# Multi-Thresholding Image Segmentation Using Genetic Algorithm



Omar Banimelhem
Jordan University of Science and Technology

# Multi-Thresholding Image Segmentation Using Genetic Algorithm

**Omar Banimelhem**[1] **and Yahya Ahmed Yahya**[2]
[1]Department of Network Engineering and Security, Jordan University of Science and Technology, Irbid, Jordan
[2]Department of Computer Engineering, Jordan University of Science and Technology, Irbid, Jordan

**Abstract** - *Image segmentation is one of the essential problems in computer vision and image processing. It works by partitioning a digital image into multiple regions or sets. The increasing importance of image segmentation in multiple issues and applications has motivated the researchers to propose and improve algorithms that support image segmentation process. There are many methods for image segmentation. In this paper, we use thresholding technique with genetic algorithm to find optimal thresholds between the various objects and the background.*

**Keywords:** Image segmentation; Thresholding; Genetic algorithm.

## 1 Introduction

Image segmentation is the process of distinguishing the objects and background in an image. It is an essential preprocessing task for many applications that depend on computer vision such as medical imaging, locating objects in satellite images, machine vision, finger print and face recognition, agricultural imaging and other many applications. The accuracy of image segmentation stage would have a great impact on the effectiveness of subsequent stages of the image processing. Image segmentation problem has been studied by many researchers for several years; however, due to the characteristics of the images such as their different modal histograms, the problem of image segmentation is still an open research issue and so further investigation is needed.

Recently, many techniques have been proposed for image segmentation including graph-based algorithms, edge detection algorithms, and threshold-based algorithms [1-5]. For their simplicity, thresholding techniques have been given a considerable attention in the last few years.

On the other hand, and due to their capability in providing near optimal solutions for many practical applications, genetic algorithms have recently been employed in order to find solutions for the problem of image segmentations. In general, a Genetic Algorithm (GA) is a soft computational model that simulates the biological evolutionary process of natural selection [6].

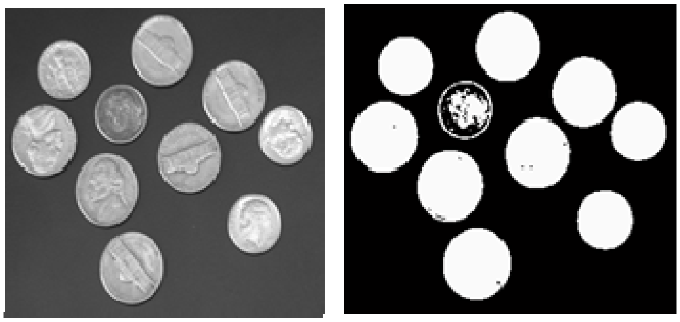
In this paper, image thresholding approach is employed with genetic algorithm to convert thresholding problem into an optimization problem by finding threshold(s). The proposed method aims to maximize the intra-class variance between object(s) and background and minimize the enter-class variance between background pixels among object pixels themselves [7]. Image segmentation quality is affected by the homogeneously between the gray level of pixels in each object. If object pixels gray levels vary, the results will probably be bad [7].

The rest of the paper is organized as follows. In Section 2, image thresholding approach is described in details. In Section 3, the concept of genetic algorithm is discussed. We discuss the proposed approach in Section 4. Section 5 presents the simulation results. Finally, the paper is concluded in Section 6.

## 2 Image Thresholding

Digital image can be viewed as two dimensional matrix or two variables function. It consists of discrete points called pixels. In color images, each pixel has three values: red, green and blue. Each value has a range between 0 and L-1, where L is the number of levels of precision. On the other hand, gray level images are composed of pixels where each pixel has only one value between 0 and L-1 called gray level. For many image processing problems, it would be simple and more efficient to deal with gray level images than with color images. For that reason, color images are often converted to gray level images before applying image processing algorithms. The most widely adopted gray level is 256 (i.e. the value of each pixel is between 0 and 255).

