

Research Article

Multilevel Thresholding Segmentation Based on Harmony Search Optimization

Diego Oliva,¹ Erik Cuevas,² Gonzalo Pajares,¹ Daniel Zaldivar,² and Marco Perez-Cisneros²

¹ *Departamento de Ingeniería del Software e Inteligencia Artificial, Facultad Informática, Universidad Complutense, 28040 Madrid, Spain*

² *Departamento de Ciencias Computacionales, Universidad de Guadalajara, CUCEI, CU-TONALA, Avenida Revolución 1500, C.P 44430, Guadalajara, JAL, Mexico*

Correspondence should be addressed to Erik Cuevas; erik.cuevas@cucei.udg.mx

Received 14 June 2013; Revised 17 August 2013; Accepted 20 August 2013

Academic Editor: Zong Woo Geem

Copyright © 2013 Diego Oliva et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a multilevel thresholding (MT) algorithm based on the harmony search algorithm (HSA) is introduced. HSA is an evolutionary method which is inspired in musicians improvising new harmonies while playing. Different to other evolutionary algorithms, HSA exhibits interesting search capabilities still keeping a low computational overhead. The proposed algorithm encodes random samples from a feasible search space inside the image histogram as candidate solutions, whereas their quality is evaluated considering the objective functions that are employed by the Otsu's or Kapur's methods. Guided by these objective values, the set of candidate solutions are evolved through the HSA operators until an optimal solution is found. Experimental results demonstrate the high performance of the proposed method for the segmentation of digital images.

1. Introduction

Segmentation is one of the most important tasks in image processing that endeavors to identify whether a pixel intensity corresponds to a predefined class. Thresholding is the easiest method for segmentation as it works taking a threshold (th) value so that pixels whose intensity value is higher than th are labeled as first class while the rest correspond to a second class label. When the image is segmented into two classes, the task is called bilevel thresholding (BT) and requires only one th value. On the other hand, when pixels are separated into more than two classes, the task is named as MT and demands more than one th value [1].

In recent years image processing has been applied to different areas as engineering, medicine, agriculture, and so forth. Since most of such implementations use a TH methodology, several techniques had been studied. Generally, TH methods are divided into parametric and nonparametric [2–5]. Parametric approaches need to estimate values of a probability density function to model each class. The estimation

process is time consuming and computationally expensive. On the other hand, the TH nonparametric employs several criteria such as the between-class variance, the entropy, and the error rate [6–8] in order to verify the quality of a th value. These metrics could also be used as optimization functions since they result as an attractive option due their robustness and accuracy.

There exist two classical thresholding methods. The first, proposed by Otsu in [6] that maximizes the variance between classes while the second method, submitted by Kapur et al. in [7], uses the maximization of the entropy to measure the homogeneity among classes. Their efficiency and accuracy have been already proved for a bilevel segmentation [4]. Although they can be expanded for MT, their computational complexity increases exponentially when a new threshold is incorporated [5].

As an alternative to classical methods, the MT problem has also been handled through evolutionary optimization methods. In general, they have demonstrated to deliver better results than those based on classical techniques in terms

of accuracy, speed, and robustness. Numerous evolutionary approaches have been reported in the literature.

Hammouche et al. in [9] provides an interesting survey of how different evolutionary algorithms are used to solve the Kaptur's and Otsu's problems. The study uses four classical evolutionary algorithms to test their efficiency in MT. Such methods include differential evolution (DE) [10], simulated annealing (SA) [11], and tabu search (TS) [12].

Genetic algorithms (GA) [13], inspired on the biological evolution, have been also used for solving segmentation tasks. One interesting example is presented in [14], where a GA-based algorithm is combined with Gaussian models for multilevel thresholding. Other similar works, such as that of Yin [15] proposes an improved GA for optimal thresholding. In the approach, it is used as a learning strategy to increase the speed of convergence.

Evolutionary approaches inspired on swarm behavior, such as particle swarm optimization (PSO) [16] and artificial bee colony (ABC) [17], have been employed to face the segmentation problem. In [18], both methods are used to find the optimal multilevel threshold points by using the Kapur's entropy as fitness function.

Finally, in [19], the optimal segmentation threshold values are determined by using the bacterial foraging algorithm (BFA). Such method aims to maximize the Kapur's and Otsu's objective functions by considering a set of operators that are based on the social foraging behavior of the bacteria *Escherichia Colli*.

On the other hand, the harmony search algorithm (HSA) introduced by Geem et al. [20] is an evolutionary optimization algorithm which is based on the metaphor of the improvisation process that occurs when a musician searches for a better state of harmony. The HSA generates a new candidate solution from all existing solutions. The solution vector is analogous to the harmony in music while the local and global search schemes are analogous to musician's improvisations. In comparison to other metaheuristics methods in the literature, HSA imposes fewer mathematical requirements as it can be easily adapted for solving several sorts of engineering optimization challenges [21]. Furthermore, numerical comparisons have demonstrated that the convergence for the HSA is faster than GA [22] which attracts further attention. It has been successfully applied to solve a wide range of practical optimization problems such as discrete and continuous structural optimization [23], parameter estimation of the nonlinear Muskingum model [24], design optimization of water distribution networks [25], vehicle routing [26], combined heat and power economic dispatch [27], design of steel frames [28], and image processing [29]. Although the standard HSA presents good search characteristics, several modifications to the original HSA have been proposed in the literature in order to enhance its own features [30].

In this paper, a novel multithresholding segmentation algorithm is introduced. The proposed approach, called the harmony search multithresholding algorithm (HSMA), combines the original harmony search algorithm (HSA) and the Otsu's and Kapur's methodologies. The proposed algorithm takes random samples from a feasible search space

inside the image histogram. Such samples build each harmony (candidate solution) in the HSA context, whereas its quality is evaluated considering the objective function that is employed by the Otsu's or the Kapur's method. Guided by these objective values, the set of candidate solutions are evolved using the HSA operators until the optimal solution is found. The approach generates a multilevel segmentation algorithm which can effectively identify the threshold values of a digital image within a reduced number of iterations. Experimental results over several complex images have validated the efficiency of the proposed technique regarding accuracy, speed, and robustness.

The paper is organized as follows. In Section 2, the HSA is introduced. Section 3 gives a brief description of the Otsu's and Kapur's methods. Section 4 explains the implementation of the proposed algorithm. Section 5 discusses experimental results after testing the proposed method over a set benchmark images. Finally, some conclusions are discussed in Section 6.

2. Harmony Search Algorithm

2.1. The Harmony Search Algorithm. In the basic HSA, each solution is called a "harmony" and is represented by an n -dimension real vector. An initial population of harmony vectors are randomly generated and stored within a harmony memory (HM). A new candidate harmony is thus generated from the elements in the HM by using a memory consideration operation either by a random reinitialization or a pitch adjustment operation. Finally, the HM is updated by comparing the new candidate harmony and the worst harmony vector in the HM. The worst harmony vector is replaced by the new candidate vector when the latter delivers a better solution in the HM. The above process is repeated until a certain termination criterion is met. The basic HS algorithm consists of three main phases: HM initialization, improvisation of new harmony vectors, and updating the HM. The following discussion addresses details about each stage.

2.1.1. Initializing the Problem and the Algorithm Parameters. In general, the global optimization problem can be summarized as follows:

$$\begin{aligned} &\text{minimize } f(\mathbf{x}), \quad \mathbf{x} = (x(1), x(2), \dots, x(n)) \in \mathbf{R}^n, \\ &\text{subject to: } x(j) \in [l(j), u(j)] \quad j = 1, 2, \dots, n, \end{aligned} \quad (1)$$

where $f(\mathbf{x})$ is the objective function, $\mathbf{x} = (x(1), x(2), \dots, x(n))$ is the set of design variables, n is the number of design variables, and $l(j)$ and $u(j)$ are the lower and upper bounds for the design variable $x(j)$, respectively. The parameters for HSA are the harmony memory size, that is, the number of solution vectors lying on the harmony memory (HM), the harmony-memory consideration rate (HMCR), the pitch adjusting rate (PAR), the distance bandwidth (BW), and the number of improvisations (NI) which represents the total number of iterations. The performance of HSA is strongly influenced by values assigned to such parameters, which in turn, depend on the application domain [31].

2.1.2. Harmony Memory Initialization. In this stage, initial vector components at HM, that is, HMS vectors, are configured. Let $\mathbf{x}_i = \{x_i(1), x_i(2), \dots, x_i(n)\}$ represent the i th randomly-generated harmony vector: $x_i(j) = l(j) + (u(j) - l(j)) \cdot \text{rand}(0, 1)$ for $j = 1, 2, \dots, n$ and $i = 1, 2, \dots, \text{HMS}$, where $\text{rand}(0, 1)$ is a uniform random number between 0 and 1, the upper and lower bounds of the search space are defined by $l(j)$ and $u(j)$, respectively. Then, the HM matrix is filled with the HMS harmony vectors as follows:

$$\text{HM} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{\text{HMS}} \end{bmatrix}. \quad (2)$$

2.1.3. Improvisation of New Harmony Vectors. In this phase, a new harmony vector \mathbf{x}_{new} is built by applying the following three operators: memory consideration, random reinitialization, and pitch adjustment. Generating a new harmony is known as ‘‘improvisation.’’ In the memory consideration step, the value of the first decision variable $x_{\text{new}}(1)$ for the new vector is chosen randomly from any of the values already existing in the current HM, that is, from the set $\{x_1(1), x_2(1), \dots, x_{\text{HMS}}(1)\}$. For this operation, a uniform random number r_1 is generated within the range $[0, 1]$. If r_1 is less than HMCR, the decision variable $x_{\text{new}}(1)$ is generated through memory considerations; otherwise, $x_{\text{new}}(1)$ is obtained from a random reinitialization between the search bounds $[l(1), u(1)]$. Values of the other decision variables $x_{\text{new}}(2), x_{\text{new}}(3), \dots, x_{\text{new}}(n)$ are also chosen accordingly. Therefore, both operations, memory consideration and random reinitialization, can be modeled as follows:

$$x_{\text{new}}(j) = \begin{cases} x_i(j) \in \{x_1(j), x_2(j), \dots, x_{\text{HMS}}(j)\}, \\ \quad \text{with probability HMCR}, \\ l(j) + (u(j) - l(j)) \cdot \text{rand}(0, 1), \\ \quad \text{with probability } 1 - \text{HMCR}. \end{cases} \quad (3)$$

Every component obtained by memory consideration is further examined to determine whether it should be pitch-adjusted. For this operation, the pitch-adjusting rate (PAR) is defined as to assign the frequency of the adjustment and the bandwidth factor (BW) to control the local search around the selected elements of the HM. Hence, the pitch-adjusting decision is calculated as follows:

$$x_{\text{new}}(j) = \begin{cases} x_{\text{new}}(j) = x_{\text{new}}(j) \pm \text{rand}(0, 1) \cdot \text{BW}, \\ \quad \text{with probability PAR}, \\ x_{\text{new}}(j), \quad \text{with probability } (1 - \text{PAR}). \end{cases} \quad (4)$$

Pitch adjusting is responsible for generating new potential harmonies by slightly modifying original variable positions. Such operation can be considered similar to the mutation process in evolutionary algorithms. Therefore, the decision

variable is either perturbed by a random number between 0 and BW or left unaltered. In order to protect the pitch adjusting operation, it is important to assure that points lying outside the feasible range $[l, u]$ must be reassigned, that is, truncated to the maximum or minimum values of the interval.

2.1.4. Updating the Harmony Memory. After a new harmony vector x_{new} is generated, the harmony memory is updated by the survival of the fit competition between x_{new} and the worst harmony vector x_w in the HM. Therefore x_{new} will replace x_w and become a new member of the HM in case the fitness value of x_{new} is better than the fitness value of x_w .

2.1.5. Computational Procedure. The computational procedure of the basic HSA can be summarized as in Algorithm 1 [20].

This procedure is implemented for minimization. If the intention is to maximize the objective function, a sign modification of step 4 ($\mathbf{x}_w = \mathbf{x}_{\text{new}}$ if $f(\mathbf{x}_{\text{new}}) > f(\mathbf{x}_w)$) is required. In this paper the HSA is used for maximization proposes.

3. Image Multilevel Thresholding (MT)

Thresholding is a process in which the pixels of a gray scale image are divided in sets or classes depending on their intensity level (L). For this classification it is necessary to select a threshold value (th) and follow the simple rule of

$$\begin{aligned} C_1 &\leftarrow p \quad \text{if } 0 \leq p < \text{th}, \\ C_2 &\leftarrow p \quad \text{if } \text{th} \leq p < L - 1, \end{aligned} \quad (5)$$

where p is one of the $m \times n$ pixels of the gray scale image I_g that can be represented in L gray scale levels $L = \{0, 1, 2, \dots, L-1\}$. C_1 and C_2 are the classes in which the pixel p can be located, while th is the threshold. The rule in (5) corresponds to a bilevel thresholding and can be easily extended for multiple sets:

$$\begin{aligned} C_1 &\leftarrow p \quad \text{if } 0 \leq p < \text{th}_1, \\ C_2 &\leftarrow p \quad \text{if } \text{th}_1 \leq p < \text{th}_2, \\ C_i &\leftarrow p \quad \text{if } \text{th}_i \leq p < \text{th}_{i+1}, \\ C_n &\leftarrow p \quad \text{if } \text{th}_n \leq p < L - 1, \end{aligned} \quad (6)$$

where $\{\text{th}_1, \text{th}_2, \dots, \text{th}_i, \text{th}_{i+1}, \text{th}_k\}$ represent different thresholds. The problem for both bilevel and MT is to select the th values that correctly identify the classes. Although, Otsu’s and Kapur’s methods are well-known approaches for determining such values, both propose a different objective function which must be maximized in order to find optimal threshold values, just as it is discussed below.

