A Comparison of Stochastic Optimization Techniques for Image Segmentation

Suchendra M. Bhandarkar,[1,] * Hui Zhang[2,] †
[1] *Department of Computer Science, The University of Georgia, Athens, Georgia 30602*
[2] *DataStream Systems Inc., 50 DataStream Plaza, Greenville, South Carolina 29605*

Image segmentation denotes a process by which a raw input image is partitioned into nonoverlapping regions such that each region is homogeneous and the union of any two adjacent regions is heterogenous. A segmented image is considered to be the highest domain-independent abstraction of an input image. In this paper, the image segmentation problem is treated as one of combinatorial optimization. A cost function which incorporates both, edge information and region gray-scale variances is defined. The cost function is shown to be multivariate with several local minima. Three stochastic optimization techniques, namely, simulated annealing (SA), microcanonical annealing (MCA), and the random cost algorithm (RCA) are investigated and compared in the context of minimization of the aforementioned cost function for image segmentation. Experimental results on gray-scale images are presented.

I. INTRODUCTION

Image segmentation denotes a process by which a raw image is partitioned into nonoverlapping regions such that each region is connected and homogeneous and the union of no two spatially adjacent regions is homogenous. A segmented image is often considered to be the highest domain-independent or data-driven abstraction of an input image and constitutes the primary input to high-level vision which then utilizes domain-specific knowledge to interpret and analyze the image contents.[47] Each region in a segmented image needs to simultaneously satisfy the properties of homogeneity and connectivity. A region is considered homogeneous if all of its pixels satisfy a homogeneity criterion defined over one or more pixel attributes such as intensity, texture, color, range, etc. A region is considered connected if there exists a connected path between any two pixels within the region. More formally, if F is the set of all image

*Author to whom correspondence should be addressed; e-mail: suchi@cs.uga.edu.
†e-mail: zhanghu@dstm.com.
Contract grant sponsor: University of Georgia Research Foundation.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 15, 441–476 (2000)

pixels and $P(\cdot)$ is a homogeneity predicate defined over groups of connected pixels, then image segmentation is a partitioning of the set F into a set of connected subsets or regions $\{S_1, S_2, \ldots, S_n\}$ such that

$$\bigcup_{i=1}^{n} S_i = F \quad \text{with } S_i \cap S_j = \varnothing,\ i \neq j \tag{1}$$

The homogeneity predicate $P(\cdot)$ is such that

$$\begin{aligned} P(S_i) &= \textit{true} \text{ for all regions } S_i \text{ and} \\ P(S_i \cup S_j) &= \textit{false} \text{ for any two adjacent regions } S_i \text{ and } S_j \end{aligned} \tag{2}$$

Computing such an image partition is a problem of very high combinatorial complexity. Consequently, the research literature in computer vision and image processing is replete with several image segmentation techniques most of which have been developed with a certain class of images and a problem domain in mind. Development of a unified approach to image segmentation, however, remains an open problem.

Given the astronomical size of the search space, an exhaustive enumeration of all possible image partitions to arrive at a segmented image is practically infeasible. As a result, it is expedient to cast the segmentation procedure as one of combinatorial optimization by defining an objective function that embodies the homogeneity predicate. The desired segmented image is then considered to be one that globally optimizes the objective function. The objective function is typically multivariate with a landscape that is fraught with several local optima. This renders deterministic optimization techniques such as local hill-climbing search and gradient descent, ineffective since they are prone to get trapped in a local optimum. Stochastic optimization techniques, on the other hand, are capable of forgoing the local optima in the solution space in favor of a global optimum making them more suitable for image segmentation via combinatorial optimization.

II. STOCHASTIC OPTIMIZATION—A BRIEF OVERVIEW

Stochastic optimization algorithms such as simulated annealing (SA),[1] microcanonical annealing (MCA),[2] and the random cost algorithm (RCA)[3,4] are a subcategory of stochastic hill-climbing search techniques and are characterized by their capacity to escape from local optima in the objective function. A single iteration of a stochastic optimization algorithm consists of three phases: (i) perturb, (ii) evaluate, and (iii) decide. In the perturb phase, the current solution $\mathbf{x}_i$ to a multivariate objective function $E(\mathbf{x})$, which is to be minimized, is systematically perturbed to yield another candidate solution $\mathbf{x}_j$. In the evaluate phase, $E(\mathbf{x}_j)$ is computed. In the decide phase, $\mathbf{x}_j$ is accepted and replaces $\mathbf{x}_i$ *probabilistically* using a stochastic decision function. The stochastic decision function is *annealed* in a manner such that the search process resembles a

random search in the earlier stages and a greedy local search or a deterministic hill-climbing search in the latter stages. The major differences between SA, MCA, and the RCA arise from the differences in the stochastic decision function used in the decision phase. However, their common feature is that, starting from an initial solution, they generate, in the limit, an ergodic Markov chain of solution states which asymptotically converges to a stationary Boltzmann distribution.[5] The Boltzmann distribution asymptotically converges to a globally optimal solution when subject to the annealing process.[6]

A. Simulated Annealing

In the decide phase of SA, the new candidate solution $\mathbf{x}_j$ is accepted with probability p which is computed using the Metropolis function,[7]

$$p = \begin{cases} 1 & \text{if } E(\mathbf{x}_j) < E(\mathbf{x}_i) \\ \exp\left(- \dfrac{E(\mathbf{x}_j) - E(\mathbf{x}_i)}{T} \right) & \text{if } E(\mathbf{x}_j) \geq E(\mathbf{x}_i) \end{cases} \tag{3}$$

or using the Boltzmann function $B(T)$,[5]

$$p = B(T) = \frac{1}{1 + \exp\left(\left[E(\mathbf{x}_j) - E(\mathbf{x}_i) \right] / T \right)} \tag{4}$$

at a given value of temperature T, whereas $\mathbf{x}_i$ is retained with probability $(1 - p)$.