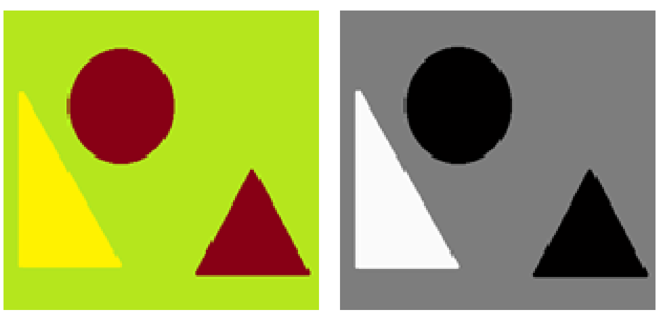
Image thresholding is an image segmentation method that works with gray level images. The idea is to find a threshold and if the pixel is below the threshold value, it is considered as a background, otherwise it is considered as part of an object. For example, the image in Figure 1-a has one object and background. The result of image segmentation is shown in Figure 1-b.



(a) (b)

**Figure 1**: Original image (a) and result of segmentation (b)

Threshold-based algorithms are divided into single-level and multi-level thresholding categories. Multi-thresholding approach generalizes the image thresholding by finding multiple thresholds which aim to separate multiple objects. For example, the image in Figure 2-a has three objects and the result of image segmentation is shown in Figure 2-b. In general, for segmenting an image that has *n* objects and background, *n* thresholds can be used.



(a) (b)

**Figure 2**: An image (a) show 3 objects and image (b) result of segmentation

To find the thresholds that best separate objects, it is easier to deal with statistics of the image instead of the image itself. Image histogram statistic is one dimensional representation that shows the frequency of each gray level in the image. It is computed by counting number of pixels that have the same gray level. Figure 3 shows the image histogram of the image showed in Figure1.

Converting color images into gray level images and then using the one dimensional histogram of the image makes thresholding-based segmentation process an easy and computationally efficient task, which can be used in many real time applications.

There are many methods to find thresholds from image histogram. In the case of one threshold, we can try all the values between 0 and L-1 and then we choose the value that gives the best segmentation to use it as a threshold.

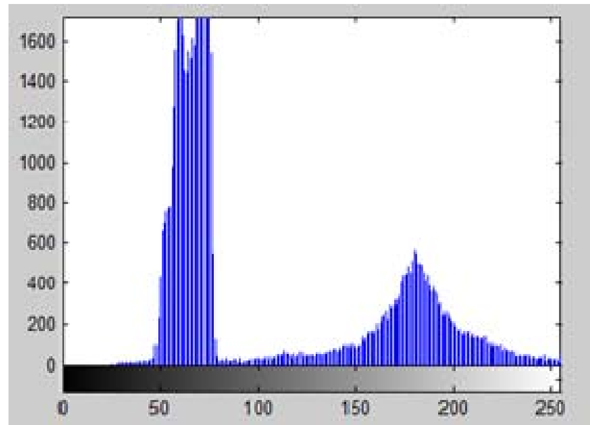



**Figure 3:** Image histogram of Figure1

However, for multiple thresholds segmentation, trying all the possible combinations needs $L\times(L-1)\times\ldots\times(L-t+1)$ trails, where *t* is the number of thresholds. It is obvious that the naïve method of trying all possible thresholds is not computationally efficient nor feasible for larger number of *t*. Indeed, the computational complexity of naïve approach suggest using more efficient algorithms for finding values of *t* thresholds.

Also it is critical to have a measure of goodness of thresholding to know if one set of thresholds perform valid thresholding or not. In natural images, one can see that same objects has similar pixels and different objects has unsimilar pixels, this suggest that the measure that will be taken into consideration is the inter-class variance and intra-class variance; that is the variance in gray level between the pixels belong to one object and variance between pixels in different objects.

The computational complexity of naïve approach, and the existence of goodness measure in the case of multiple thresholds motivated the use of an efficient search algorithm. In the next section, we show how to use genetic algorithm to achieve that.

# 3 Genetic Algorithms

Genetic algorithm (GA) is a natural inspired metaheuristic that mimics genes. A genetic algorithm is used to search for near optimal solutions when no deterministic method exist or if the deterministic method is computationally complex, GA is a population based algorithm (i.e. it generates multiple solutions each iteration). The number of solutions per iteration is called population size. Each solution is represented as chromosome and each chromosome is built up from genes.

For a genetic algorithm of population size *n*, it starts with *n* random solutions. Then it chooses the best member solutions for mating to generate new solutions. The best generated solutions will be added to the next iteration while the bad solutions will be rejected. While the algorithm iterates its solutions, these solutions are improved up to a point where a converge to near optimal solution is achieved.