3.1. Between-Class Variance (Otsu’s Method). This is a non-parametric technique for thresholding proposed by Otsu [6] that employs the maximum variance value of the different

Step 1. Set the parameters HMS, HMCR, PAR, BW and NI.
Step 2. Initialize the HM and calculate the objective function value of each harmony vector.
Step 3. Improvise a new harmony \mathbf{x}_{new} as follows:
 for ($j = 1$ to n) do
 if ($r_1 < \text{HMCR}$) then
 $x_{\text{new}}(j) = x_a(j)$ where $a \in (1, 2, \dots, \text{HMS})$
 if ($r_2 < \text{PAR}$) then
 $x_{\text{new}}(j) = x_{\text{new}}(j) \pm r_3 \cdot \text{BW}$ where $r_1, r_2, r_3 \in \text{rand}(0, 1)$
 end if
 if $x_{\text{new}}(j) < l(j)$
 $x_{\text{new}}(j) = l(j)$
 end if
 if $x_{\text{new}}(j) > u(j)$
 $x_{\text{new}}(j) = u(j)$
 end if
 else
 $x_{\text{new}}(j) = l(j) + r \cdot (u(j) - l(j))$, where $r \in \text{rand}(0, 1)$
 end if
 end for
Step 4. Update the HM as $\mathbf{x}_w = \mathbf{x}_{\text{new}}$ if $f(\mathbf{x}_{\text{new}}) < f(\mathbf{x}_w)$
Step 5. If NI is completed, the best harmony vector \mathbf{x}_b in the HM is returned; otherwise go back to Step3.

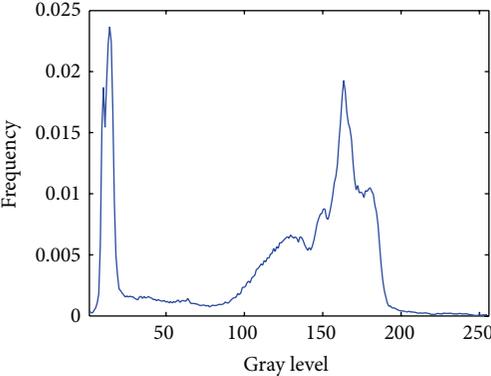
ALGORITHM 1

Step 1. Read the image I and if it is RGB separate it into I_R, I_G and I_B . If I is gray scale store it into I_{Gr} . $c = 1, 2, 3$ for RGB images or $c = 1$ for gray scale images.
Step 2. Obtain histograms: for RGB images h^R, h^G, h^B and for gray scale images h^{Gr} .
Step 3. Calculate the probability distribution using (7) and obtain the histograms.
Step 4. Initialize the HSA parameters: HMS, k , HMCR, PAR, BW, NI, and the limits l and u .
Step 5. Initialize a HM \mathbf{x}_i^c of HMS random particles with k dimensions.
Step 6. Compute the values ω_i^c and μ_i^c . Evaluate each element of **HM** in the objective function $J(\mathbf{HM})$ (14) or (20) depending on the thresholding method (Otsu or Kapur respectively).
Step 7. Improvise a new harmony $\mathbf{x}_{\text{new}}^c$ as follows:
 for ($j = 1$ to n) do
 if ($r_1 < \text{HMCR}$) then
 $x_{\text{new}}^c(j) = x_a^c(j)$ where $a \in (1, 2, \dots, \text{HMS})$
 if ($r_2 < \text{PAR}$) then
 $x_{\text{new}}^c(j) = x_a^c(j) \pm r_3 \cdot \text{BW}$ where $r_1, r_2, r_3 \in \text{rand}(0, 1)$
 end if
 if $x_{\text{new}}^c(j) < l(j)$
 $x_{\text{new}}^c(j) = l(j)$
 end if
 if $x_{\text{new}}^c(j) > u(j)$
 $x_{\text{new}}^c(j) = u(j)$
 end if
 else
 $x_{\text{new}}^c(j) = l(j) + r \cdot (u(j) - l(j))$ where $r \in \text{rand}(0, 1)$
 end if
 end for
Step 8. Update the HM as $\mathbf{x}_{\text{worst}}^c = \mathbf{x}_{\text{new}}^c$ if $f(\mathbf{x}_{\text{new}}^c) > f(\mathbf{x}_{\text{worst}}^c)$
Step 9. If NI is completed or the stop criteria is satisfied, then jump to Step 10; otherwise go back to Step 6.
Step 10. Select the harmony that has the best x_{best}^c objective function value.
Step 11. Apply the thresholds values contained in x_{best}^c to the image I (6).

ALGORITHM 2



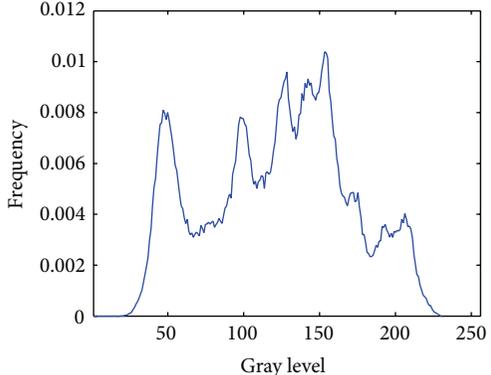
(a)



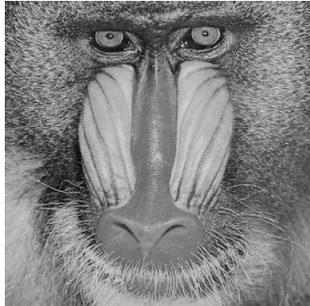
(b)



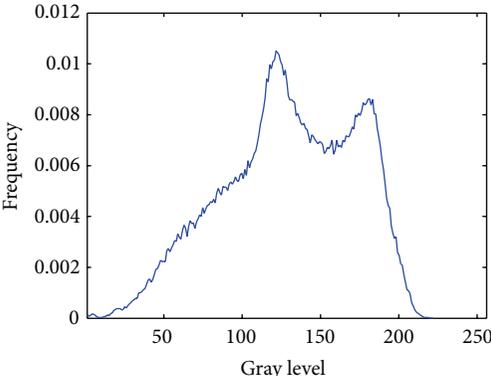
(c)



(d)



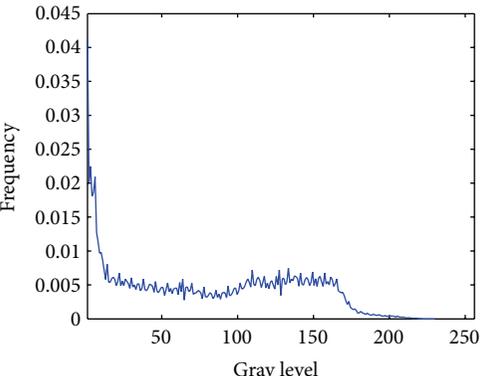
(e)



(f)



(g)



(h)

FIGURE 1: Continued.

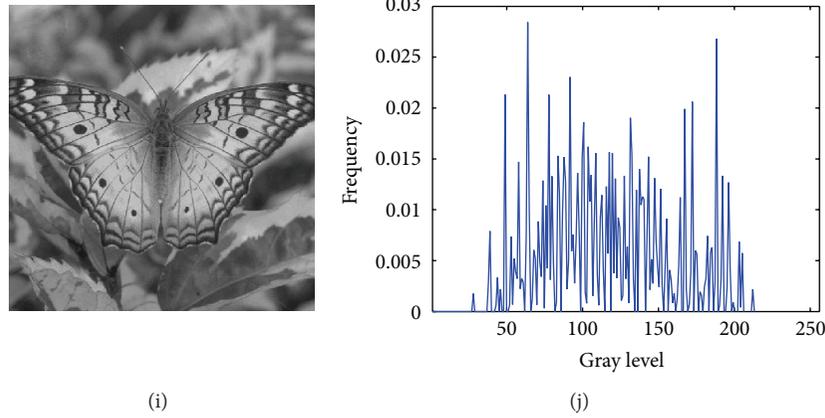


FIGURE 1: (a) Camera man, (c) Lena, (e) Baboon, (g) Hunter, and (i) Butterfly, the selected benchmark images. (b), (d), (f), (h), and (j) histograms of the images.

classes as a criterion to segment the image. Taking the L intensity levels from a gray scale image or from each component of a RGB (red, green, and blue) image, the probability distribution of the intensity values is computed as follows:

$$\text{Ph}_i^c = \frac{h_i^c}{\text{NP}}, \quad \sum_{i=1}^{\text{NP}} \text{Ph}_i^c = 1, \quad (7)$$

$$c = \begin{cases} 1, 2, 3, & \text{if RGB Image,} \\ 1, & \text{if Gray scale Image,} \end{cases}$$

where i is a specific intensity level ($0 \leq i \leq L - 1$), c is the component of the image which depends if the image is gray scale or RGB, whereas NP is the total number of pixels in the image. h_i^c (histogram) is the number of pixels that corresponds to the i intensity level in c . The histogram is normalized within a probability distribution Ph_i^c . For the simplest segmentation (bilevel) two classes are defined as

$$C_1 = \frac{\text{Ph}_1^c}{\omega_0^c(\text{th})}, \dots, \frac{\text{Ph}_{\text{th}}^c}{\omega_0^c(\text{th})}, \quad C_2 = \frac{\text{Ph}_{\text{th}+1}^c}{\omega_1^c(\text{th})}, \dots, \frac{\text{Ph}_L^c}{\omega_1^c(\text{th})}, \quad (8)$$

where $\omega_0(\text{th})$ and $\omega_1(\text{th})$ are probabilities distributions for C_1 and C_2 , as it is shown by

$$\omega_0^c(\text{th}) = \sum_{i=1}^{\text{th}} \text{Ph}_i^c, \quad \omega_1^c(\text{th}) = \sum_{i=\text{th}+1}^L \text{Ph}_i^c. \quad (9)$$

It is necessary to compute mean levels μ_0^c and μ_1^c that define the classes using (10). Once those values are calculated, the Otsu variance between classes σ^{2^c} is calculated using (11) as follows:

$$\mu_0^c = \sum_{i=1}^{\text{th}} \frac{i \text{Ph}_i^c}{\omega_0^c(\text{th})}, \quad \mu_1^c = \sum_{i=\text{th}+1}^L \frac{i \text{Ph}_i^c}{\omega_1^c(\text{th})}, \quad (10)$$

$$\sigma^{2^c} = \sigma_1^c + \sigma_2^c, \quad (11)$$

Notice that for both equations, (10) and (11), c depends on the type of image. In (11) the number two is part of the Otsu's variance operator and does not represent an exponent in the mathematical sense. Moreover σ_1^c and σ_2^c in (11) are the variances of C_1 and C_2 which are defined as

$$\sigma_1^c = \omega_0^c (\mu_0^c + \mu_T^c)^2, \quad \sigma_2^c = \omega_1^c (\mu_1^c + \mu_T^c)^2, \quad (12)$$

where $\mu_T^c = \omega_0^c \mu_0^c + \omega_1^c \mu_1^c$ and $\omega_0^c + \omega_1^c = 1$. Based on the values σ_1^c and σ_2^c , (13) presents the objective function:

$$J(\text{th}) = \max(\sigma^{2^c}(\text{th})), \quad 0 \leq \text{th} \leq L - 1, \quad (13)$$

where $\sigma^{2^c}(\text{th})$ is the Otsu's variance for a given th value. Therefore, the optimization problem is reduced to find the intensity level (th) that maximizes (13).

Otsu's method is applied for a single component of an image. In case of RGB images, it is necessary to apply separation into single component images. The previous description of such bilevel method can be extended for the identification of multiple thresholds. Considering k thresholds, it is possible separate the original image into k classes using (6); then it is necessary to compute the k variances and their respective elements. The objective function $J(\text{th})$ in (13) can thus be rewritten for multiple thresholds as follows:

$$J(\mathbf{TH}) = \max(\sigma^{2^c}(\mathbf{TH})), \quad 0 \leq \text{th}_i \leq L - 1, \quad i = 1, 2, \dots, k, \quad (14)$$

where $\mathbf{TH} = [\text{th}_1, \text{th}_2, \dots, \text{th}_{k-1}]$, is a vector containing multiple thresholds and the variances are computed through

$$\sigma^{2^c} = \sum_{i=1}^k \sigma_i^c = \sum_{i=1}^k \omega_i^c (\mu_i^c - \mu_T^c)^2. \quad (15)$$

Here, i represents the i class, ω_i^c and μ_j^c are, respectively, the probability of occurrence and the mean of a class. In MT, such values are obtained as

$$\begin{aligned}\omega_0^c(\text{th}) &= \sum_{i=1}^{\text{th}_1} \text{Ph}_i^c, \\ \omega_1^c(\text{th}) &= \sum_{i=\text{th}_1+1}^{\text{th}_2} \text{Ph}_i^c, \\ &\vdots \\ \omega_{k-1}^c(\text{th}) &= \sum_{i=\text{th}_{k-1}+1}^L \text{Ph}_i^c.\end{aligned}\quad (16)$$

And, for the mean values

$$\begin{aligned}\mu_0^c &= \sum_{i=1}^{\text{th}_1} \frac{i \text{Ph}_i^c}{\omega_0^c(\text{th}_1)}, \\ \mu_1^c &= \sum_{i=\text{th}_1+1}^{\text{th}_2} \frac{i \text{Ph}_i^c}{\omega_0^c(\text{th}_2)}, \\ &\vdots \\ \mu_{k-1}^c &= \sum_{i=\text{th}_{k-1}+1}^L \frac{i \text{Ph}_i^c}{\omega_0^c(\text{th}_k)}.\end{aligned}\quad (17)$$

Similar to the bilevel case, for the MT using the Otsu's method, c corresponds to the image components, RGB $c = 1, 2, 3$, and gray scale $c = 1$.

