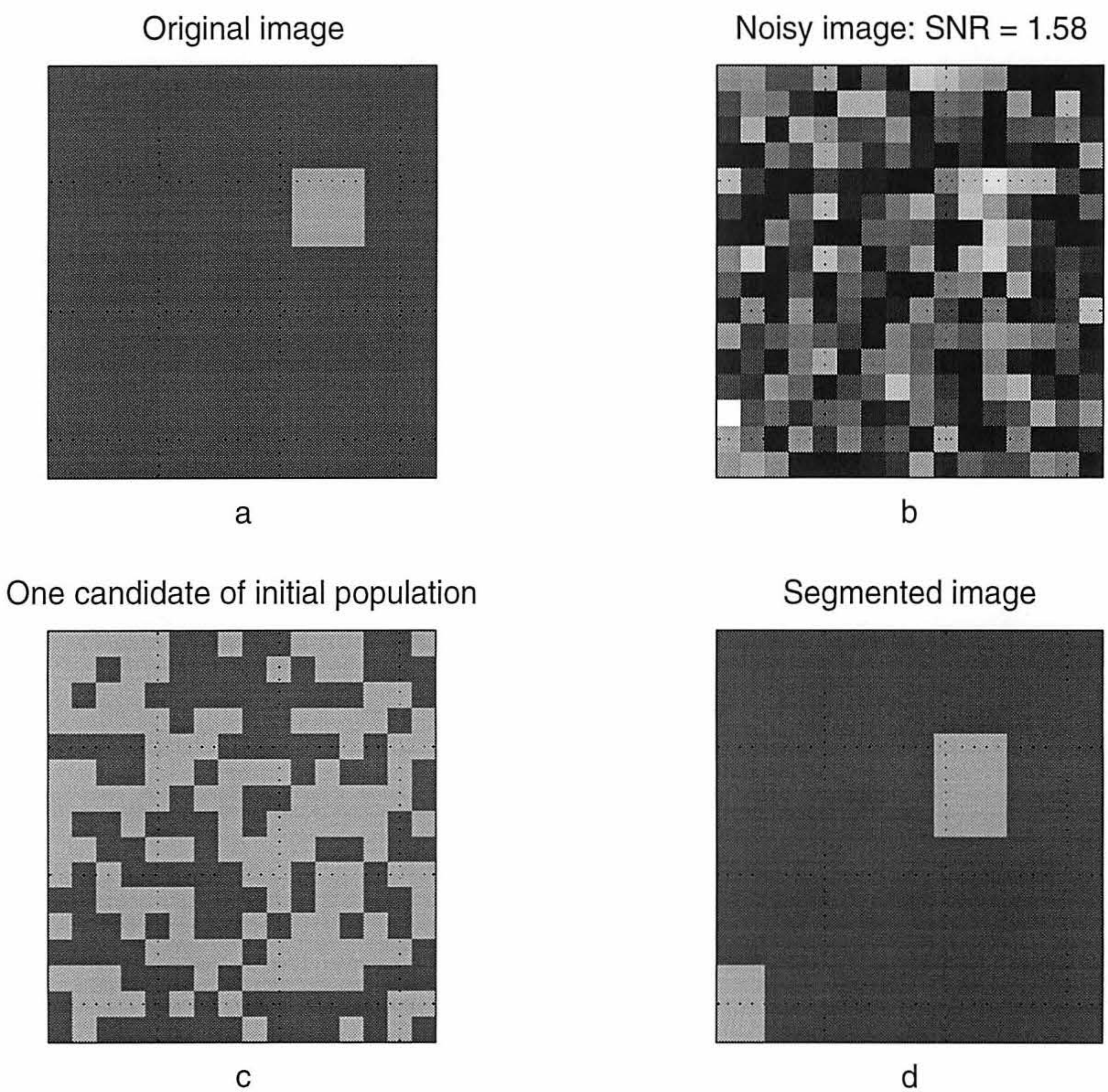


Figure 6.21 Results for two-point crossover technique with small object

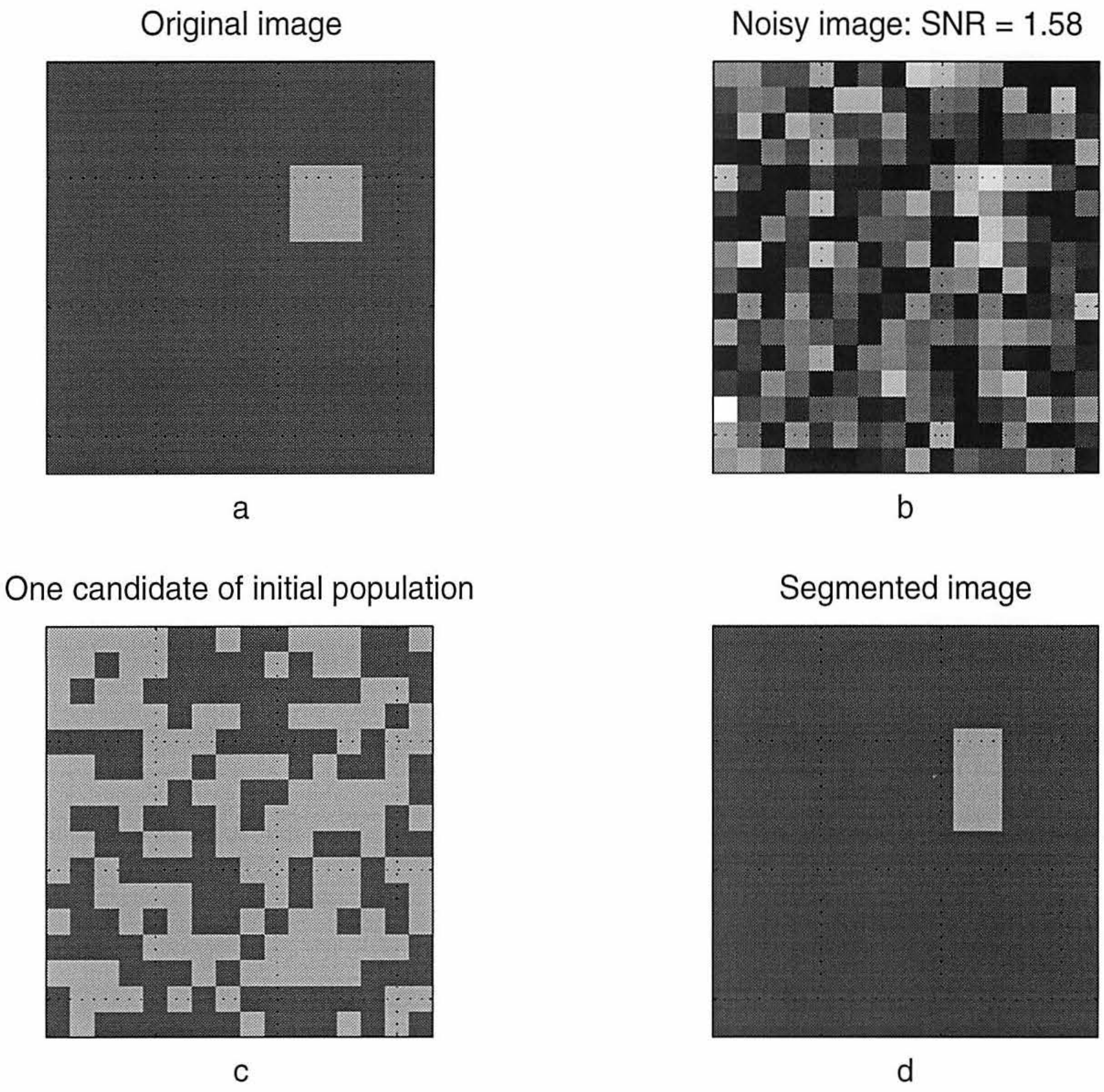


Figure 6.22 Results for multi-point crossover technique with small object

6.3.3 Mutation

The mutation operator used in this algorithm is a 4-neighborhood mutation. The reason for choosing this is that the 4-neighborhood mutation does not eliminate small objects. Another extended mutation operator is based on the 8-neighborhood and it only performs well on continuous large objects. Objects of small size have been studied to further investigate the mutation operator. The results are presented in Figures 6.23 and 6.24. For this small object, it is clearly shown that the 4-neighborhood mutation is much better than 8-neighborhood. After all the offspring are generated and the mutation operator is applied, the individuals for the next generation are selected based on their fitness.

6.4 Application to Larger Images

To apply this algorithm to large images, the larger image needs to be divided into 16×16 subimages first. The algorithm is then applied to these subimages and the segmented subimages are combined to form the final result. The study is conducted on a multi-object image with $SNR = 1.58$. The size of structuring element is 3×3 and the two-point crossover technique is used. The result shown in Figure 6.25 indicates that the algorithm successfully segmented the four objects with different shapes and sizes. Further investigation has been done with varying the structuring element size and the multi-point crossover technique. The results shown in Figure 6.26 used both 5×5 and 3×3 structuring elements on the subimages and those with better final average fitness are selected and are combined to form the final segmented image. At the same time, the multi-point crossover technique is used for all the