Many factors should be taken into consideration when the genetic algorithm is used. The first factor is the representation of chromosome and genes because bad representation may result in slower convergence. Another important factor in the genetic algorithms is the mechanism of producing new solutions from the old ones. The most popular mechanisms are crossover and mutation. The third factor is how to find a fitness function (i.e. method to evaluate the solutions) in order to accept or reject the solutions, and how to select the best members for mating.

In general, a genetic algorithm has four stages: population initialization, evaluation of fitness, reproduction and termination.

Initialization is the process of creating initial random solutions, which can be done by setting genes to random values. In the initialization process, *n* chromosomes are created as the first generation of solutions. After the initialization, each chromosome fitness (i.e. solution goodness) is evaluated using the fitness function.

Reproduction process has four steps: selection, crossover, mutation, and accepting the solution.

- In the selection step, the fittest members in the current population are selected in order to reproduce new solutions. However, less fitness members will has also a chance to be selected. The selection step can be implemented by many mechanisms. One popular method is rollet wheel. This method works as follows: imagine a rollet wheel is partitioned into *n* sections where each section represents a chromosome. The area of each section is mapped according to chromosome fitness. Then a number between 0 and 360 is selected randomly and is mapped to the rollet. This gives more chance to fittest chromosomes and little but possible chance to less fitted members. In practice this can be done by evaluating each chromosome fitness and then normalize the sum to 1. This process converts the fitness measure into probability value, then a cumulative distribution is found and a random number from 0 to 1 is generated. The chromosome correspond to the random number in the cumulative is selected. This selection will be performed on two chromosomes to reproduce two new chromosomes each time.
- After selecting the chromosomes, a crossover operation is performed by selecting a random point in chromosomes and exchange genes after this point as shown in Figure 4.

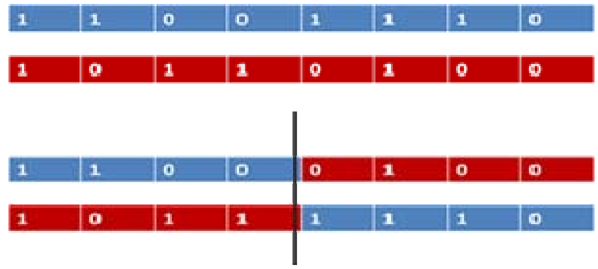



**Figure 4:** Crossover operation

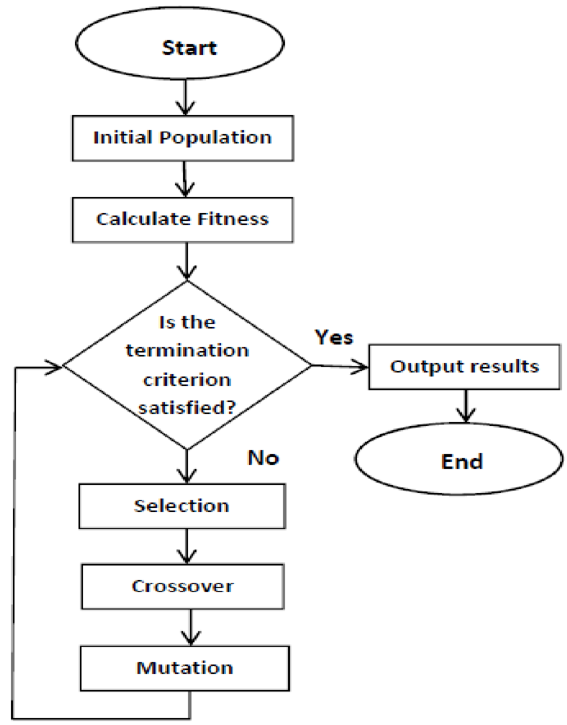



**Figure 5:** Flowchart for a general genetic algorithm [6]

- Crossover may stuck in local optima. To overcome this problem, a tie breaker is needed which can be achieved by using mutation operation where a gene is selected randomly and its value is changed. A widely used representation for genes is bits where each gene is represented by a bit. In this case, mutation is done by flipping a bit randomly in the chromosome.
- After crossover and mutation, two new chromosomes are reproduced. The final step is accepting these two chromosomes to be in the new population. Typically, the new chromosomes are accepted if they are better than their parents.