3.2. Entropy Criterion Method (Kapur's Method). Another nonparametric method that is used to determine the optimal threshold values has been proposed by Kapur et al. [7]. It is based on the entropy and the probability distribution of the image histogram. The method aims to find the optimal th that maximizes the overall entropy. The entropy of an image measures the compactness and separability among classes. In this sense, when the optimal th value appropriately separates the classes, the entropy has the maximum value. For the bilevel example, the objective function of the Kapur's problem can be defined as

$$\begin{aligned}J(\text{th}) &= H_1^c + H_2^c, \\ c &= \begin{cases} 1, 2, 3, & \text{if RGB Image,} \\ 1, & \text{if Gray scale Image,} \end{cases}\end{aligned}\quad (18)$$

where the entropies H_1 and H_2 are computed by the following model:

$$\begin{aligned}H_1^c &= \sum_{i=1}^{\text{th}} \frac{\text{Ph}_i^c}{\omega_0^c} \ln \left(\frac{\text{Ph}_i^c}{\omega_0^c} \right), \\ H_2^c &= \sum_{i=\text{th}+1}^L \frac{\text{Ph}_i^c}{\omega_1^c} \ln \left(\frac{\text{Ph}_i^c}{\omega_1^c} \right).\end{aligned}\quad (19)$$

Ph_i^c is the probability distribution of the intensity levels which is obtained using (7). $\omega_0(\text{th})$ and $\omega_1(\text{th})$ are probabilities distributions for C_1 and C_2 . $\ln(\cdot)$ stands for the natural logarithm. Similar to the Otsu's method, the entropy-based approach can be extended for multiple threshold values; for such a case, it is necessary to divide the image into k classes using the similar number of thresholds. Under such conditions, the new objective function is defined as:

$$\begin{aligned}J(\mathbf{TH}) &= \max \left(\sum_{i=1}^k H_i^c \right), \\ c &= \begin{cases} 1, 2, 3, & \text{if RGB Image,} \\ 1, & \text{if Gray scale Image,} \end{cases}\end{aligned}\quad (20)$$

where $\mathbf{TH} = [\text{th}_1, \text{th}_2, \dots, \text{th}_{k-1}]$ is a vector that contains the multiple thresholds. Each entropy is computed separately with its respective th value, so (21) is expanded for k entropies:

$$\begin{aligned}H_1^c &= \sum_{i=1}^{\text{th}_1} \frac{\text{Ph}_i^c}{\omega_0^c} \ln \left(\frac{\text{Ph}_i^c}{\omega_0^c} \right), \\ H_2^c &= \sum_{i=\text{th}_1+1}^{\text{th}_2} \frac{\text{Ph}_i^c}{\omega_1^c} \ln \left(\frac{\text{Ph}_i^c}{\omega_1^c} \right), \\ &\vdots \\ H_k^c &= \sum_{i=\text{th}_{k-1}+1}^L \frac{\text{Ph}_i^c}{\omega_{k-1}^c} \ln \left(\frac{\text{Ph}_i^c}{\omega_{k-1}^c} \right).\end{aligned}\quad (21)$$

The values of the probability occurrence ($\omega_0^c, \omega_1^c, \dots, \omega_{k-1}^c$) of the k classes are obtained using (16) and the probability distribution Ph_i^c with (10). Finally, it is necessary to use (6) to separate the pixels into the corresponding classes.

4. Multilevel Thresholding Using Harmony Search Algorithm (HSMA)

4.1. Harmony Representation. Each harmony (candidate solution) uses k different elements as decision variables within the optimization algorithm. Such decision variables represent a different threshold point th that is used for the segmentation. Therefore, the complete population is represented as

$$\begin{aligned}\mathbf{HM} &= [\mathbf{x}_1^c, \mathbf{x}_2^c, \dots, \mathbf{x}_{\text{HMS}}^c]^T, \\ \mathbf{x}_i^c &= [\text{th}_1^c, \text{th}_2^c, \dots, \text{th}_k^c],\end{aligned}\quad (22)$$

where T refers to the transpose operator, HMS is the size of the harmony memory, \mathbf{x}_i is the i th element of HM, and $c = 1, 2, 3$ is set for RGB images while $c = 1$ is chosen for gray scale images. For this problem, the boundaries of the search space are set to $l = 0$ and $u = 255$, which correspond to image intensity levels.

4.2. HMA Implementation. The proposed segmentation algorithm has been implemented considering two different objective functions: Otsu and Kapur. Therefore, the HSA has been

coupled with the Otsu and Kapur functions, producing two different segmentation algorithms. The implementation of both algorithms can be summarized as in Algorithm 2.

4.3. Parameter Setting. The performance of HSA is strongly influenced by values assigned to parameters HM, HMCR, PAR, BW, and NI. Determining the most appropriate parameter values for an arbitrary problem is a complex issue, since such parameters interact to each other in a highly nonlinear manner, and no mathematical models of such interaction currently exist. The common method to find the best set of parameter values is to fix each parameter value to a random number within the parameter limits and then HSA is executed. If the final result is not satisfactory; then a new set of parameter values is defined, and the evolutionary algorithm is executed again. Evidently, this process can be very expensive (in terms of computational time), since many different trials may be required before reaching a set of satisfactory parameter values. Additionally, the set of values chosen by the user are not necessarily the best possible, but only the best from the arbitrary number of trials performed by the user. In order to reduce the number of experiments in this paper, it has used the factorial design method proposed in [32, 33] to systematically identify the best parameters of HSA.

The factorial design method [34] is a statistical technique that evaluates at the same time all process variables in order to determine which ones really exert significant influence on the final response. All variables are called factors and the different values chosen to study the factors are called levels. The factors to be considered in the factorial design are the HSA parameters, the harmony memory (HM), the harmony-memory consideration rate (HMCR), the pitch adjusting rate (PAR), the distance bandwidth (BW), and the number of improvisations (NI), whereas the response is the best fitness value obtained as a consequence of the HSA execution. Table 1 show the levels of the quantitative factors used in the factorial design. The values of zero level (central point) are based on the suggestions of the literature [33].

Each experiment is conducted combining the two different levels that define each parameter considering as a problem an image histogram example. Since the factors are five, a 2^{5-1} fractional factorial design is chosen, requiring sixteen optimization experiments plus one optimization trial for the central point. The results obtained from seventeen runs were analyzed according to [32, 33] using a general linear form of analysis of variance (ANOVA) [34] considering a 95% of confidence. After such analysis, it was found that the best possible configuration of HSA is shown in Table 2. These results were consistent considering six replications using different image histograms and the Otsu (14) or Kapur (20) functions, indistinctly. For more information on how to build fractional factorial designs, the reader is referred to [32, 33].

5. Experimental Results

The HSMA has been tested under a set of 11 benchmark images. Some of these images are widely used in the image

TABLE 1: Levels of the factors used for the factorial design method.

HSA parameters (Factors)	(-) Level	Central point	(+) Level
HM	50	100	200
HMCR	0.5	0.75	0.9
PAR	0.1	0.5	0.9
BW	0.1	0.3	0.5
NI	200	250	300

TABLE 2: HSMA parameter values obtained by the factorial design method.

HM	HMCR	PAR	BW	NI
100	0.75	0.5	0.5	300

processing literature to test different methods (Lena, Cameraman, Hunter, Baboon, etc.) [3, 9]. All the images have the same size (512×512 pixels), and they are in JPEG format. For the sake of representation, only five images which are presented in Figure 1 have been used to show the visual results; however, the numerical outcomes are analyzed considering the complete set.

Since HSMA is stochastic, it is necessary to employ an appropriate statistical metrics to measure its efficiency. Hence, the results have been reported executing the algorithm 35 times for each image. In order to maintain compatibility with similar works reported in the literature [14, 15, 18, 19], the number of thresholds points used in the test are $th = 2, 3, 4, 5$. In the experiments, the stop criterion is the number of times in which the best fitness values remain with no change. Therefore, if the fitness value for the best harmony remains unspoiled in 10% of the total number of iterations (NI), then the HSA is stopped.

To evaluate the stability and consistency, it has been computed the standard deviation (STD) from the results obtained in the 35 executions. Since the STD represents a measure about how the data are dispersed, the algorithm becomes more instable as the STD value increases [19]. Equation (23) shows the model used to calculate the STD value:

$$STD = \sqrt{\frac{\sum_{i=1}^{NI} (bf_i - av)^2}{Ru}}, \quad (23)$$

where bf_i is the best fitness of the i th iteration, av is the average value of bf , and Ru is the number of total executions ($Ru = 35$).

On the other hand, as an index of quality, the peak-to-signal ratio (PSNR) is used to assess the similarity of the segmented image against a reference image (original image) based on the produced mean square error (MSE) [18, 35]. Both PSNR and MSE are defined as

$$PSNR = 20 \log_{10} \left(\frac{255}{RMSE} \right), \quad (dB) \quad (24)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{ro} \sum_{j=1}^{co} (I_o^c(i, j) - I_{th}^c(i, j))^2}{ro \times co}}$$

TABLE 3: Result after applying the HSMA using Otsu's function to the set of benchmark images.

Image	k	Thresholds x_{best}^c	PSNR	STC
Camera man	2	70, 144	17.247	$2.30E - 12$
	3	59, 119, 156	20.211	$1.55E - 02$
	4	42, 95, 140, 170	21.533	$2.76E - 12$
	5	36, 82, 122, 149, 173	23.282	$5.30E - 03$
Lena	2	91, 150	15.401	$9.22E - 13$
	3	79, 125, 170	17.427	$2.99E - 02$
	4	73, 112, 144, 179	18.763	$2.77E - 01$
	5	71, 107, 134, 158, 186	19.443	$3.04E - 01$
Baboon	2	97, 149	15.422	$6.92E - 13$
	3	85, 125, 161	17.709	$1.92E - 02$
	4	71, 105, 136, 167	20.289	$5.82E - 02$
	5	66, 97, 123, 147, 173	21.713	$4.40E - 01$
Hunter	2	51, 116	17.875	$2.30E - 12$
	3	36, 86, 135	20.350	$2.30E - 12$
	4	27, 65, 104, 143	22.203	$1.22E - 02$
	5	22, 53, 88, 112, 152	23.703	$1.84E - 12$
Airplane	2	113, 173	15.029	$9.22E - 13$
	3	92, 144, 190	18.854	$4.83E - 01$
	4	84, 129, 172, 203	20.735	$7.24E - 01$
	5	68, 106, 143, 180, 205	23.126	$8.38E - 01$
Peppers	2	72, 138	16.299	$1.38E - 12$
	3	65, 122, 169	18.359	$4.61E - 13$
	4	50, 88, 128, 171	20.737	$4.61E - 13$
	5	48, 85, 118, 150, 179	22.310	$1.84E - 012$
Living room	2	87, 145	15.999	$1.15E - 12$
	3	76, 123, 163	18.197	$6.92E - 12$
	4	56, 97, 132, 168	20.673	$9.22E - 12$
	5	49, 88, 120, 146, 178	22.225	$2.86E - 02$
Blonde	2	106, 155	14.609	$6.92E - 13$
	3	53, 112, 158	19.157	$9.23E - 13$
	4	50, 103, 139, 168	20.964	$5.48E - 01$
	5	49, 92, 121, 152, 172	22.409	$6.50E - 01$
Bridge	2	91, 56	13.943	$4.61E - 13$
	3	72, 120, 177	17.019	$7.10E - 01$
	4	63, 103, 145, 193	18.872	$2.91E - 01$
	5	56, 91, 124, 159, 201	20.539	$3.57E - 01$
Butterfly	2	99, 151	13.934	$7.30E - 02$
	3	82, 119, 160	16.932	$6.17E - 01$
	4	71, 102, 130, 163	19.259	$3.07E + 00$
	5	62, 77, 109, 137, 167	21.450	$3.87E + 00$
Lake	2	85, 154	14.638	$4.61E - 13$
	3	78, 140, 194	15.860	$1.84E - 12$
	4	67, 110, 158, 198	17.629	$2.68E - 01$
	5	57, 88, 127, 166, 200	19.416	$1.12E - 01$

where I_o^c is the original image, I_{th}^c is the segmented image, $c = 1$ for gray scale, and $c = 3$ for RGB images, whereas ro,

co are the total number of rows and columns of the image, respectively.