The Metropolis function and the Boltzmann function give SA the capability of *probabilistically* accepting new candidate solutions that are locally suboptimal compared to the current solution thus enabling it to climb out of local minima. Several iterations of SA are carried out for a given value of T which is then systematically reduced using an annealing function. The iterations carried out for a single value of T are referred to as an *annealing step*. As can be seen from Eqs. (3) and (4), at sufficiently high temperatures, SA resembles a completely random search whereas at lower temperatures it acquires the characteristics of a deterministic hill-climbing search or local greedy search.

Both the Metropolis function and the Boltzmann function ensure that SA generates an asymptotically ergodic (and hence stationary) Markov chain of solution states at a given temperature value. Geman and Geman[6] have shown that logarithmic annealing schedules of the form $T_k = R/\log k$ for some value of $R > 0$ are asymptotically good, i.e., they ensure asymptotic convergence to a global minimum with unit probability in the limit $k \to \infty$.

B. Microcanonical Annealing

MCA models a physical system whose total energy, i.e., sum of kinetic energy and potential energy, is always conserved. The potential energy of the system is the multivariate objective function $E(\mathbf{x})$ to be minimized whereas the

kinetic energy $E_k > 0$ is represented by a demon or a collection of demons. In the latter case, the total kinetic energy is the sum of all the demon energies. The demon energy (or energies) serve(s) to provide the system with an extra degree (or degrees) of freedom enabling MCA to escape from local minima.

In the decide phase of MCA, if $E(\mathbf{x}_j) < E(\mathbf{x}_i)$ then $\mathbf{x}_j$ is accepted as the new solution. If $E(\mathbf{x}_j) \geq E(\mathbf{x}_i)$ then $\mathbf{x}_j$ is accepted as the new solution only if $E_k \geq E(\mathbf{x}_j) - E(\mathbf{x}_i)$. If $E(\mathbf{x}_j) \geq E(\mathbf{x}_i)$ and $E_k < E(\mathbf{x}_j) - E(\mathbf{x}_i)$ then the current solution $\mathbf{x}_i$ is retained. In the event that $\mathbf{x}_j$ is accepted as the new solution, the kinetic energy demon is updated $E_k^{n+1} = E_k^n + [E(\mathbf{x}_i) - E(\mathbf{x}_j)]$ to ensure the conservation of the total energy. The kinetic energy parameter E_k is annealed in a manner similar to the temperature parameter T in SA. MCA can also be shown to converge to a global minimum with unit probability given a logarithmic annealing schedule.[8]

C. Random Cost Algorithm

In the RCA, a random noise variable $\mathbf{U}$ with a uniform distribution in the range $[-\delta, +\delta]$ (where $\delta > 0$ is the noise amplitude) takes on the role of the temperature parameter T in SA and the kinetic energy parameter E_k in MCA. In the decide phase of the RCA, a new candidate solution $\mathbf{x}_j$ replaces the current solution $\mathbf{x}_i$ if $[E(\mathbf{x}_j) - E(\mathbf{x}_i)] + \mathbf{U} < 0$. Since the value of $\mathbf{U}$ could be either positive or negative, it means that the RCA could forgo, in certain cases, moves toward the local minimum [when $\mathbf{U} > -[E(\mathbf{x}_j) - E(\mathbf{x}_i)]$] while accepting moves that steer it away from a local minimum [when $\mathbf{U} < -[E(\mathbf{x}_j) - E(\mathbf{x}_i)]$]. The noise amplitude δ, which determines the bounds of the distribution of $\mathbf{U}$, is annealed in a manner similar to temperature parameter T in SA and the kinetic energy parameter E_k in MCA. The RCA can also be shown to converge to a global minimum with unit probability given a logarithm annealing schedule.[3]

III. IMAGE SEGMENTATION—A BRIEF REVIEW

Several image segmentation techniques that have been proposed and described in the computer vision and image processing research literature. This proliferation is in part due to the fact that there exist several problem domains and applications that need to process and interpret image data in a domain-specific or application-specific manner. Moreover, depending on the problem domain or application, there are several types of images that could be processed and analyzed (gray-scale, range, MRI, X-ray, thermal, etc.). The exhaustive nature of the topic, makes it impossible to review each individual image segmentation technique in the literature. Fu and Mui,[9] Haralick and Shapiro,[10] Pal and Pal,[11] and Sahoo et al.[12] have surveyed and categorized several image segmentation techniques. Most existing image segmentation techniques can be broadly classified as based on:

(i) Gray level thresholding which include local, global, deterministic, fuzzy, and stochastic thresholding schemes.[13–17]

(ii) Iterative pixel classification which includes deterministic and stochastic relaxation.[18–20]
(iii) Markov random field models.[21–26]
(iv) Neural networks.[27–31]
(v) Surface fitting, surface classification and surface–region growing.[32,33]
(vi) Parameter space clustering which includes crisp, probabilistic, and fuzzy clustering.[34,35]
(vii) Edge detection.[36–38]
(viii) Cost function optimization.[39–41]

The segmentation technique considered in this paper is based on cost function optimization. Our previous experience[39] and that of Tan, Gelfand, and Delp[40,41] in using the cost function optimization approach to edge detection showed this technique to be both broad in its applicability and capable of producing good results. This provided the motivation for our considering the cost function optimization approach to image segmentation. In this paper, we consider conventional light intensity (i.e., gray-scale) images but the approach is general enough to be extensible to other types of images as well.