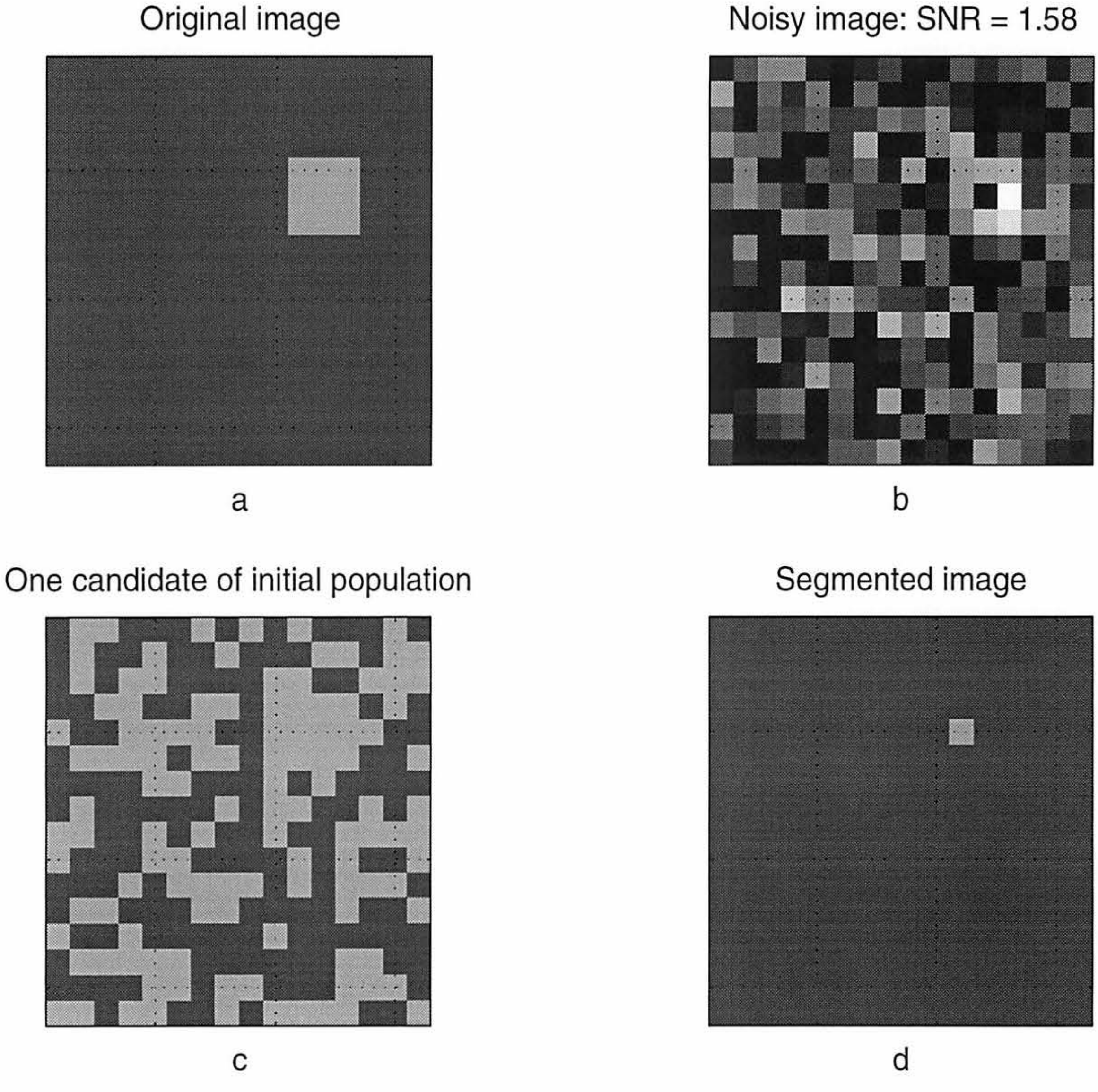


Figure 6.23 Results of 8-neighborhood mutation for small object

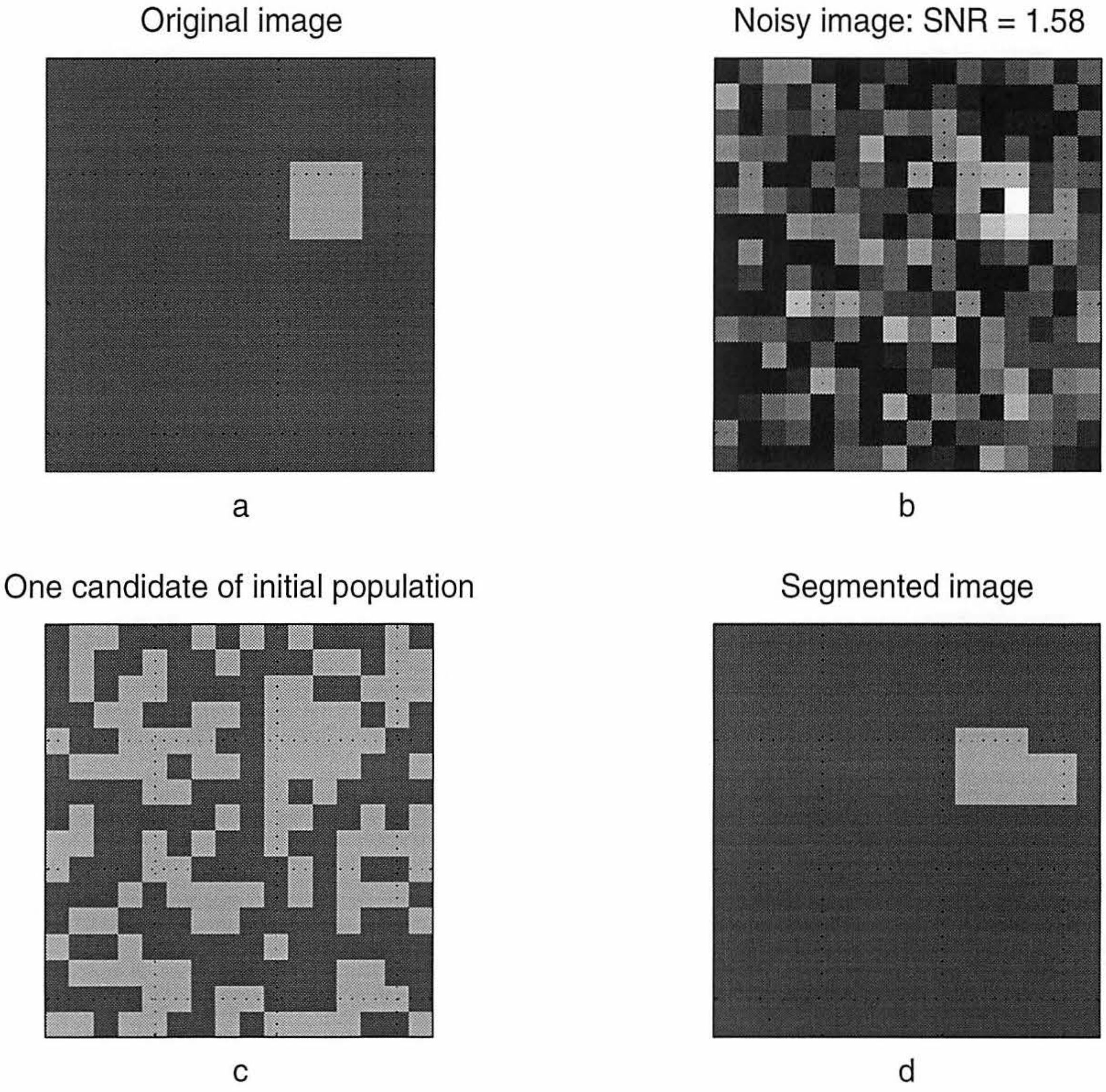


Figure 6.24   Results of 4-neighborhood mutation for small object