5.1. Otsu's Results. This section analyzes the results of HSMA after considering the variance among classes (14) as the objective function, just as it has been proposed by Otsu [6]. The approach is applied over the complete set of benchmark images, whereas the results are registered in Table 3. Such results present the best threshold values after testing the proposed method with four different threshold points $\text{th} = 2, 3, 4, 5$. The table also features the PSNR and the STD values. It is evident that the PSNR and STD values increase their magnitude as the number of threshold points also increases.

For the sake of representation, it has been selected only five images of the set to show (graphically) the segmentation results. Figure 1 presents the images selected from the benchmark set and their respective histograms which possess irregular distributions (see Figure 1(j) in particular). Under such circumstances, classical methods face great difficulties to find the best threshold values.

The processing results for the selected original images are presented in five tables: Tables 4, 5, 6, 7, and 8. Such results show the segmented images considering four different threshold points, $\text{th} = 2, 3, 4, 5$. The tables also show the evolution of the objective function during one execution.

5.2. Kapur's Results. This section analyzes the performance of HSMA after considering the entropy function (20) as objective function, as it has been proposed by Kapur et al. in [7]. Table 9 presents the experimental results after the application of HSMA over the entire set of benchmark images. The values listed are PSNR, STD, and the best threshold values of the last population (x_t^B). The same test procedure that was previously applied to the Otsu's method (Section 5.1) is used with the Kapur's method, also considering the same stop criterion and a similar HSA parameter configuration.

The results after applying the HSMA to the selected benchmark images are presented in Tables 10, 11, 12, 13, and 14. Four different threshold points have been employed: $\text{th} = 2, 3, 4, 5$. All tables exhibit the segmented image, the approximated histogram, and the evolution of the fitness value during the execution of the HSA method.

From the results of both Otsu's and Kapur's methods, it is possible to appreciate that the HSMA converges (stabilizes) after a determined number of iterations depending on the th value. For experimental purposes HSMA continues running still further, even though the stop criterion is achieved. In this way, the graphics show that convergence is often reached in the first iterations of the optimization process. The segmented images provide evidence that the outcome is better with $\text{th} = 4$ and $\text{th} = 5$; however, if the segmentation task does not require to be extremely accurate then it is possible to select $\text{th} = 3$.

5.3. Comparisons. In order to analyze the results of HSMA, two different comparisons are executed. The first one involves the comparison between the two versions of the proposed approach, one with the Otsu function and the other with

TABLE 4: Results after applying the HSMA using Otsu's over the Camera man image.

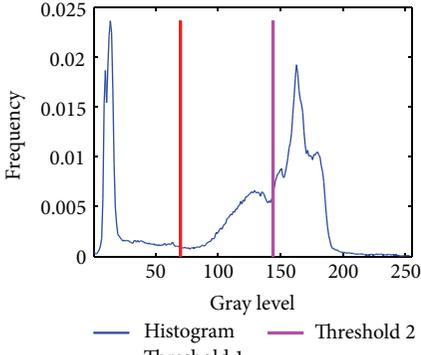
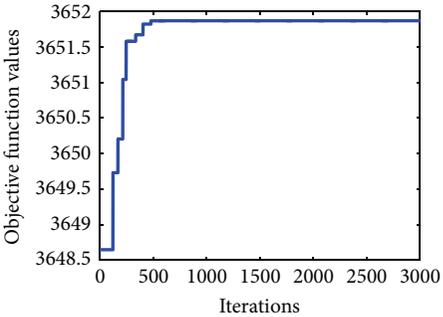
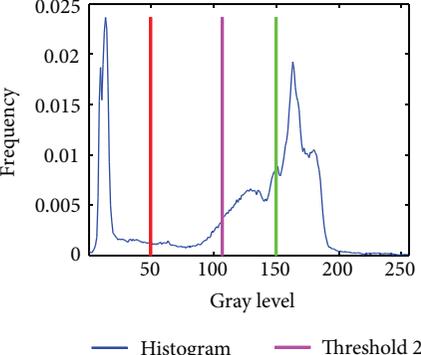
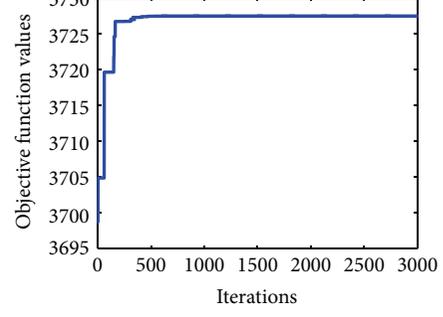
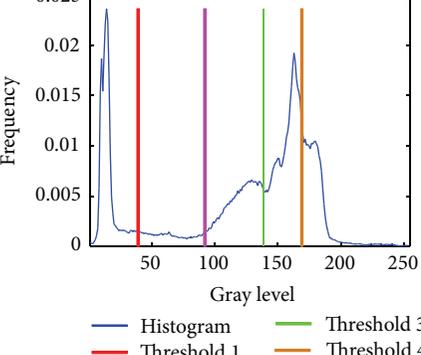
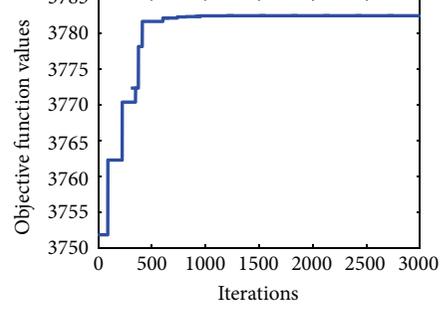
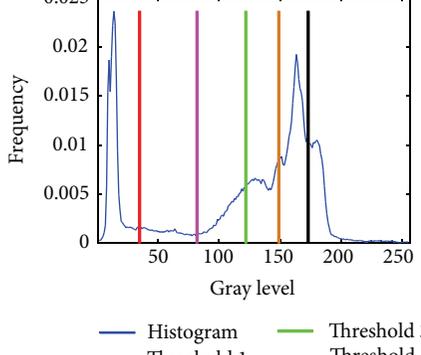
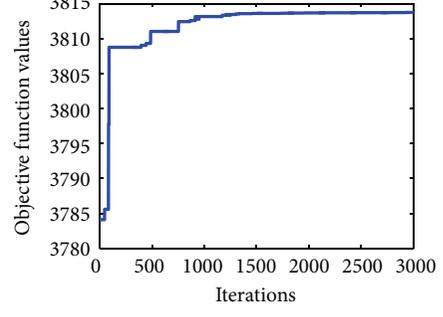
th = 2		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1</p>	 <p>Objective function values</p> <p>Iterations</p>
th = 3		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1 — Threshold 3</p>	 <p>Objective function values</p> <p>Iterations</p>
th = 4		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2</p>	 <p>Objective function values</p> <p>Iterations</p>
th = 5		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2 — Threshold 5</p>	 <p>Objective function values</p> <p>Iterations</p>

TABLE 5: Results after applying the HSMA using Otsu's over Lena image.

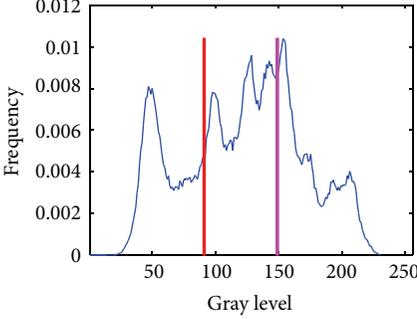
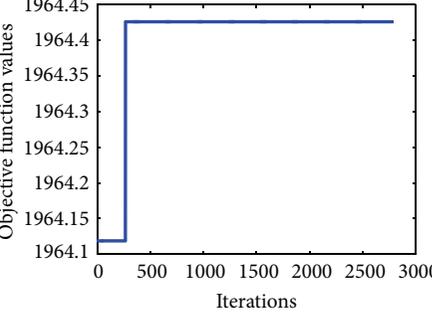
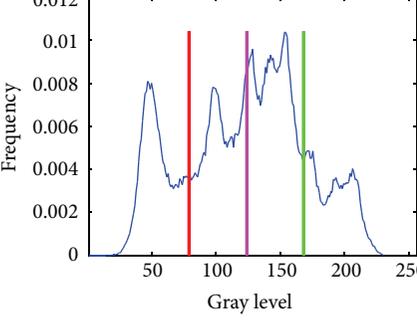
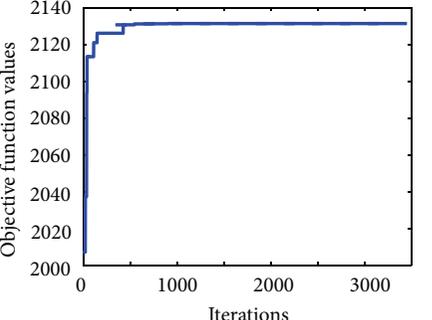
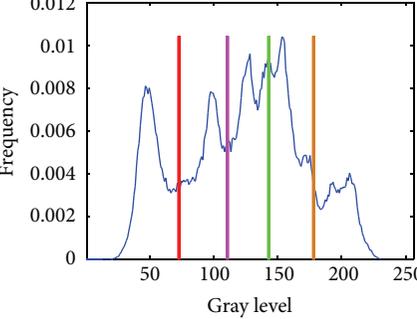
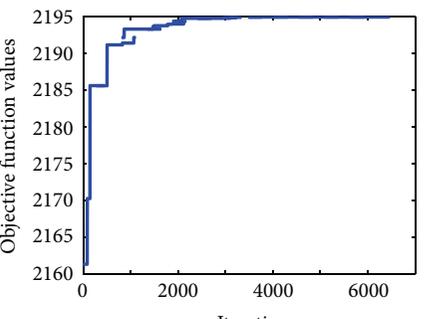
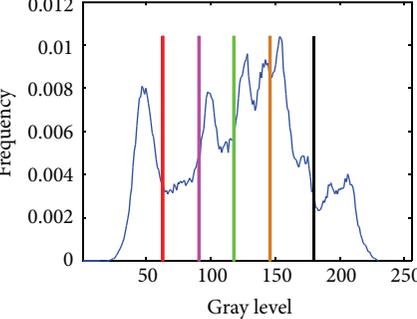
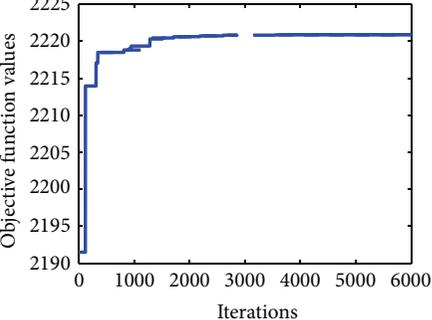
<p>th = 2</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 3</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1 — Threshold 3</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 4</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 5</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2 — Threshold 5</p>	 <p>Objective function values</p> <p>Iterations</p>

TABLE 6: Results after applying the HSMA using Otsu's over the Baboon image.