IV. IMAGE SEGMENTATION VIA COST FUNCTION MINIMIZATION

The cost function optimization approach taken in this paper consists of two phases: (i) the preprocessing phase and (ii) the cost minimization phase. Given a raw intensity image I, the first step in our segmentation procedure is dissimilarity enhancement where the pixels in the image that are likely candidates for edge points are selective enhanced. Next, the enhanced image is subject to an iterative thresholding and connected component labeling process to obtain an initial assessment of region labels to pixels. As a result, an initial segmented image is generated. The initial segmented image is then subject to a boundary tracing algorithm which finds all the edge pixels to obtain an initial edge configuration. A region adjacency graph (RAG) is generated from the initial segmented image. The RAG nodes represent the regions in the segmented image and an RAG arc between two nodes represents the edge contour separating the corresponding regions. The RAG is assigned a cost which is computed using the cost function defined in Section IV.B, and then subject to a minimization procedure. The RAG is manipulated and modified during the minimization procedure. When the minimization procedure is deemed to have converged to a global minimum of the cost function, the segmented image corresponding to the RAG is considered to be the desired segmented image.

A. Preprocessing Phase

The preprocessing phase includes image dissimilarity enhancement, thresholding, connected component labeling, edge tracing, RAG generation, and initial cost evaluation. Edge pixels in an image separate regions that are dissimilar. During dissimilarity enhancement, the potential edge pixels are enhanced, i.e., large values are assigned to those pixels in the image that possess

this property. The enhanced image $D = \{d(i, j);\ 1 \le i,\ j \le N\}$ is a collection of pixels where each pixel value is proportional to the degree of region dissimilarity that exists at that pixel site. The pixel values in D lie in the range $[0, 1]$. Pixels with values close to 1 are potential edge points. The dissimilarity enhancement algorithm is similar to that described in Refs. 39–41.

The enhanced image D is subject to an iterative thresholding scheme, followed by connected component labeling. The iterative thresholding scheme results in a binary image where pixels are classified as either foreground or background pixels. The connected component labeling algorithm results in the formation of numerous connected regions in the image. The initial edge (boundary) pixels for each of these regions are then determined using the following algorithm

(1) Raster scan the image starting from the first pixel $p \in$ region S (the foreground).
(2) Let n_i; $1 \le i \le 4$, be the 4-neighbors of pixel p.
(3) If any of n_i; $1 \le i \le 4$, $\notin$ region S, then p is an edge pixel (i.e., on the boundary of S), else p is an interior pixel.
(4) If p is the last pixel in region S then go to (5) else p = next pixel in the raster scan. Go to step (2).
(5) Stop.

After the RAG is generated, the cost of the initial segmented image is computed using the cost function described in the following subsection.

B. Cost Function Evaluation

The cost function for image segmentation is defined as

$$E = w_1\left(\sum_{i=1}^{R} \sigma_i^2\right) + w_2 C_e \tag{5}$$

where σ_i^2 is the gray level variance of region i, R is the total number of regions, and C_e is the cost associated with the resulting edge configuration. The edge cost function C_e is similar to the one used by Bhandarkar, Zhang, and Potter[39] and Tan, Gelfand, and Delp[40,41] in their work on edge detection.

The edge cost function $C_{i,j}$ at each pixel site (i, j) is a weighted sum of the following terms:

(1) c_1 which is a measure of local region dissimilarity. This term tends to place edge pixels at sites of high region dissimilarity and penalizes pixel sites with little gray level variation.
(2) c_2 which penalizes pixel sites that result in thick edges while favoring pixel sites that result in thin edges, i.e., edges that are one pixel thick.
(3) c_3 which penalizes nonendpoint edge pixels based on local measure of curvature. This term tends to smooth out or remove highly kinky edges.
(4) c_4 which penalizes nonendpoint edge pixels that correspond to fragmented edges. This term links up or removes fragmented edges.

(5) c_5 which assigns a unit cost to each edge pixel, thereby preventing an excessive number of edge pixels from being detected.

The edge cost function at an individual pixel site (i, j) is given by

$$C_{i,j} = \sum_{k=1}^{5} w_k c_k \tag{6}$$

where the w_is are predetermined weights assigned to the respective terms. The edge cost function for an entire image of size $M \times N$ pixels is given by

$$C_e = \sum_{i=1}^{M} \sum_{j=1}^{N} C_{i,j} \tag{7}$$

The cost function associated with region assignment is defined as the sum of the gray level variances of all the resulting regions. The total cost associated with the image segmentation is the weighted sum of the edge configuration cost and the region assignment cost.