Termination is the last step in the genetic algorithms, Usually, the iteration of the genetic algorithm is stopped when a certain criteria is met. The most widely used stopping criteria is the number of iterations. When a predefined number of iterations is satisfied, the genetic algorithm is terminated. A general genetic algorithm is showed in Figure 5 [6].

# 4 The Proposed Algorithm

The proposed algorithm uses a genetic algorithm to find near optimal thresholds. We aim at maximizing the intra-class variance and minimizing the inter-class variance. The proposed algorithm uses chromosomes which are encoded as a vector of bits. Each vector has $L*n$ bits where $L$ is the log(number of gray levels) and $n$ is the number of thresholds used. Each $L$ bits in the chromosome represents a threshold value. The chromosome structure is shown in Figure 6.

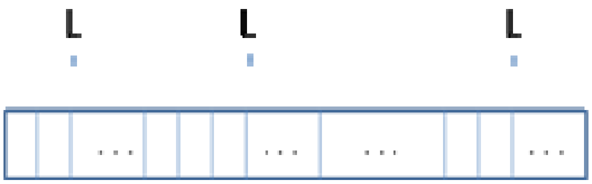



**Figure 6:** Chromosome structure

First of all, the histogram of an image is found. The histogram gives information that will be used for evaluating fitness. Then, $K$ initial chromosomes are generated randomly. After that, the genetic algorithm iterates for fixed number of iteration and finally the best chromosome will be selected as the solution. Figure 7 shows the pseudo code for the proposed algorithm

The Proposed Image Segmentation Algorithm
**Inputs**: image Im, population size, crossover rate, mutation rate, number of iterations, number of thresholds
**Outputs**: segmented image

---

```
1. Im ← Read image
2. Hist ← image histogram (Im)
3. Pop ← initial K chromosomes
4. For i =1 to number of iterations
5.      While(size(Pop) > size(NewPop))
6.      Ch1,Ch2 ← Select two chromosoms by rolletwheel
7.         If (rand < crossover rate) then
8.            Chnew1, Chnew2← crossover (Ch1,Ch2)
9.         If (rand < mutation rate) then
10.              Chnew1 ← mutate (Chnew1)
11.           If (rand< mutation rate) then
12.              Chnew2 ← mutate (Chnew2)
        If(fit(Chnew1) >max(fit(Ch1),fit(CH2) )) then
           NewPop.Add(Chnew1)
13.           If(fit(Chnew2) >max(fit(Ch1),fit(CH2) )) then
14.              NewPop.Add(Chnew2)
```

---

**Figure 7:** The proposed Image Segmentation Algorithm

The fitness function measures the goodness of the segmentation. Normally, in natural images the variance between pixels from different objects is large while the variance between pixels within the same object is small. This affects the design of fitness function. In the proposed algorithm, we used the following function, $F$, as a measure of the fitness of threshold $i$ [7]:

$$F(\text{fitness}, i) = S_{\textit{Between objects}} / S_{\textit{within objects}}$$

Where
The variance between objects, $S_{\textit{Between objects}}$, is given as:

$$S_{\textit{Between}\ \text{objects}} = \sum_i P_i (m_i - m_g)^2$$

And the variance within the objects is given as

$$S_{\textit{Within}\ \text{objects}} = \sum_i \frac{S_i}{S_g}$$

Where

$mi$ is the mean of pixels in the segment whose threshold value is $thsld\ i$ , $mi$ is given as:

$$m_i = \sum_{thsld\ i}^{thsld\ i+1} x \times hist(x)$$

the probability of segment i, $Pi$, is given as:

$$p_i = \sum_{thsld\ i}^{thsld\ i+1} hist(x)$$

and the global mean of the image is given as:

$$mg = mi \times pi$$

the variance in segment $i$ is given as:

$$S_i = \sum_{thsld\ i}^{thsld\ i+1} (hist(x) - m_i)^2$$

And the global variance of the image is given as:

$$S_g = \sum_X hist(x) \times (X - m_g)^2$$

# 5 Implementation and Results

The algorithm is implemented in Matlab. The initial population size is ten chromosomes. The crossover and mutation rates are 0,95 and 0.05, respectively. The algorithm iterates for 500 iterations. It was run on 2.27 GHz core i3 processor.

The algorithm showed the ability to segment image with high quality results. A set of samples of input images and segmented results are shown in Figures 8 through 11.