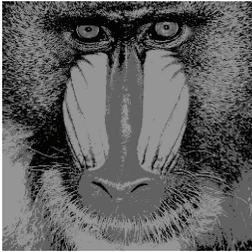
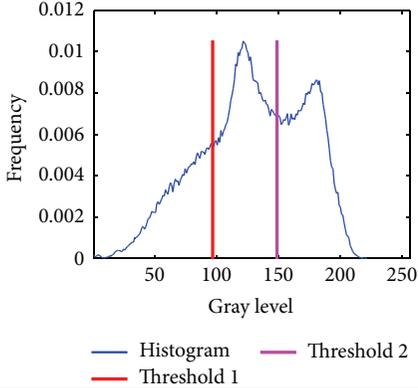
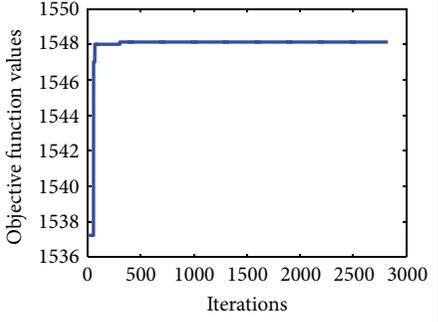
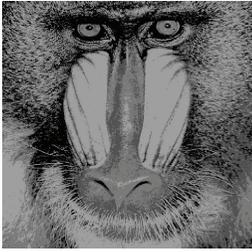
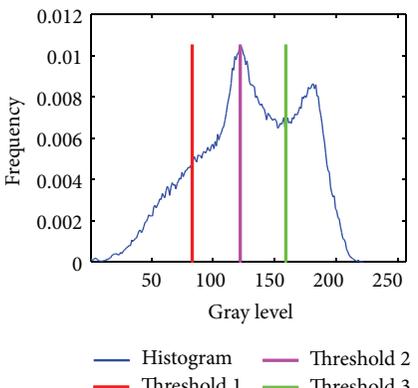
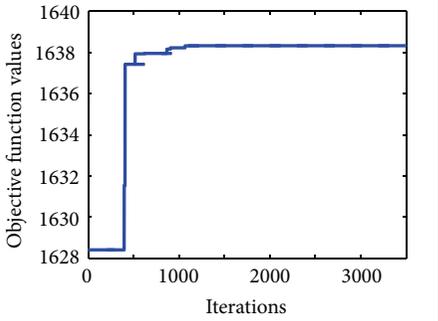
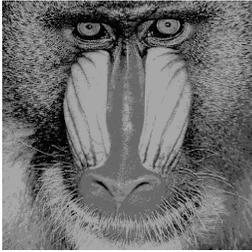
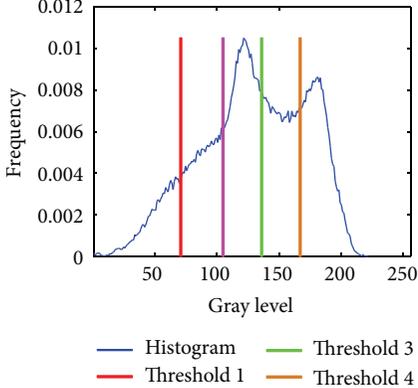
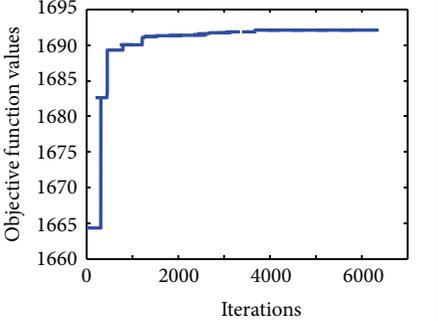
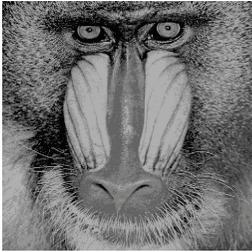
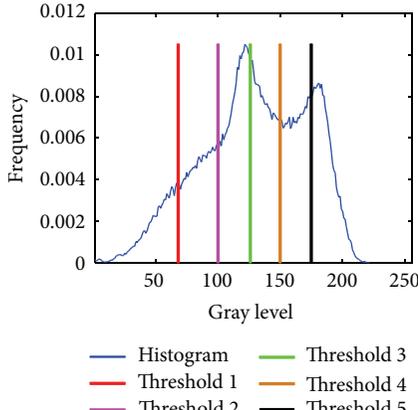
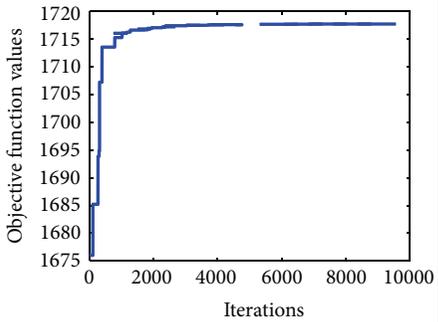
<p>th = 2</p>		 <p>— Histogram — Threshold 2 — Threshold 1</p>	
<p>th = 3</p>		 <p>— Histogram — Threshold 2 — Threshold 1 — Threshold 3</p>	
<p>th = 4</p>		 <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2</p>	
<p>th = 5</p>		 <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2 — Threshold 5</p>	

TABLE 7: Results after applying the HSMA using Otsu's over the Hunter image.

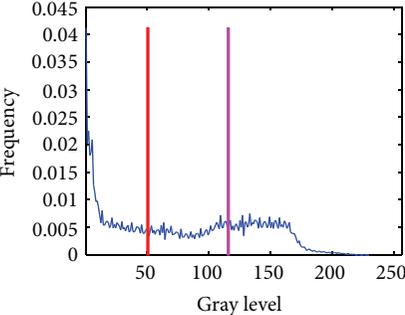
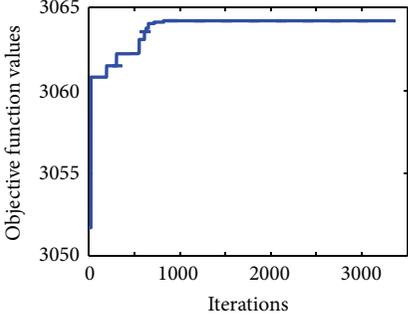
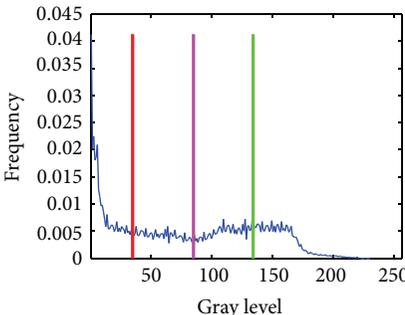
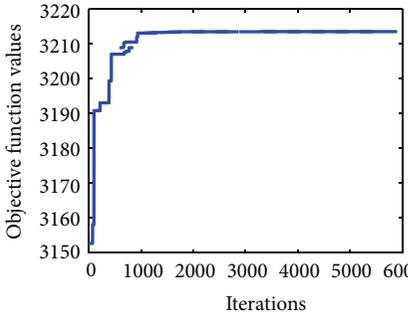
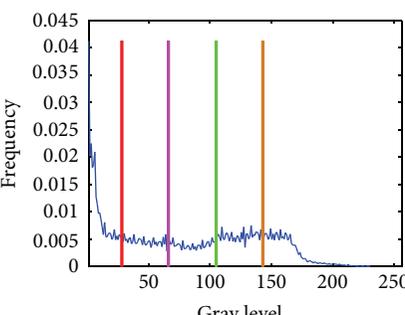
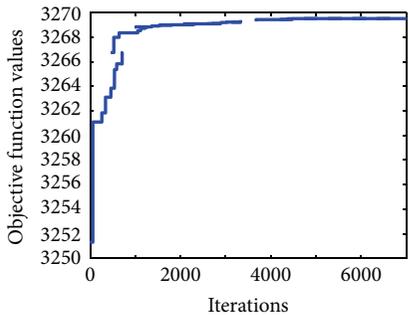
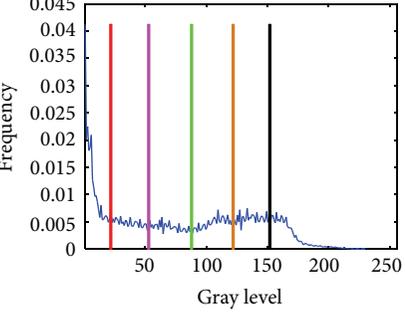
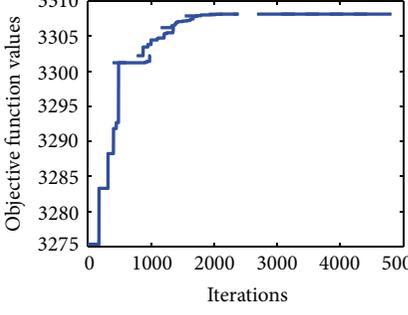
<p>th = 2</p>		 <p>— Histogram — Threshold 2 — Threshold 1</p>	
<p>th = 3</p>		 <p>— Histogram — Threshold 2 — Threshold 1 — Threshold 3</p>	
<p>th = 4</p>		 <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2</p>	
<p>th = 5</p>		 <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2 — Threshold 5</p>	

TABLE 8: Results after applying the HSMA using Otsu's over the Butterfly image.

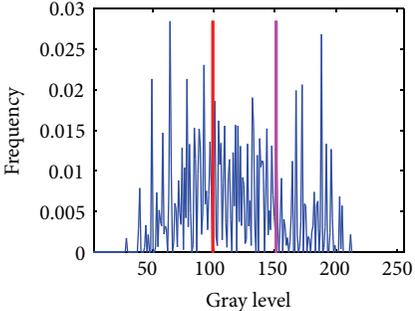
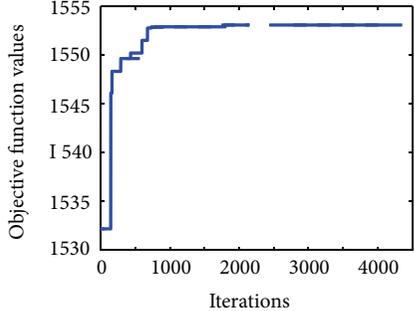
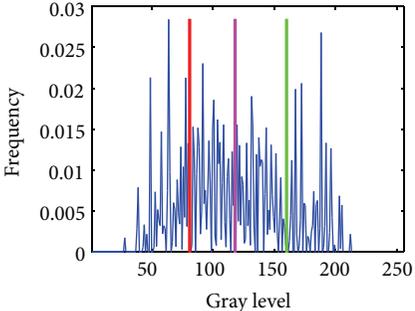
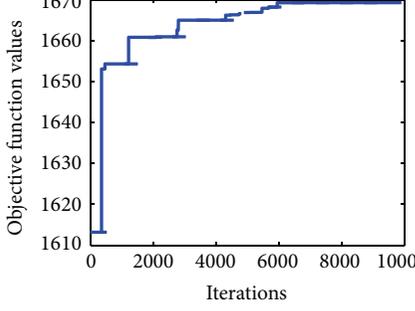
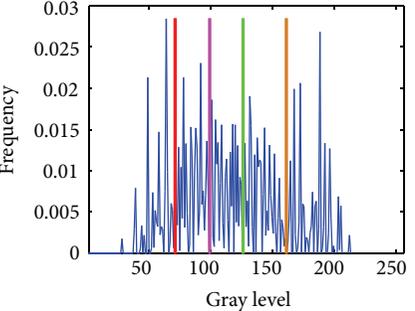
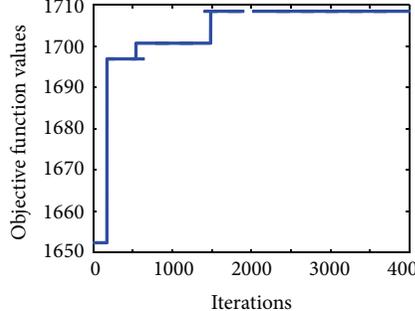
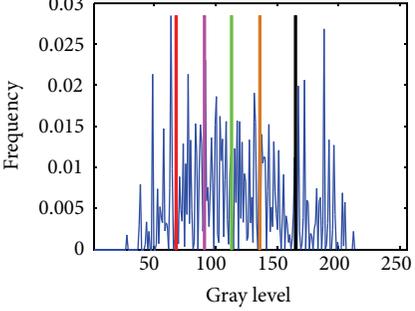
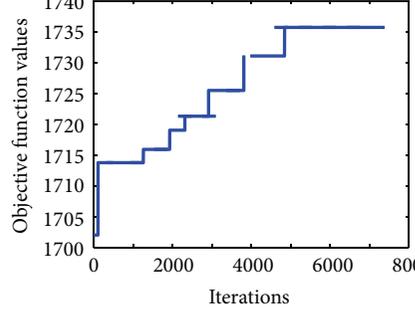
<p>th = 2</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 3</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1 — Threshold 3</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 4</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 5</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2 — Threshold 5</p>	 <p>Objective function values</p> <p>Iterations</p>

TABLE 9: Results after applying the HSMA using Kapur's function to the set of benchmark images.

Image	k	Thresholds x_{best}^c	PSNR	STD
Camera man	2	128, 196	13.626	$3.60E - 15$
	3	44, 103, 196	14.460	$1.40E - 03$
	4	44, 96, 146, 196	20.153	$1.20E - 03$
	5	24, 60, 98, 146, 196	20.661	$2.75E - 02$
Lena	2	96, 163	14.638	$3.60E - 15$
	3	23, 96, 163	16.218	$7.66E - 02$
	4	23, 80, 125, 173	19.287	$1.44E - 14$
	5	23, 71, 109, 144, 180	21.047	$1.22E - 02$
Baboon	2	79, 143	16.016	$1.08E - 14$
	3	79, 143, 231	16.016	$7.19E - 02$
	4	44, 98, 152, 231	18.485	$8.47E - 02$
	5	33, 74, 114, 159, 231	20.507	$1.08E - 14$
Hunter	2	92, 179	15.206	$1.44E - 14$
	3	59, 117, 179	18.500	$4.16E - 04$
	4	44, 89, 133, 179	21.065	$4.31E - 04$
	5	44, 89, 133, 179, 222	21.086	$3.43E - 02$
Airplane	2	70, 171	15.758	$3.30E - 03$
	3	68, 126, 182	18.810	$1.08E - 14$
	4	68, 126, 182, 232	18.810	$1.82E - 01$
	5	64, 104, 143, 184, 232	20.321	$1.80E - 01$
Peppers	2	66, 143	16.265	$7.21E - 15$
	3	62, 112, 162	18.367	$1.80E - 14$
	4	62, 112, 162, 227	18.376	$2.39E - 02$
	5	48, 86, 127, 171, 227	18.827	$4.17E - 04$
Living room	2	89, 170	14.631	$1.40E - 03$
	3	47, 103, 175	17.146	$2.70E - 03$
	4	47, 98, 149, 197	19.144	$1.34E - 02$
	5	42, 85, 125, 162, 197	21.160	$1.89E - 02$
Blonde	2	125, 203	12.244	$1.44E - 14$
	3	65, 134, 203	16.878	$1.00E - 03$
	4	65, 113, 155, 203	20.107	$5.50E - 03$
	5	65, 113, 155, 203, 229	20.107	$4.48E - 02$
Bridge	2	94, 171	13.529	$7.40E - 03$
	3	65, 131, 195	16.806	$1.44E - 02$
	4	53, 102, 151, 199	18.902	$2.47E - 02$
	5	40, 85, 131, 171, 211	20.268	$2.12E - 02$
Butterfly	2	27, 213	8.1930	$2.25E - 02$
	3	27, 120, 213	13.415	$8.60E - 04$
	4	27, 96, 144, 213	16.725	$3.80E - 03$
	5	27, 83, 118, 152, 213	19.413	$3.90E - 03$
Lake	2	91, 163	14.713	$1.44E - 14$
	3	73, 120, 170	16.441	$3.05E - 04$
	4	69, 112, 156, 195	17.455	$4.53E - 02$
	5	62, 96, 131, 166, 198	18.774	$3.66E - 02$

TABLE 10: Results after applying the HSMA using Kapur's over the Camera man image.