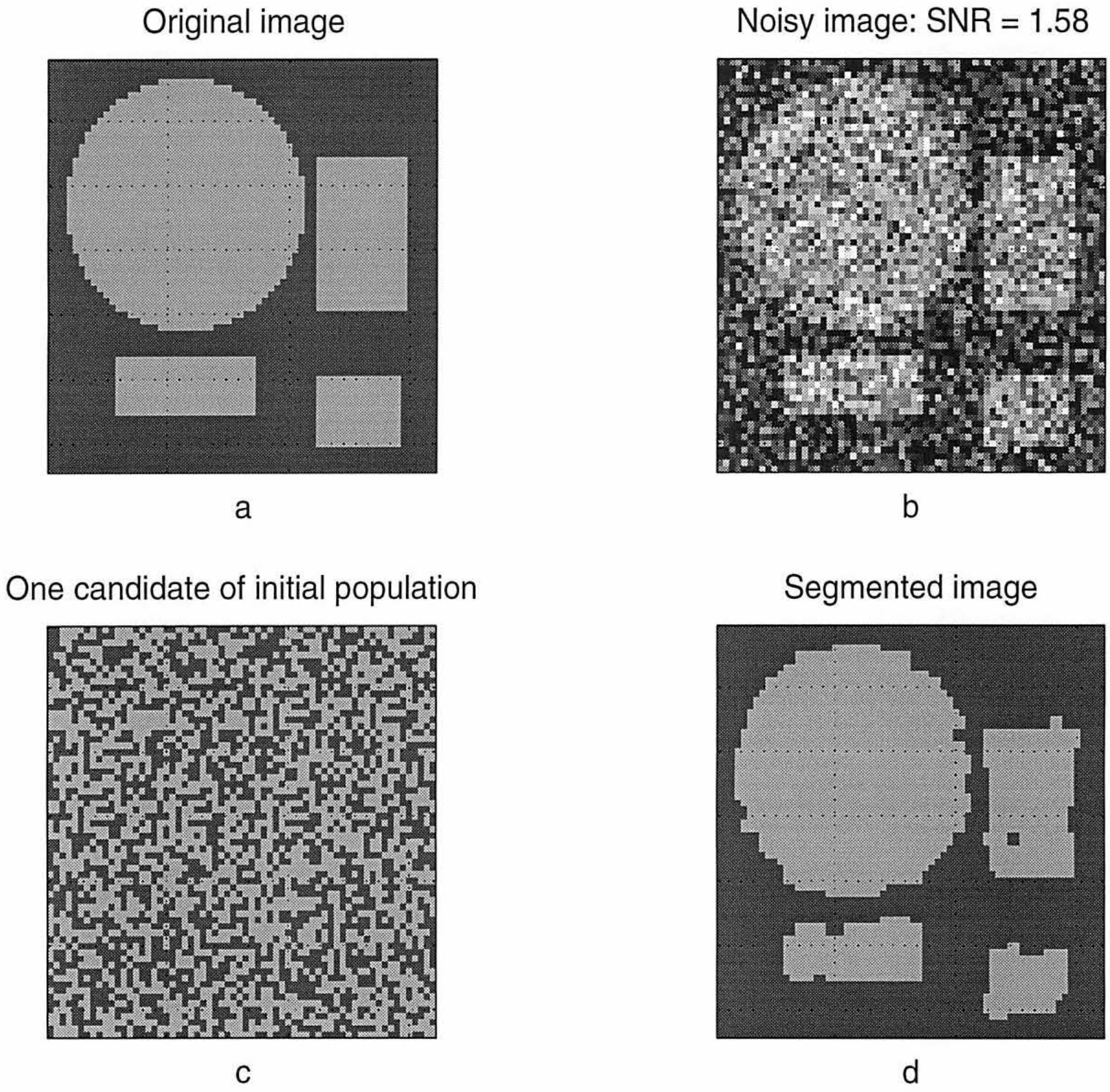


Figure 6.25 Results of 64×64 image with two-point crossover and structuring element of 3×3

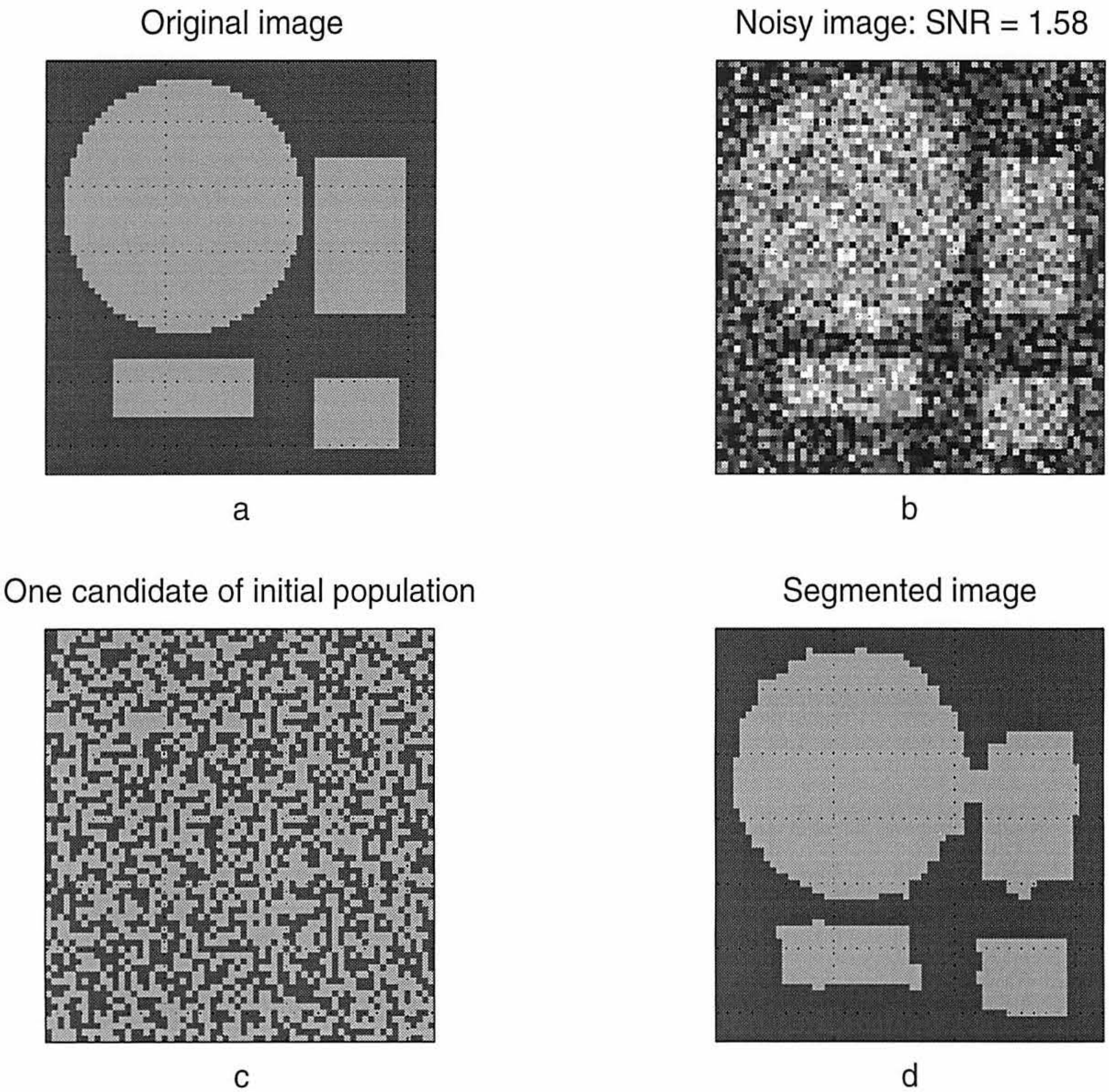


Figure 6.26 Results of 64×64 image with multi-points crossover and varying structuring elements (3×3 or 5×5)

subimages. The segmented image in Figure 6.26 is more accurate even though there is a misconnection between top two objects.