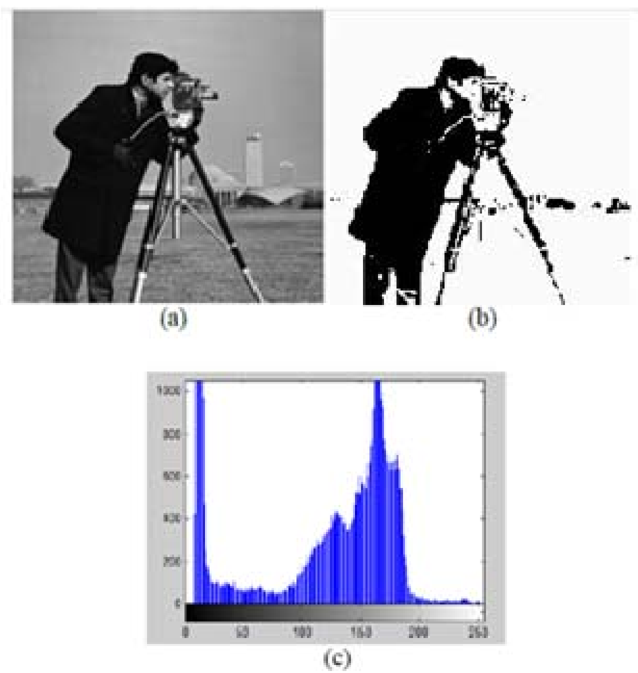


**Figure 8:** (a) original image of cameraman (b) result after one segmentation (c) histogram image and threshold at 83 levels

The simulation time and the threshould values of the four images are given in Table 1.

**Table 1:** Simulation time and thresholds values for the four images

| Image Name | Simulation Time (sec) | Threshold Value |
|---|---|---|
| Rice Image | 4.312212 | 130 |
| Camera man | 4.191182 | 83 |
| Liftingbody | 4.562393 | 119-191 |
| Fishman | 12.931214 | 111-168 |

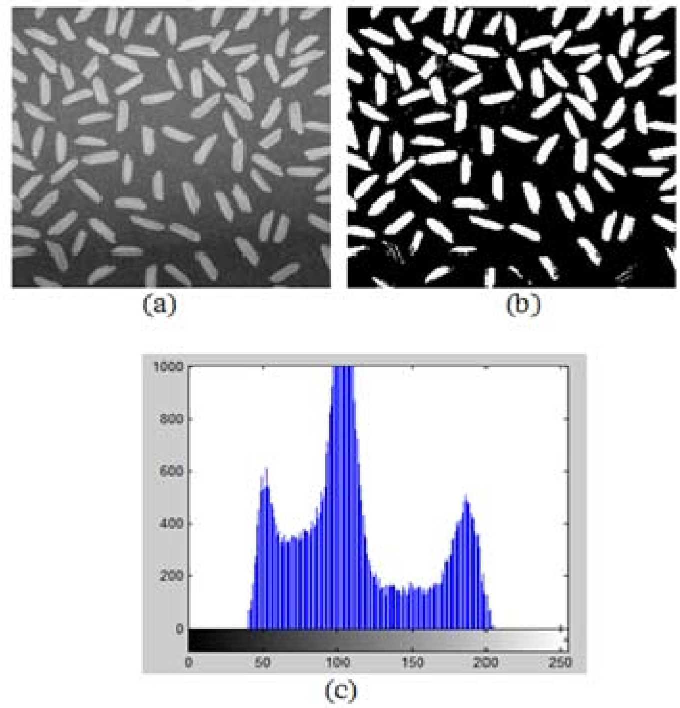


**Figure 9:** (a) original image of Rice (b) result after one segmentation (c) histogram image and threshold at 130 levels