The initial segmented image is then subject to a cost minimization procedure. All the stochastic optimization algorithms considered in this paper, i.e., SA, MCA, and the RCA, entail an appropriate and computationally efficient state transition or solution perturbation strategy. The following state transition strategy was implemented:

(1) Let p be the current pixel, starting with the first pixel.
(2) Let n_i; $1 \le i \le 8$, be the 8-neighbors of pixel p.
(3) Generate a random number k between 1 and 8.
(4) If region_label(p) = region_label(n_k), go to (6) else let region_label(p) = region_label(n_k), i.e., assign p to the neighboring region, update the RAG, go to (5).
(5) Since region_label(p) is changed, update its edge configuration cost and that of its 8-neighboring pixels n_i; $1 \le i \le 8$.
(6) Let p = next pixel in raster scan order, go to (2).
(7) Stop.

After a new candidate state is generated, it would be necessary to recompute the cost for the new candidate state. Since it is computationally inefficient to recalculate the cost function *from scratch* for every new candidate state, the cost functions associated with the edge configuration and region assignment are *incrementally computed* for each state transition. Since the edge configuration cost function is computed *locally* at each pixel site, it is amenable to incremental computation.[39–41] To render the region assignment function suitable for incremental computation, the following information is maintained in each region node of the RAG. The number of pixels in each region: n, the sum of the gray-scale values of the pixels in the region: $\sum_i g_i$, the sum of the squares of the gray level values of the pixels in the region: $\sum_i g_i^2$, and the square of the sum of the gray level values of the pixels in the region: $(\sum_i g_i)^2$. After each transition,

the RAG is updated and the mean μ and the variance σ^2 of each region involved in the state transition are incremented as

$$\Delta\mu = \frac{1}{(n+\Delta n)}\Delta\left(\sum_i g_i\right) - \frac{\Delta n}{n(n+\Delta n)}\left(\sum_i g_i\right) \tag{8}$$

$$\Delta\sigma^2 = \frac{1}{(n+\Delta n)}\Delta\left(\sum_i g_i^2\right) - \frac{1}{(n+\Delta n)^2}\left[\Delta\left(\sum_i g_i\right)\right]^2 \tag{9}$$

Equations (8) and (9) are used in computing the change ΔE in the cost function E defined in Eq. (5).

C. Cost Function Minimization Using SA

SA starts with the initial segmented image at a high temperature value and successively generates a series of candidate regions and edge configurations using the state transition strategy proposed in Section IV.B. Each state transition is associated with a change ΔE in the total cost which is computed using Eqs. (5), (8), and (9). The state transition is accepted deterministically if $\Delta E < 0$ and probabilistically using the Metropolis function [Eq. (3)] if $\Delta E \geq 0$.

The initial temperature T_i is chosen high enough so that most of the state transitions are accepted and there is little chance of the algorithm being trapped in a local minimum in the early stages. A common rule of thumb, which has been followed in this paper, is that T_i should be chosen such that 90% of the state transitions are accepted.

An annealing or cooling scheme is required for reducing the temperature over the course of the algorithm. We use a geometric annealing schedule of the form $T_{n+1} = \alpha T_n$ where $\alpha < 1$. To enable convergence, α has to be close to 1; a value of $\alpha = 0.95$ has worked well in our experiments. Although the aforementioned geometric annealing schedule does not strictly ensure asymptotic convergence to a global optimum as the logarithmic annealing schedule does, it is much faster and has been found to yield very good solutions in practice.[42] The number of solution perturbations at a given temperature has to be chosen large enough to ensure the asymptotic stationarity of the Markov chain of solution states at each value of T. We have found raster scanning the image four times at each value of T, resulting in $4MN$ solution perturbations for an $M \times N$ image, to give good results.

The SA-based cost minimization procedure can be described as follows:

(1) Start with an initial region assignment and edge configuration with an initial cost of E_0.
(2) Fix the initial temperature T_i such that 90% of the state transitions are accepted.
(3) Starting with the first pixel, check if it is an edge pixel, if not, go to step (6), if yes, make a local change in the region assignment and edge configuration at each pixel site in accordance with the transition strategy described in Section IV.B.

(4) Evaluate the resulting change in cost function ΔE.
(5) Accept or reject the transition to the new edge and region configuration depending on the values of ΔE and the Metropolis function [Eq. (3)].
(6) Repeat steps (3)–(5) for the next pixel in raster scan order.
(7) Repeat steps (3)–(6) for four raster scans of the image.
(8) Check for convergence. If yes, go to step (10).
(9) Update the temperature according using the geometric annealing schedule, go to step 3.
(10) Stop.

The convergence condition typically is the fact that the cost function has not changed for the past k successive T values.

D. Cost Function Minimization Using MCA

MCA starts with a high value of E_k, that is, the kinetic energy demon is capable of changing most (90% in our case) of the pixel states in the early annealing stages. In the standard version of MCA, a single kinetic energy demon travels around the entire system and the decision to accept or reject a transition depends on the current value of E_k. We found that this basic version of MCA was inefficient and did not exhibit good convergence properties in the context of image segmentation. Instead, we use a lattice of kinetic energy demons with a demon at every pixel site. Thus, we have MN demons for an image of size $M \times N$ pixels. The total kinetic energy E_k is the sum of all the demon energies. Initially, the total kinetic energy E_k is evenly distributed in the demon lattice, i.e., each demon has the same initial kinetic energy. During a raster scan of the image, each demon tries to change the state of its associated pixel. After a complete raster scan, the demon lattice is subject to a random permutation. However, due to the large size of the image, we only permute the demons in a 3×3 neighborhood at each demon site after it has been visited.