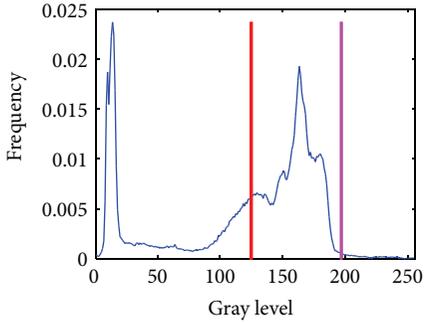
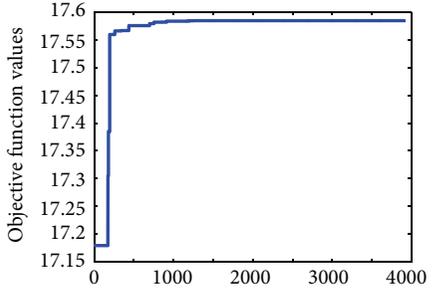
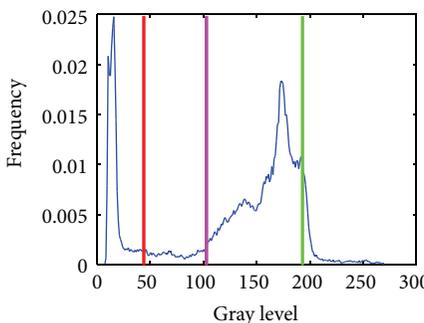
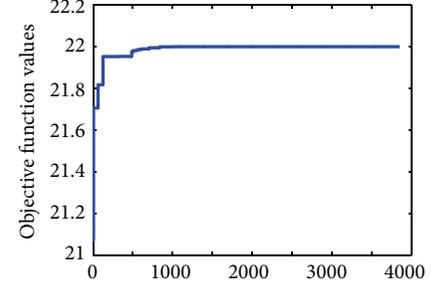
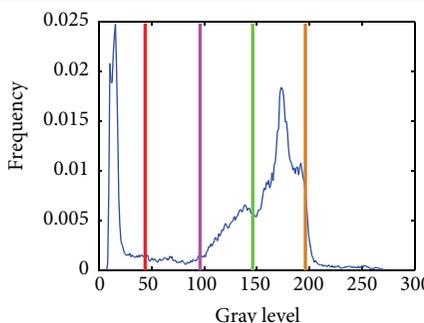
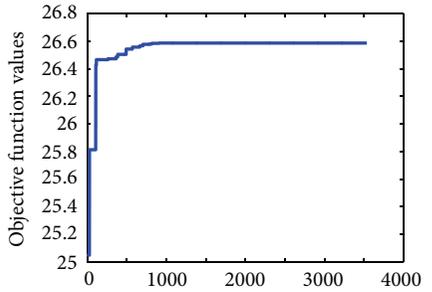
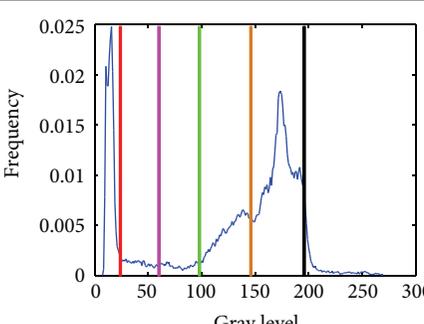
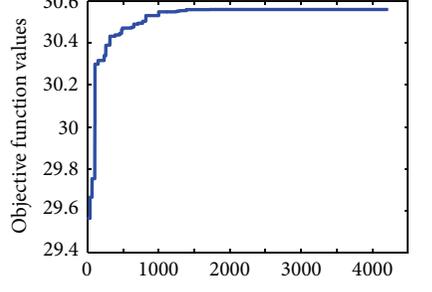
<p>th = 2</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 3</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1 — Threshold 3</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 4</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 5</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2 — Threshold 5</p>	 <p>Objective function values</p> <p>Iterations</p>

TABLE II: Results after applying the HSMA using Kapur's over Lena's image.

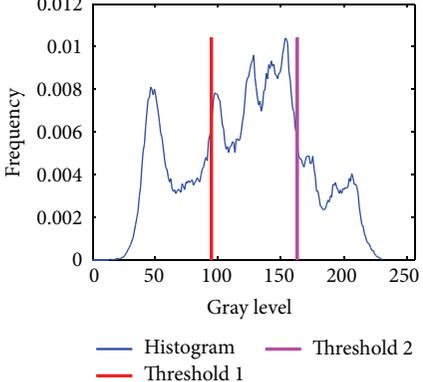
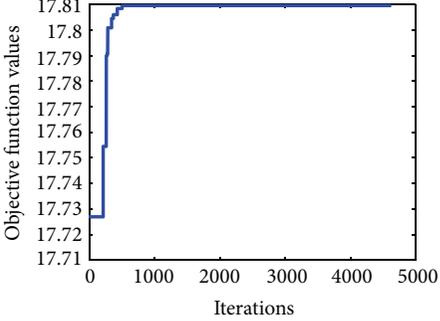
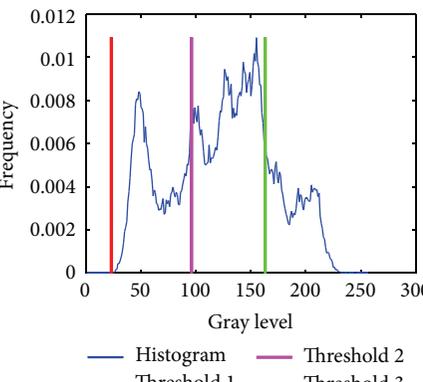
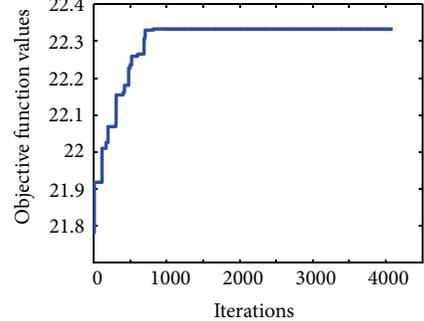
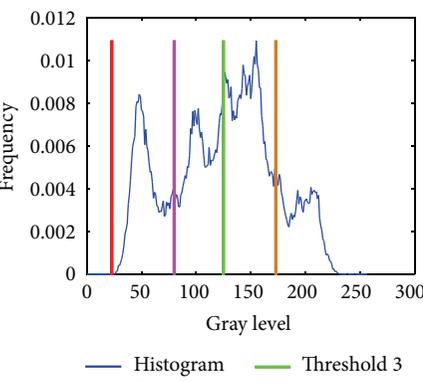
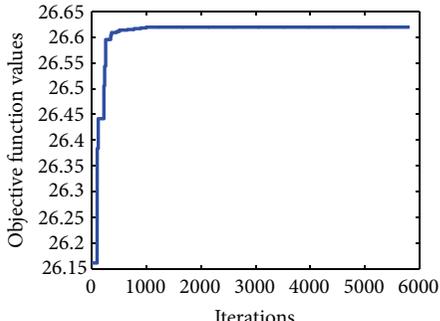
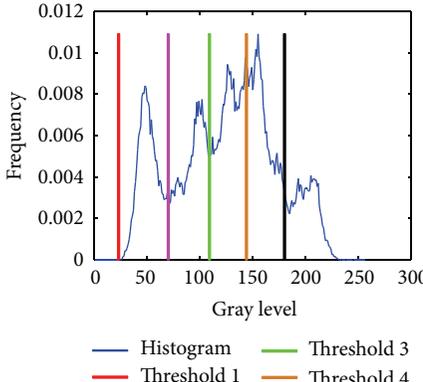
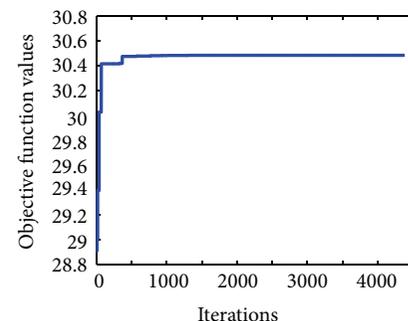
th = 2		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1</p>	 <p>Objective function values</p> <p>Iterations</p>
th = 3		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1 — Threshold 3</p>	 <p>Objective function values</p> <p>Iterations</p>
th = 4		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2</p>	 <p>Objective function values</p> <p>Iterations</p>
th = 5		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2 — Threshold 5</p>	 <p>Objective function values</p> <p>Iterations</p>

TABLE 12: Results after applying the HSMA using Kapur's over the Baboon image.

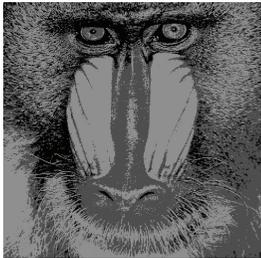
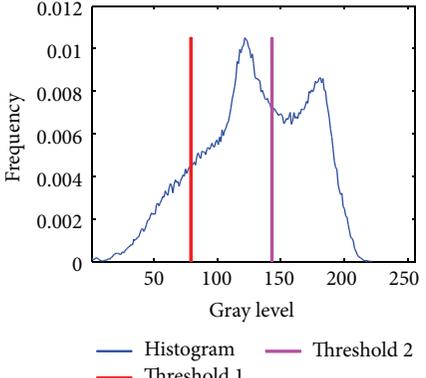
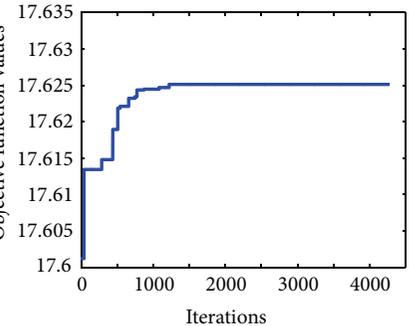
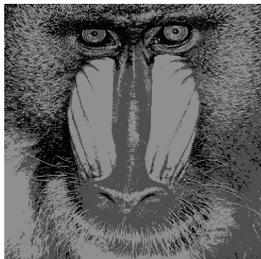
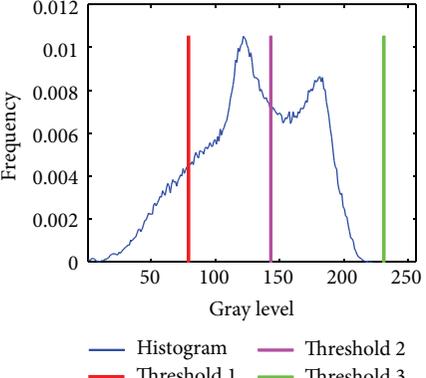
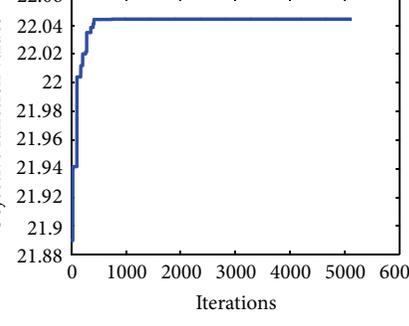
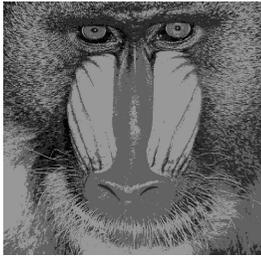
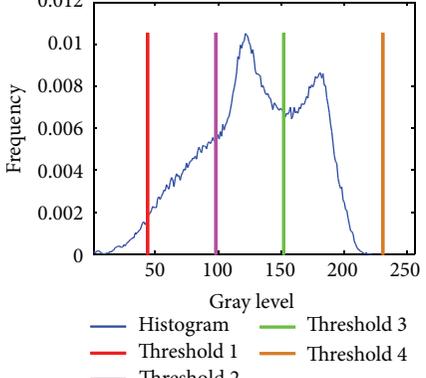
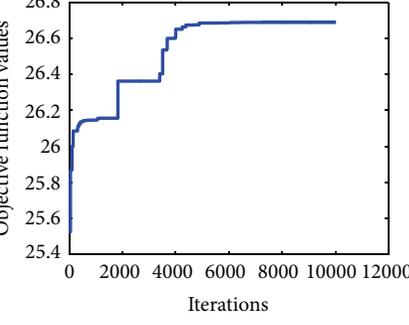
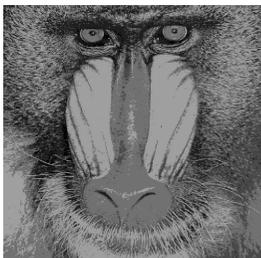
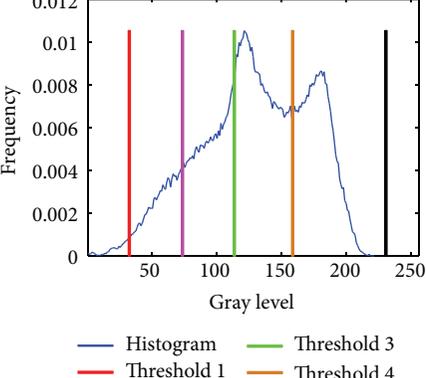
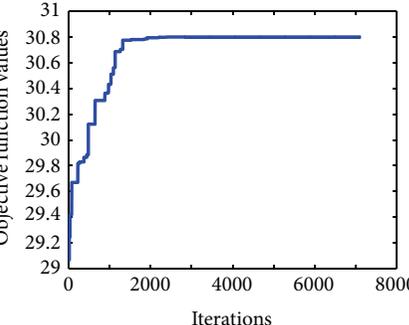
<p>th = 2</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 3</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1 — Threshold 3</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 4</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 5</p>		 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2 — Threshold 5</p>	 <p>Objective function values</p> <p>Iterations</p>

TABLE 13: Results after applying the HSMA using Kapur's over the Hunter image.