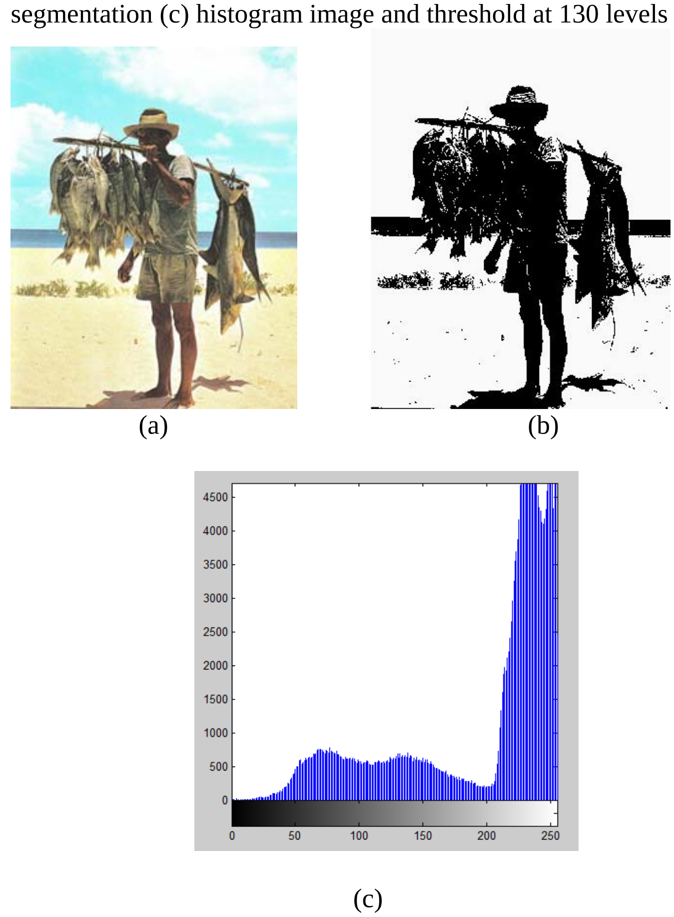
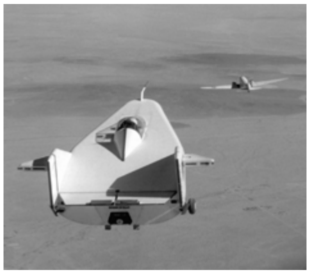
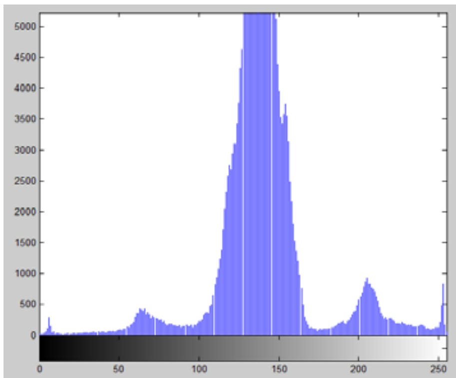


**Figure 10:** (a) original image of fishman (b) result after two segmentations (c) histogram image and threshold at 111 and 168 levels



(a)

(b)




(c)

**Figure 11:** (a) original image of liftingbody (b) result after two segmentations (c) histogram image and threshold at 119 and 191 levels

# 6 Conclusions

In this paper, we proposed a new method of multi-thresholding image segmentation using genetic algorithm. The chromosomes are constructed as vectors of bits that represent the genes where each vectors is modeled by *t* levels and each level is represented by *Log L* bits. In the future work, other fitness functions will be developed.

# 7 References

[1] P.F. Felzenszwalb and D.P. Huttenlocher, " Efficient graph-based image segmentation," Int. J. of Computer. Vision., 59(2), 2004, pp. 167–181

[2] S.Lakshmi and V.Sankaranarayanan," A study of Edge Detection Techniques for Segmentation Computing Approaches," IJCA Special Issue on “Computer Aided Soft Computing Techniques for Imaging and Biomedical Applications” , CASCT, 2010, pp. 35-40.

[3] Sahoo, P. K., S. Soltani and A. K. C. Wang.," A survey of thresholding techniques," Computer Vision, Graphics, and Image Processing,1988, 41(2), pp. 233-260.

[4] N. Senthilkumaran and R. Rajesh, “Edge Detection Techniques for Image Segmentation - A Survey”, Proceedings of the International Conference on Managing Next Generation Software Applications (MNGSA-08), 2008, pp.749-760.

[5] S. Chen, Y. Lin, Y. Liu and Z.Lu, " Image Segmentation by Multi-threshold based on Fisher Function and Histogram Algorithm ", International Conference on Computer Application and System Modeling, 2010

[6] Goldberg, E. David. Genetic Algorithms in Search Optimization and Machine Learning. Addison Wesley. pp. 41, 1989.

[7] R. Gonzalez and R. Woods, "Digital image processing", third edition, Pearson Prentice Hall, 2008, pp. 738-761.