6.5 Application to Low Contrast Images

In low contrast images, the object intensity (R_o) and background intensity (R_b) are 125 and 75, respectively. A random gaussian noise is added to the images with σ^2=2500 to get a *SNR* = 1. The algorithm with new multi-point crossover technique was used. All the other parameters were optimized as discussed in previous sections. The results are shown in Figure 6.27. It is clearly shown that the algorithm works well on low contrast and low *SNR* images.

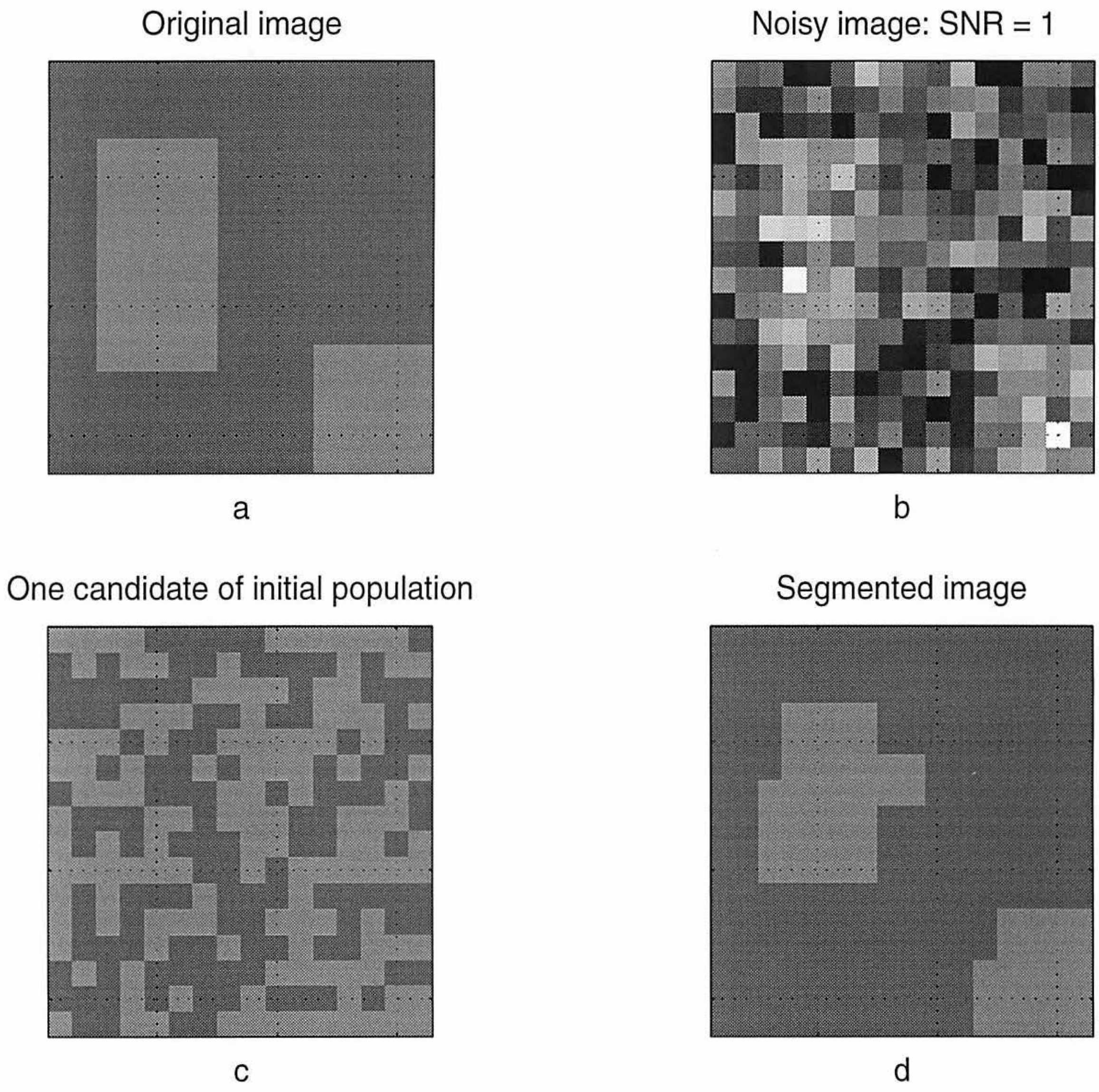


Figure 6.27 Results of low contrast images ($SNR = 1$)

CHAPTER 7. CONCLUSION AND FUTURE WORK

The final goal of this thesis is to develop and study the feasibility of an Image Segmentation technique that is based on the Genetic Algorithm and Mathematical Morphology. In this approach, image segmentation is considered as an optimization problem and it is carried out using genetic algorithms. Genetic algorithms are based on the evolution theory also known as "survival of the fittest" principle. In genetic algorithm, an initial population of candidates is required to start the evolution. All these individuals are generated based on the problem to be solved. In the process of reproduction, the morphological operations are used before crossover and mutation. The effect of mathematical morphology is to fill small holes and eliminate the small speckles. After the morphological operation, the fitness of all the individuals is evaluated and only the fitter candidates are allowed to produce offsprings. The offspring so created tend to inherit the "best feature" of their parents. Generation after generation, the overall fitness of the entire population is improved and finally each individual in the population is as good as others in the sense of being the fittest individual.