The demon energies are annealed after each iteration, resulting in a reduction in the total system energy. As in the case of SA, a geometric annealing schedule of the form $E_k^{n+1}(i, j) = \alpha E_k^n(i, j)$ (where $\alpha = 0.95$) was used in our implementation. In MCA, the system evolves to shrink to a set of low energy states. Analogous to SA, the MCA algorithm is considered to have converged if the cost function has not changed over the past k successive annealing steps. The outline of the MCA algorithm is as follows:

(1) Begin with an initial region and edge configuration with an initial cost E_0.
(2) Initialize the kinetic energy demon lattice $E_k(i, j)$ for all i, j such that 90% of the state transitions are accepted.
(3) Starting with the first pixel, check if it is an edge pixel. If not, go to step (7); if yes, make a local change in the region assignment and edge configuration at the pixel site in accordance with the transition strategy described in Section IV.B.
(4) Evaluate the resulting change in cost ΔE.
(5) Accept the transition to the new edge configuration based on the value of $E_k(i, j)$ and the state transition criterion described in Section II.B.
(6) Perform a random local permutation of the demons.

(7) Repeat steps (3)–(6) for the next pixel in raster scan order.
(8) Repeat steps (3)–(7) for four raster scans of the image.
(9) Check for convergence, if yes go to step (11).
(10) Update the demon energies according to a geometric annealing schedule: for all i, j, $E_k^{n+1}(i, j) = \alpha E_k^{n+1}(i, j)$. Go to step (3).
(11) Stop.

E. Cost Function Minimization Using the RCA

The random noise parameter in the RCA is uniformly distributed in the range $[-\delta, +\delta]$ with zero mean. The RCA starts with a high value for δ which is annealed using a geometric schedule of the form $\delta_{n+1} = \alpha\delta_n$ where $\alpha = 0.95$. Analogous to SA and MCA, the RCA is considered to have converged if the cost function has not changed over the past k successive annealing steps. The outline of the RCA for image segmentation is as follows:

(1) Begin with an initial region and edge configuration with an initial cost of E_0.
(2) Initialize the noise amplitude parameter δ such that 90% of the state transitions are accepted.
(3) Starting from the first pixel, check if is an edge pixel. If not, go to step (6), if yes, make a local change in the region assignment and edge configuration at the pixel state in accordance with the transition strategy described in Section IV.B.
(4) Evaluate the resulting change in cost function ΔE.
(5) Accept the transition according to the criterion described in Section II.C.
(6) Repeat steps (3)–(5) for the next pixel in raster scan order.
(7) Repeat steps (3)–(6) for four raster scans of the image.
(8) Check for convergence. If yes, go to step (10).
(9) Update the noise parameter δ according to a geometric annealing schedule: $\delta_{n+1} = \alpha\delta_n$. Go to Step (3).
(10) Stop.

V. EXPERIMENTAL RESULTS

All three stochastic optimization algorithms, i.e., SA, MCA, and the RCA were implemented and tested on several gray-scale images. In this section, we present and compare our results obtained from each of these algorithms on four different images shown in Figures 1–4 and referred to as *Boeing*, *Computer*, *CRT*, and *Phone*, respectively. To test the segmentation algorithms in the presence of noise, each of these images was corrupted with zero mean Gaussian noise with varying variance (σ_n^2) values. For example, Figures 5 and 6 show the *Boeing* and *CRT* images each corrupted with zero mean Gaussian noise with variance $\sigma_n^2 = 100$. All the original and noisy images were subject to the preprocessing phase described in Section IV.A and initial cost evaluation. Table I summarizes the results obtained from the preprocessing phase. Figures 7 and 8 show the result of preprocessing on the *Boeing* and the *CRT* images, respectively.

Figure 1. The *Boeing* image.

The preprocessed images were then subject to cost minimization using SA, MCA, and the RCA. The parameter settings such as the initial temperature, initial noise amplitude, initial demon energy value, annealing rate, annealing schedule, and the stopping criterion were determined experimentally and are summarized in Table II. We found that the standard convergence criterion, i.e.,

Figure 2. The *CRT* image.

Figure 3. The *Computer* image.

Figure 4. The *Phone* image.

Figure 5. The *Boeing* image with additive Gaussian noise, $\sigma_n^2 = 100$.

the fact that the cost function has not changed over the past k successive annealing steps resulted in excessive run times for SA, MCA, and the RCA. Consequently, the convergence criterion was modified whereby during the course of annealing we kept track of the maximum and minimum cost ($E_{\max}$ and $E_{\min}$) for the iterations at each annealing step, i.e., at a given value of tempera-

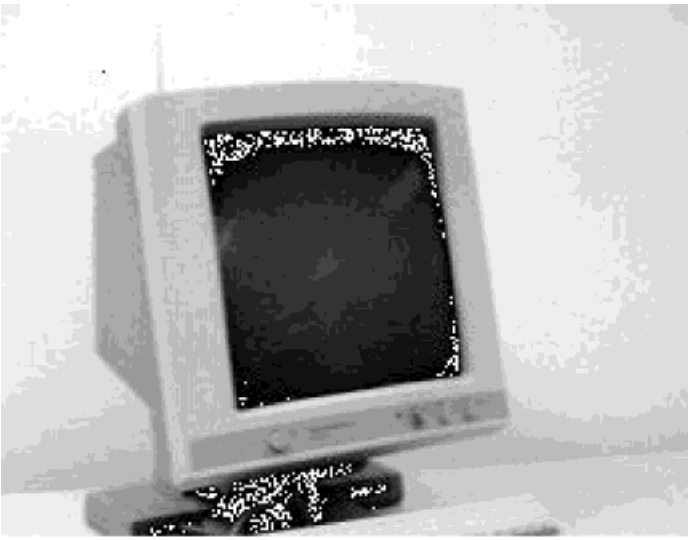

Figure 6. The *CRT* image with additive Gaussian noise, $\sigma_n^2 = 100$.