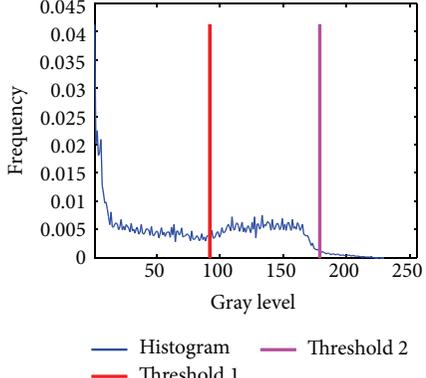
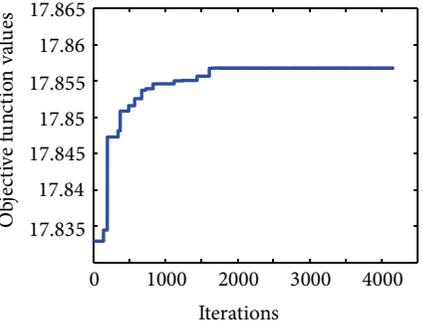
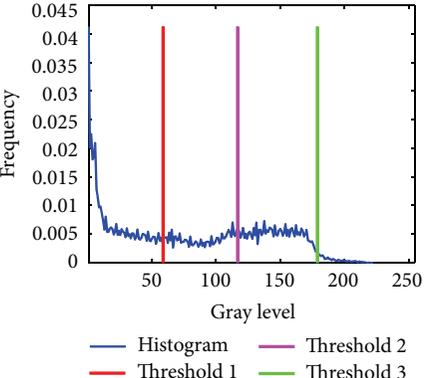
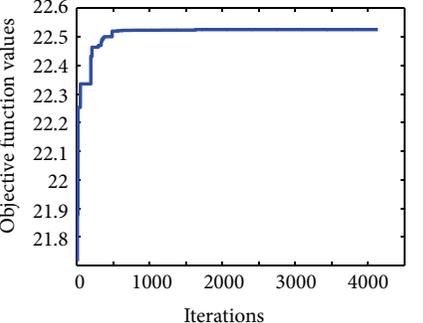
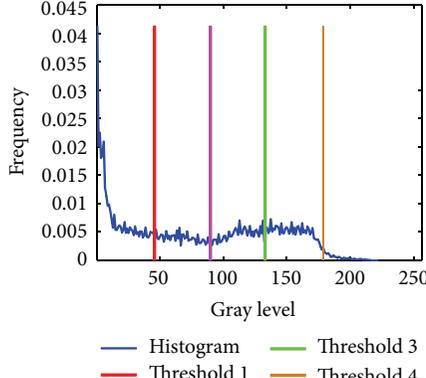
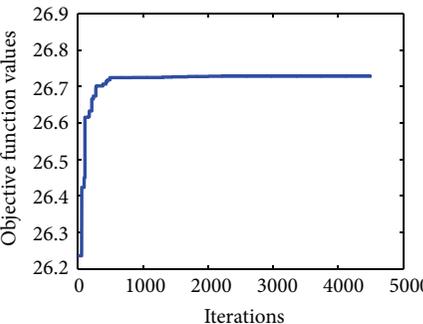
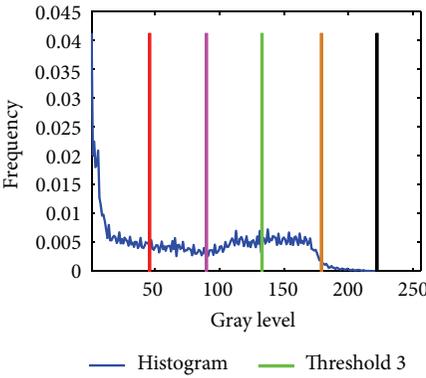
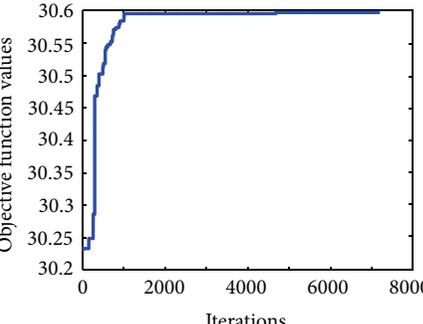
<p>th = 2</p>		 <p>— Histogram — Threshold 2 — Threshold 1</p>	
<p>th = 3</p>		 <p>— Histogram — Threshold 2 — Threshold 1 — Threshold 3</p>	
<p>th = 4</p>		 <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2</p>	
<p>th = 5</p>		 <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2 — Threshold 5</p>	

TABLE 14: Results after applying the HSMA using Kapur's over the Butterfly image.

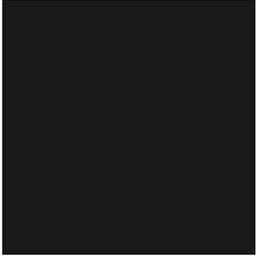
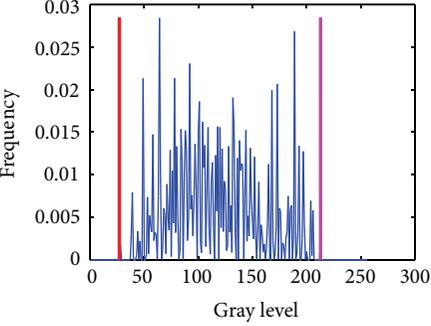
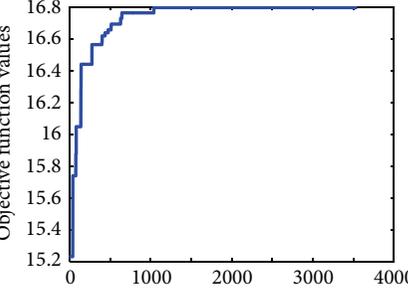
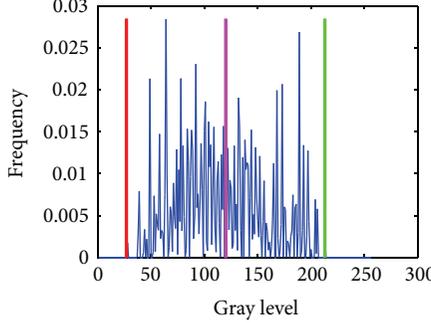
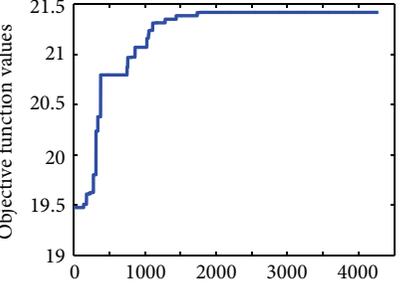
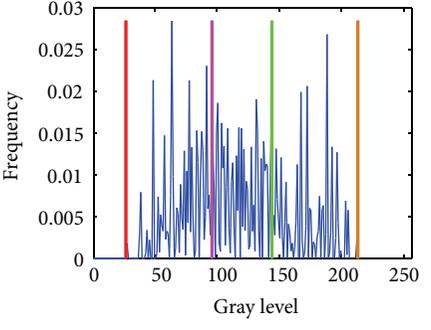
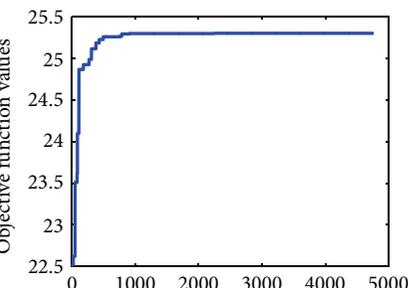
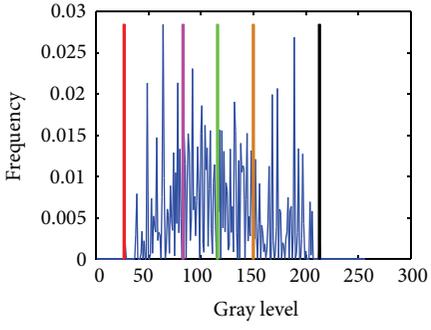
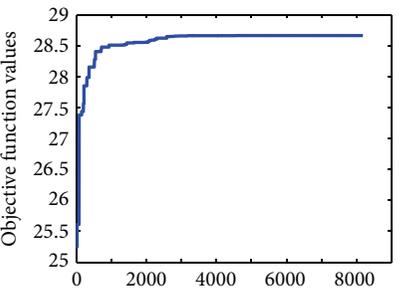
<p>th = 2</p> 	 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 3</p> 	 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 2 — Threshold 1 — Threshold 3</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 4</p> 	 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2</p>	 <p>Objective function values</p> <p>Iterations</p>
<p>th = 5</p> 	 <p>Frequency</p> <p>Gray level</p> <p>— Histogram — Threshold 3 — Threshold 1 — Threshold 4 — Threshold 2 — Threshold 5</p>	 <p>Objective function values</p> <p>Iterations</p>

TABLE 15: P values produced by Wilcoxon's test comparing Otsu versus Kapur over the averaged PSNR from Tables 3 and 9.

Image	k	P -value Otsu versus Kapur
Camera man	2	$1.0425E - 16$
	3	$2.1435E - 15$
	4	$2.6067E - 16$
	5	$6.2260E - 16$
Lena	2	$1.0425E - 16$
	3	$9.4577E - 15$
	4	$9.7127E - 15$
	5	$1.2356E - 12$
Baboon	2	$1.0425E - 16$
	3	$1.7500E - 02$
	4	$5.3417E - 14$
	5	$1.4013E - 14$
Hunter	2	$1.0425E - 16$
	3	$2.6067E - 16$
	4	$6.6386E - 14$
	5	$6.4677E - 15$
Airplane	2	$1.0425E - 16$
	3	$4.6500E - 02$
	4	$1.7438E - 13$
	5	$6.2475E - 13$
Peppers	2	$1.0425E - 16$
	3	$1.0425E - 16$
	4	$6.0498E - 15$
	5	$5.3194E - 15$
Living room	2	$3.2405E - 15$
	3	$1.4223E - 15$
	4	$1.3175E - 14$
	5	$6.2220E - 14$
Blonde	2	$1.0425E - 16$
	3	$2.6067E - 16$
	4	$4.0480E - 13$
	5	$7.8167E - 04$
Bridge	2	$7.3588E - 06$
	3	$1.1300E - 04$
	4	$1.9400E - 02$
	5	$2.1900E - 02$
Butterfly	2	$1.1615E - 14$
	3	$2.5697E - 14$
	4	$3.7190E - 13$
	5	$1.7941E - 06$
Lake	2	$1.0425E - 16$
	3	$5.6120E - 16$
	4	$9.8174E - 14$
	5	$2.2292E - 14$

the Kapur criterion. The second analyses the comparison between the HSMA and other state-of-the-art approaches.

5.3.1. Comparison between Otsu and Kapur HSMA. In order to statistically compare the results from Tables 3 and 9, a non-parametric significance proof known as the Wilcoxon's rank test [36, 37] for 35 independent samples has been conducted. Such proof allows assessing result differences among two related methods. The analysis is performed considering a 5% significance level over the peak-to-signal ratio (PSNR) data corresponding to the five threshold points. Table 15 reports the P values produced by Wilcoxon's test for a pairwise comparison of the PSNR values between the Otsu and Kapur objective functions. As a null hypothesis, it is assumed that there is no difference between the values of the two objective functions. The alternative hypothesis considers an existent difference between the values of both approaches. All P values reported in Table 15 are less than 0.05 (5% significance level) which is a strong evidence against the null hypothesis, indicating that the Otsu PSNR mean values for the performance are statistically better and it has not occurred by chance.

5.3.2. Comparison among HSMA and Other MT Approaches. The results produced by HSMA have been compared with those generated by state-of-the-art thresholding methods such genetic algorithms (GA) [15], particle swarm optimization (PSO) [18], and bacterial foraging (BF) [19].

All the algorithms run 35 times over each selected image. The images at this test are the same as in Sections 5.2 and 5.1 (Camera man, Lena, Baboon, Hunter, and Butterfly). For each image, the PSNR, the STD, and the mean of the objective function values are calculated. Moreover, the entire test is performed using both Otsu's and Kapur's objective functions.

Table 16 presents the computed values for a reduced benchmark test (five images). It is clear that the HSMA delivers better performance than the others. Such values are computed using the Otsu's method as the objective function. On the other hand, the same experiment has been performed using the Kapur's method. Using the same criteria (as those described for the Otsu's method), the algorithm runs over 35 times at each image. The results of this experiment are presented in Table 17 and show that the proposed HSMA algorithm is better in comparison with the GA, PSO, and BF.