7.1 Summary

The proposed approach has been applied to 16×16 noisy images with pixel intensity varying from 0 to 255. The size of the subimage is optimized based on the experimental results and the computation time. The algorithm can be applied to any size images as long as these images are divided into 16×16 subimages. These subimages can be processed independently and can be combined to form final segmented images.

Initially, the populations of size N are generated randomly based on the knowledge of the object and background intensity. The size of the population is pre determined and all the individuals are represented in a string format with the same size as the subimage. Instead of having various intensities as in the noisy subimage, the candidates are binary strings of value R_o (object intensity) or R_b (background intensity).

The most critical step in GA is the reproduction because it is responsible for evolution. There are three steps in reproduction (1) mathematical morphology to produce parents from the individuals in the population, (2) crossover of the "fitter" parents to produce offsprings and (3) mutation of top ranking individuals chosen from old population and the new offspring. The morphological operation is necessary for large objects because it tends to remove small spikes in the background and fill in small holes in objects (usually less than 3×3 pixels). After the morphological operation, the fitness of all candidates is evaluated based on the fitness function and a certain number of candidates are selected to the mating pool. In mating process, two different crossover techniques are presented. In one of the techniques, the two parents are randomly chosen and the two cross points are randomly generated. In the other technique, after the two parents are randomly chosen, the pixel wise fitness is compared to the average fitness. Based on the difference, two new offsprings are produced. After the fitness of the offspring is evaluated and the top ranking individuals are selected, the mutation operator is applied to the population to add some diversity and form the next generation. In mutation operation, a 4-pixel neighborhood is defined and the value of the pixel is flipped depending on majority pixels in the neighborhood.

The next generation has the best N (population size) candidates from the previous generation and their offsprings. The algorithm is terminated after a fixed number of generations or convergence of the average fitness value is achieved.

7.2 Conclusion

An exhaustive parametric study of the algorithm on image segmentation was performed. The parameters that affect the performance of the algorithm were studied and discussed. The following observations can be made:

1. The initial population is generated randomly and no knowledge of the noise distribution is needed.
2. The optimal population size in each generation is seen to be around 100.
3. The algorithm was terminated after 60 generations because the results do not show significant improvement after 50 generations.
4. The optimal size of the structuring element is 5×5 for the 16×16 subimage with single object or two objects away from each other. But for small object or multiple objects which are close to each other, it is better to use a 3×3 structuring element.
5. The new multi-point crossover technique gives better results than the two-point crossover technique. It also brings the diversity into the population. This is one of the reason that it gives better results.
6. The algorithm is simple, fast and works very well on low signal-to-noise ratio image and on large size image.

7.3 Future Work

Future work must be mainly concentrated on the improvement and efficiency of the algorithm. There are several aspects which may improve the algorithm:

1. Different methods to generate initial population. Instead of randomly generating the initial population, a certain probability may be used depending on the neighborhood of the corresponding pixels in the noisy image. An alternative choice is a threshold image.
2. Different fitness function to evaluate the fitness of the candidates. More precise measurement may be used instead of just using the summation of pixel wise error.
3. Different selection criteria to select individuals for the mating pool. After the fitter individuals are selected, a small number of randomly selected individuals should be selected to bring more diversity into the mating pool.
4. Parallel genetic algorithm can be implemented. Since the evaluation of the fitness is the same with all the individuals, it can be done parallel to increase the speed of the algorithm.
5. The gradient searching method can be combined with GA in the algorithm to save the search time and avoid stuck in the local extrema.
6. The population size need not be kept constant. In each generation, the population size may be changed depending on the quality of the individuals in the population. At the same time, new candidates may be introduced into the population to diversify the population (i.e. to jump out of the local extrema).

7. The stopping criteria can be varied. Currently the evolution is stopped either after a certain number of generations or when the average fitness is almost a constant. But in reality, the average fitness oscillates back and forth within certain range. A new stopping criteria need to be developed to include this phenomenon so that the algorithm can stop whenever oscillations occur.

In the real segmentation problem, the object and background intensity may not be known in advance. Furthermore, the image may have multiple objects with different object intensity. In this case, the histogram may be calculated for subimages to get the object and the background intensity. But this depends on the noise distribution of the subimage and the size of the subimage because the peaks of object intensity and background intensity may be buried in noise signal if the subimage size is not large enough.

Finally, genetic algorithm is still a developing technique in digital image processing area. Although, this is the first time we combine mathematical morphology and GA the results obtained are very promising and is therefore worth studying further. Many improvements can be introduced into the basic proposed algorithm for enhancing the computation time of the algorithm and accuracy of the results.