Table I. Results obtained from the preprocessing phase

Image	Initial Regions	Initial Edge Cost	Initial Region Cost	Total Cost
Boeing	200	170,063.00	15,810.40	185,873.40
Boeing, $\sigma_n^2 = 25$	209	175,864.00	46,241.60	22,105.60
Boeing, $\sigma_n^2 = 100$	272	204,561.00	390,031.00	594,592.00
CRT	70	97,901.40	9619.91	107,521.31
CRT, $\sigma_n^2 = 25$	158	106,514.00	352,684.00	459,198.00
CRT, $\sigma_n^2 = 100$	193	130,599.00	561,472.00	692,071.00
Computer	478	245,351.00	97,080.6	342,431.60
Phone	277	253,302.00	16,048.10	269,350.10

ture, kinetic energy demon value, and noise amplitude value for SA, MCA, and the RCA, respectively. The annealing process was halted whenever the difference $E_{\max} - E_{\min}$ was less than a prespecified threshold value for $k = 5$ successive annealing steps.

The final segmentation results for the gray-scale images are shown in Figures 9–32. Each region in the segmented image is shown with a different gray level value and edges are shown in white. The convergence characteristics of the three algorithms on the noise-free and noisy *Boeing* and *CRT* images are shown in Figures 33–36.

Table III compares the results of each SA, MCA, and the RCA on the noise-free and noisy gray-scale images with varying noise variance values. All the

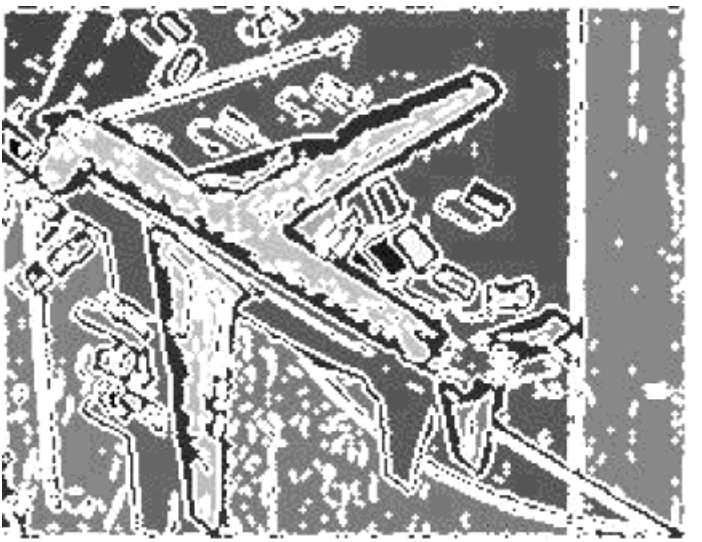

Figure 7. The preprocessed *Boeing* image.

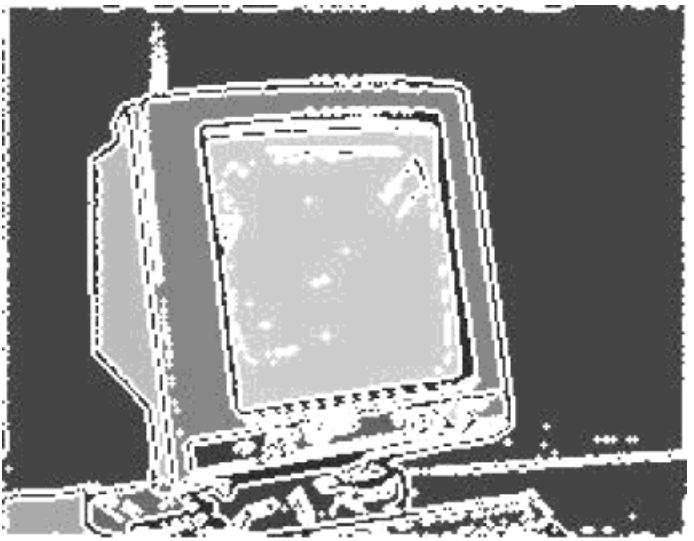

Figure 8. The preprocessed *CRT* image.

three algorithms were run on a 110 MHz SUN SPARC5 workstation with 64MB RAM. The performance parameters in Table III for each of the algorithms (i.e., execution times, final cost, and number of annealing steps) represent average values over 40 runs of that algorithm. Note, that in a single annealing step of SA, MCA, and the RCA, $4MN$ state transitions or perturbations are attempted for an $M \times N$ image. The values of the performance parameters arising from 40 runs of each algorithm could be looked upon as a random sample of size 40 drawn from a population with unknown distribution.