6. Conclusions

In this paper, a MT-based method on the original harmony search algorithm (HSA) is presented. The approach combines the good search capabilities of HSA algorithm and the use of some objective functions that have been proposed by the popular MT methods of Otsu and Kapur. In order to measure the performance of the proposed approach, the peak signal-to-noise ratio (PSNR) is used to assess the segmentation quality by considering the coincidences between the segmented and the original images. In this work, a simple HSA implementation without any modification is considered in order to demonstrate that it can be applied to image processing tasks.

The study explores the comparison between two versions of HSMA: one employs the Otsu objective function while the other uses the Kapur criterion. Results show that the

TABLE 16: Comparisons between HSMA, GA, PSO, and BF, applied over the selected test images using Otsu's method.

Image	k	HSMA			GA			PSO			BF		
		PSNR	STD	Mean	PSNR	STD	Mean	PSNR	STD	Mean	PSNR	STD	Mean
Camera man	2	17.247	$2.30E - 12$	3651.9	17.048	0.0232	3604.5	17.033	0.0341	3598.3	17.058	0.0345	3590.9
	3	20.211	$1.55E - 02$	3727.4	17.573	0.1455	3678.3	19.219	0.2345	3662.7	20.035	0.2459	3657.5
	4	21.533	$2.76E - 12$	3782.4	20.523	0.2232	3781.5	21.254	0.3142	3777.4	21.209	0.4560	3761.4
	5	23.282	$5.30E - 03$	3813.7	21.369	0.4589	3766.4	22.095	0.5089	3741.6	22.237	0.5089	3789.8
Lena	2	15.401	$9.22E - 13$	1964.4	15.040	0.0049	1960.9	15.077	0.0033	1961.4	15.031	$2.99E - 04$	1961.5
	3	17.427	$2.99E - 02$	2131.4	17.304	0.1100	2126.4	17.276	0.0390	2127.7	17.401	0.0061	2128.0
	4	18.763	$2.77E - 01$	2194.9	17.920	0.2594	2173.7	18.305	0.1810	2180.6	18.507	0.0081	2189.0
	5	19.443	$3.04E - 01$	2218.7	18.402	0.3048	2196.2	18.770	0.2181	2212.5	19.001	0.0502	2215.6
Baboon	2	15.422	$6.92E - 13$	1548.1	15.304	0.0031	1547.6	15.088	0.0077	1547.9	15.353	$8.88E - 04$	1548.0
	3	17.709	$1.92E - 02$	1638.3	17.505	0.1750	1633.5	17.603	0.0816	1635.3	17.074	0.0287	1637.0
	4	20.289	$5.82E - 02$	1692.1	18.708	0.2707	1677.7	19.233	0.0853	1684.3	19.654	0.0336	1690.7
	5	21.713	$4.40E - 01$	1717.5	20.203	0.3048	1712.9	20.526	0.1899	1712.9	21.160	0.1065	1716.7
Hunter	2	17.875	$2.30E - 12$	3054.2	17.088	0.0470	3064.1	17.932	0.2534	3064.1	17.508	0.0322	3064.1
	3	20.350	$2.30E - 12$	3213.4	20.045	0.1930	3212.9	19.940	0.9727	3212.4	20.350	0.9627	3213.4
	4	22.203	$1.22E - 02$	3269.5	20.836	0.6478	3268.4	21.128	2.2936	3266.3	21.089	2.2936	3266.3
	5	23.703	$1.84E - 12$	3308.1	21.284	1.6202	3305.6	22.026	4.1811	3276.3	22.804	3.6102	3291.1
Butterfly	2	13.934	$7.30E - 02$	1553.0	13.007	0.0426	1553.0	13.092	0.0846	1553.0	13.890	0.0643	1553.0
	3	16.932	$6.17E - 01$	1669.2	15.811	0.3586	1669.0	17.261	2.6268	1665.7	17.285	1.2113	1667.2
	4	19.259	$3.07E + 00$	1708.3	17.104	0.6253	1709.9	17.005	3.7976	1702.9	17.128	2.2120	1707.0
	5	21.450	$3.87E + 00$	1728.0	18.593	0.5968	1734.4	18.099	6.0747	1730.7	18.9061	3.5217	1733.0

TABLE 17: Comparison between HSMA GA, PSO, and BF, applied over selected test images using Kapur's method.

Image	k	HSMA			GA			PSO			BF		
		PSNR	STD	Mean	PSNR	STD	Mean	PSNR	STD	Mean	PSNR	STD	Mean
Camera man	2	13.626	$3.60E - 15$	17584	11.941	0.1270	15.341	12.259	0.1001	16.071	12.264	0.0041	16.768
	3	14.460	$1.40E - 03$	22.007	14.827	0.2136	20.600	15.211	0.1107	21.125	15.250	0.0075	21.498
	4	20.153	$1.20E - 03$	26.586	17.166	0.2857	24.267	18.000	0.2005	25.050	18.406	0.0081	25.093
	5	20.661	$2.75E - 02$	30.553	19.795	0.3528	28.326	20.963	0.2734	28.365	21.211	0.0741	30.026
Lena	2	14.638	$3.60E - 15$	17809	12.334	0.0049	16.122	12.345	0.0033	16.916	12.345	$2.99E - 4$	16.605
	3	16.218	$7.66E - 02$	22.306	14.995	0.1100	20.920	15.133	0.0390	20.468	15.133	0.0061	20.812
	4	19.287	$1.44E - 14$	26.619	17.089	0.2594	23.569	17.838	0.1810	24.449	17.089	0.0081	26.214
	5	21.047	$1.22E - 02$	30.485	19.549	0.3043	27.213	20.442	0.2181	27.526	19.549	0.0502	28.046
Baboon	2	16.016	$1.08E - 14$	17.625	12.184	0.0567	16.425	12.213	0.0077	16.811	12.216	$8.88E - 4$	16.889
	3	16.016	$7.19E - 02$	22.117	14.745	0.1580	21.069	15.008	0.0816	21.088	15.211	0.0287	21.630
	4	18.485	$8.47E - 02$	26.671	16.935	0.1765	25.489	17.574	0.0853	24.375	17.999	0.0336	25.446
	5	20.507	$1.08E - 14$	30.800	19.662	0.2775	29.601	20.224	0.1899	30.994	20.720	0.1065	30.887
Hunter	2	15.206	$1.44E - 14$	17.856	12.349	0.0148	16.150	12.370	0.0068	15.580	12.373	0.0033	16.795
	3	18.500	$4.16E - 04$	22.525	14.838	0.1741	21.026	15.128	0.0936	20.639	15.553	0.1155	21.860
	4	21.065	$4.31E - 04$	26.728	17.218	0.2192	25.509	18.040	0.1560	27.085	18.381	0.0055	26.230
	5	21.086	$3.43E - 02$	30.612	19.563	0.3466	29.042	20.533	0.2720	29.013	21.256	0.0028	28.856
Butterfly	2	8.1930	$2.25E - 02$	16.791	10.470	0.0872	15.481	10.474	0.0025	14.098	10.474	0.0014	15.784
	3	13.415	$8.60E - 04$	21.417	11.628	0.2021	20.042	12.313	0.1880	19.340	12.754	0.0118	21.308
	4	16.725	$3.80E - 03$	25.292	13.314	0.2596	23.980	14.231	0.2473	25.190	14.877	0.0166	25.963
	5	19.413	$3.90E - 03$	28.664	15.756	0.3977	27.411	16.337	0.2821	27.004	16.828	0.0877	27.980

Otsu function delivers better results than the Kapur criterion. Such conclusion has been statistically proved considering the Wilcoxon test.

The proposed approach has been compared to other techniques that implement different optimization algorithms like GA, PSO, and BF. The efficiency of the algorithm has been evaluated in terms of the PSNR index and the STD value. Experimental results provide evidence on the outstanding performance, accuracy, and convergence of the proposed algorithm in comparison to other methods. Although the results offer evidence to demonstrate that the standard HSA method can yield good results on complicated images, the aim of our paper is not to devise an MT algorithm that could beat all currently available methods, but to show that harmony search algorithms can be effectively considered as an attractive alternative for this purpose.

Acknowledgment

The proposed algorithm is part of the vision system used by a biped robot supported under the Grant CONACYT CB 181053. The first author acknowledges The National Council of Science and Technology of Mexico (CONACyT) for the doctoral Grant number 215517 and The Youth Institute of Jalisco (IJJ) for partially support this research.

References

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison Wesley, Reading, Mass, USA, 1992.
- [2] R. Guo and S. M. Pandit, "Automatic threshold selection based on histogram modes and a discriminant criterion," *Machine Vision and Applications*, vol. 10, no. 5-6, pp. 331-338, 1998.
- [3] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277-1294, 1993.
- [4] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Computer Vision, Graphics and Image Processing*, vol. 41, no. 2, pp. 233-260, 1988.
- [5] W. Snyder, G. Bilbro, A. Logenthiran, and S. Rajala, "Optimal thresholding: a new approach," *Pattern Recognition Letters*, vol. 11, no. 12, pp. 803-809, 1990.
- [6] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.
- [7] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, & Image Processing*, vol. 29, no. 3, pp. 273-285, 1985.
- [8] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition*, vol. 19, no. 1, pp. 41-47, 1986.
- [9] K. Hammouche, M. Diaf, and P. Siarry, "A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 676-688, 2010.
- [10] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.
- [11] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983.
- [12] F. Glover, "Tabu Search—part I," *ORSA Journal on Computing*, vol. 1, article 3, pp. 190-206, 1989.
- [13] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [14] C. Lai and D. Tseng, "A hybrid approach using Gaussian smoothing and genetic algorithm for multilevel thresholding," *International Journal of Hybrid Intelligent Systems*, vol. 1, pp. 143-152, 2004.
- [15] P.-Y. Yin, "A fast scheme for optimal thresholding using genetic algorithms," *Signal Processing*, vol. 72, no. 2, pp. 85-95, 1999.
- [16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, December 1995.
- [17] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, 2005.
- [18] B. Akay, "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding," *Applied Soft Computing*, vol. 13, no. 6, pp. 3066-3091, 2012.
- [19] P. D. Sathya and R. Kayalvizhi, "Optimal multilevel thresholding using bacterial foraging algorithm," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15549-15564, 2011.
- [20] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60-68, 2001.
- [21] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567-1579, 2007.
- [22] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36-38, pp. 3902-3933, 2005.
- [23] K. S. Lee, Z. W. Geem, S.-H. Lee, and K.-W. Bae, "The harmony search heuristic algorithm for discrete structural optimization," *Engineering Optimization*, vol. 37, no. 7, pp. 663-684, 2005.
- [24] J. H. Kim, Z. W. Geem, and E. S. Kim, "Parameter estimation of the nonlinear Muskingum model using harmony search," *Journal of the American Water Resources Association*, vol. 37, no. 5, pp. 1131-1138, 2001.
- [25] Z. W. Geem, "Optimal cost design of water distribution networks using harmony search," *Engineering Optimization*, vol. 38, no. 3, pp. 259-280, 2006.
- [26] Z. W. Geem, K. S. Lee, and Y. J. Park, "Application of harmony search to vehicle routing," *American Journal of Applied Sciences*, vol. 2, pp. 1552-1557, 2005.
- [27] A. Vasebi, M. Fesanghary, and S. M. T. Bathaee, "Combined heat and power economic dispatch by harmony search algorithm," *International Journal of Electrical Power and Energy Systems*, vol. 29, no. 10, pp. 713-719, 2007.
- [28] S. O. Degertekin, "Optimum design of steel frames using harmony search algorithm," *Structural and Multidisciplinary Optimization*, vol. 36, no. 4, pp. 393-401, 2008.
- [29] E. Cuevas, N. Ortega-Sánchez, D. Zaldivar, and M. Pérez-Cisneros, "Circle detection by harmony search optimization," *Journal of Intelligent and Robotic Systems*, pp. 1-18, 2011.
- [30] O. M. Alia and R. Mandava, "The variants of the harmony search algorithm: an overview," *Artificial Intelligence Review*, vol. 36, no. 1, pp. 49-68, 2011.

- [31] F. G. Lobo, C. F. Lima, and Z. Michalewicz, Eds., *Parameter Setting in Evolutionary Algorithms*, vol. 54 of *Studies in Computational Intelligence*, Springer, Berlin, Germany, 2007.
- [32] C. B. B. Costa, M. R. W. MacIel, and R. M. Filho, "Factorial design technique applied to genetic algorithm parameters in a batch cooling crystallization optimisation," *Computers and Chemical Engineering*, vol. 29, no. 10, pp. 2229–2241, 2005.
- [33] A. Khadwilard, P. Luangpaiboon, and P. Pongcharoen, "Full factorial experimental design for parameters selection of harmony search Algorithm," *The Journal of Industrial Technology*, vol. 8, no. 2, pp. 1–10, 2012.
- [34] G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistic for Experimenters—An Introduction to Design Data Analysis and Model Building*, John Wiley & Sons, New York, NY, USA, 1978.
- [35] S. K. Pal, D. Bhandari, and M. K. Kundu, "Genetic algorithms for optimal image enhancement," *Pattern Recognition Letters*, vol. 15, no. 3, pp. 261–271, 1994.
- [36] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, pp. 80–83, 1945.
- [37] S. Garcia, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special session on real parameter optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2008.