APPENDIX. IMAGE SIGNAL TO NOISE RATIO MEASUREMENT

In the development of image enhancement, restoration, and coding techniques, it is useful to have some measure of the difference between a pair of similar images. The most common difference measure is mean square error. The mean-square error measure is popular because it correlates reasonably with subjective visual quality tests and it is mathematically tractable.

Consider a discrete image F(j, k) for j = 1,2,..., J and k = 1,2,..., K, which is regarded as a reference image, and consider a second image F'(j, k) of the same spatial dimensions as F(j, k) that is to be compared to the reference image. Under the assumption that F(j, k) and F'(j, k) represent samples of a stochastic process, the mean-square error between the image pairs is defined as

$$\xi_{MSE} = E\{|F(j,k) - F'(j,k)|^2\} \tag{1}$$

where E{.} is the expectation operator. The normalized mean square error is

$$\xi_{NMSE} = \frac{E\{|F(j,k) - F'(j,k)|^2\}}{E\{|F(j,k)|^2\}} \tag{2}$$

Error measures analogous to Equations 1 and 2 have been developed for deterministic image arrays. The least-squares error for a pair of deterministic arrays is defined as

$$\xi_{LSE} = \frac{1}{JK}\sum_{j=1}^{J}\sum_{k=1}^{K}|F(j,k) - F'(j,k)|^2 \tag{3}$$

and the normalized least-squares error is

$$\xi_{NLSE} = \frac{\sum_{j=1}^{J}\sum_{k=1}^{K} |F(j,k) - F'(j,k)|^2}{\sum_{j=1}^{J}\sum_{k=1}^{K} |F(j,k)|^2} \quad (4)$$

Another common form of error normalization is to divide Equation 3 by the squared peak value of x. This peak least-squares error measure is defined as

$$\xi_{PLSE} = \frac{\sum_{j=1}^{J}\sum_{k=1}^{K} |F(j,k) - F'(j,k)|^2}{[\max\{F(j,k)\}]^2} \quad (5)$$

In the literature the least-squares error expression of Equations 3 to 5 are sometimes called mean-square error measures even though they are computed from deterministic arrays. Image error measures are often expressed in terms of a signal-to-noise ratio (*SNR*) in decibel units, which is defined as

$$SNR = -10\log_{10}\{\xi\} \quad (6)$$

A common criticism of mean-square error and least-squares error measures is that they do not always correlate well with human subjective testing. In an attempt to improve this situation, a logical extension of the measurements is to substitute processed versions of the pair of images to be compared into the error expressions. The processing is chosen to map the original images into some perceptual space in which just noticeable differences are equally perceptible.

BIBLIOGRAPHY

[1] Gonzalez, R.C. and Woods, R.E., *Digital Image Processing*. Addison-Wesley, Reading, Mass., 1992.

[2] Haralick, R.M. and Shapiro, L.G., "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing,* vol. 29, pp.100-132, 1985.

[3] Pratt, W., *Digital Image Processing*. 2nd edition, John Wiley & Sons, New York, 1991.

[4] Horowitz, S.L. and Pavlidis, T, " Picture Segmentation by a Tree Transversal Algorithm," *JACM*, vol. 23, pp. 368-388, 1976.

[5] Rosenfeld, A. and Thurston, M., "Edge and Curve Detection for Visual Scene Analysis," *IEEE Trans. Computers*, vol. C-20, no. 5, pp. 562-569, May 1971.

[6] Zucker, S.W., Rosenfeld, A. and Davis, L.S., "Picture Segmentation by Texture Discrimination," *IEEE Trans. Computers*, vol. C-24, no. 12, pp. 1228-1233, December 1975.

[7] Bhanu,B., Lee, S. and Das, S., "Adaptive Image Segmentation Using Genetic and Hybrid Search Methods," *IEEE Trans. on Aerospace and Electronic System*, vol. AES-31, no. 4, pp. 1268-1290, October 1995.

[8] Serra, J., *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.

[9] Sternberg, S.R., "Grayscale Morphology," *Computer Vision, Graphics, Image Processing*, vol. 35, pp. 33-355, 1986.

[10] Baxes, G. A., *Digital Image Processing: principles and applications.* John Wiley & Sons, New York, 1994.

[11] Robert, S.R.S., Haralick, M. and Zhuang, X., "Image analysis using mathematical morphology," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no 4, pp. 532-550, 1987.

[12] Davis, L., *The handbook of Genetic Algorithms.* Van Nostrand Reingold, New York, 1991.

[13] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, Mass., 1989.

[14] Holland, J. H., *Adaptation in Natural and Artificial Systems.* University of Michigan press, Ann Arbor, 1975.

[15] Michalewicz, Z., *Genetic Algorithms + Data Structure = Evolution Programs.* AI Series. Springer-Verlag, New York, 1994.

[16] Joines, J. and Houck, C., "On the use of non-stationary penalty function to solve constrained optimization problems with genetic algorithms," *IEEE International Symposium Evolutionary Computation*, Orlando, FL, pp. 579-584, 1994