The central limit theorem[43] states that for a sample $X_1, X_2, \ldots, X_{N_s}$ of size N_s, where N_s is large, the distribution of the sample mean $\overline{X} = 1/N_s \sum_{i=1}^{N_s} X_i$ is approximately normal with mean μ and variance σ^2/N_s where μ and σ^2 are the mean and variance of the population distribution, respectively. In other words, the distribution of the random variable $Z = (\overline{X} - \mu)/(\sigma/\sqrt{N_s})$ approaches $\mathcal{N}(0,1)$ in the limit $N_s \to \infty$ regardless of the distribution of the underlying population. For values of $N_s \geq 30$, the distribution of $\overline{X}$ could

Table II. Parameter settings for SA, MCA, and the RCA

Algorithm	T_0, δ_0, E_k^0	Initial Acceptance Ratio	Annealing Schedule	Stopping Criterion
SA	20.1	0.9	$T_{n+1} = 0.95 \times T_n$	$E_{\max} - E_{\min} < 0.05$
MCA	15.2	0.9	$E_k^{n+1} = 0.95 \times E_k^n$	$E_{\max} - E_{\min} < 0.05$
RCA	14.3	0.9	$\delta_{n+1} = 0.95 \times \delta_n$	$E_{\max} - E_{\min} < 0.05$

Figure 9. The segmented *Boeing* image resulting from SA.

be considered to be normal for most practical applications.[43] In this case, the 95% confidence interval for the mean $\bar{X}$ is given by $[\bar{X} - 1.96\sigma/\sqrt{N_s},\ \bar{X} + 1.96\sigma/\sqrt{N_s}]$. When $N_s \geq 30$, the 95% confidence interval could be approximated by $[\bar{X} - 1.96\sigma_s/\sqrt{N_s},\ \bar{X} + 1.96\sigma_s/\sqrt{N_s}]$ where σ_s^2 is the sample variance.[43] The 95% confidence interval implies that all possible null hypotheses

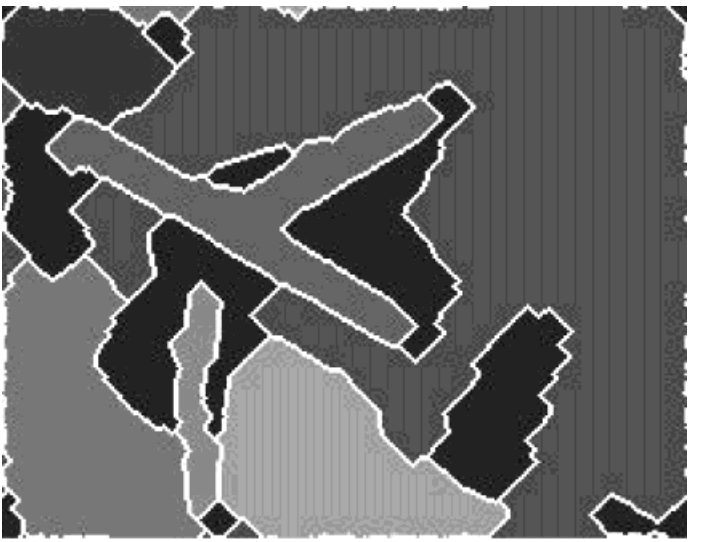

Figure 10. The segmented *Boeing* image resulting from MCA.

Figure 11. The segmented *Boeing* image resulting from the RCA.

Figure 12. The segmented *Computer* image resulting from SA.

Figure 13. The segmented *Computer* image resulting from MCA.

Figure 14. The segmented *Computer* image resulting from the RCA.

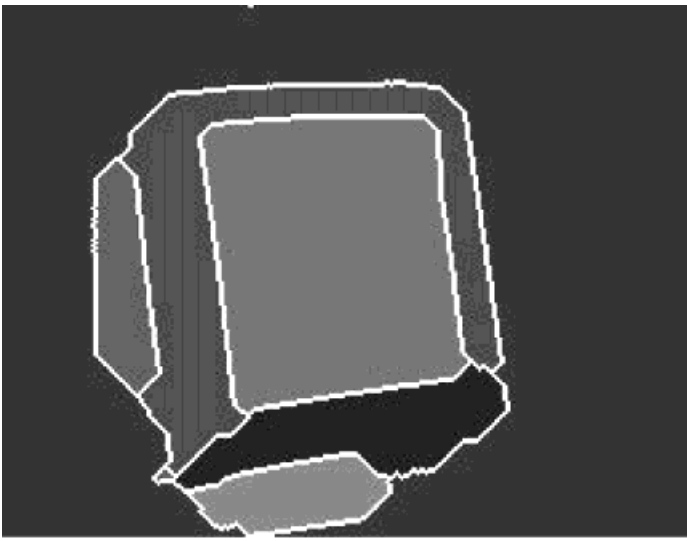

Figure 15. The segmented *CRT* image resulting from SA.

$\mu = \mu_0$ are rejected at a level of significance 0.05 if μ_0 lies outside the interval. In Table III the confidence interval is expressed as $\overline{X} \pm 1.96\sigma_s/\sqrt{N_s}$ for each of the performance parameters (where $N_s = 40$ in our case).

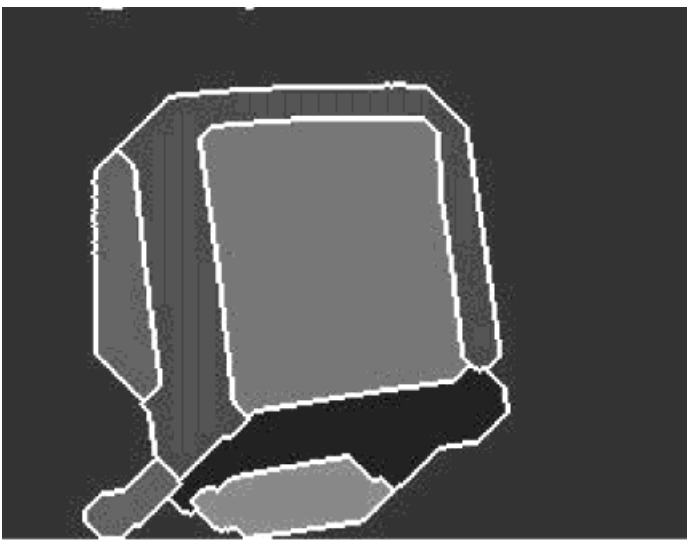

Figure 16. The segmented *CRT* image resulting from MCA.

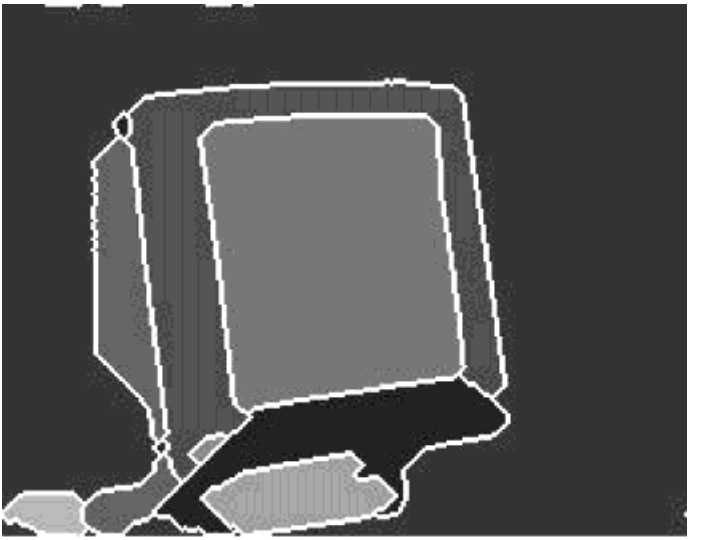

Figure 17. The segmented *CRT* image resulting from the RCA.

Figure 18. The segmented *Phone* image resulting from SA.

Figure 19. The segmented *Phone* image resulting from MCA.

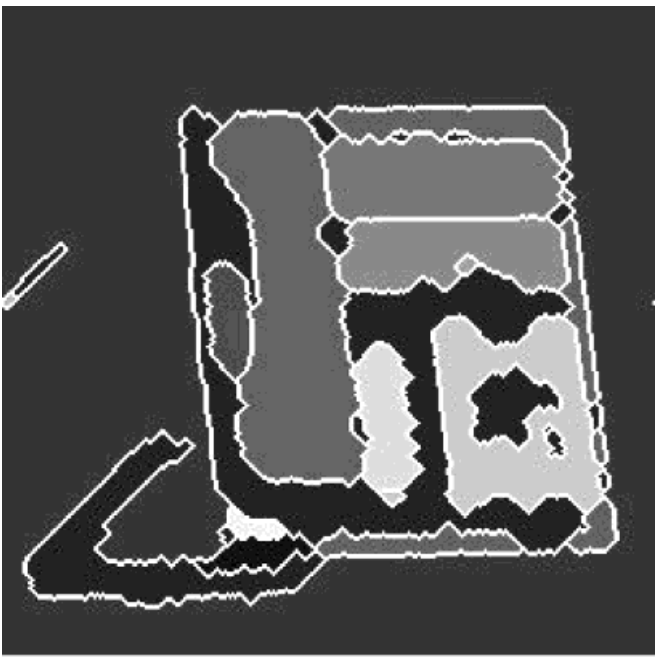

Figure 20. The segmented *Phone* image resulting from the RCA.

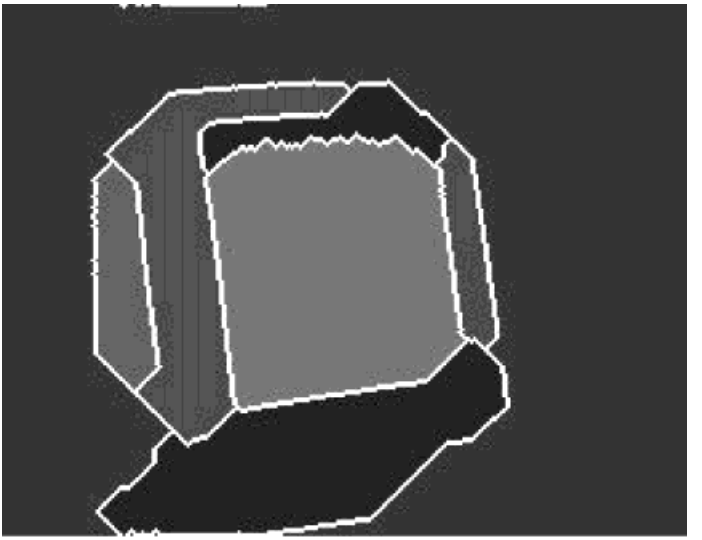

Figure 21. The segmented noisy *CRT* image ($\sigma_n^2 = 25$) resulting from SA.

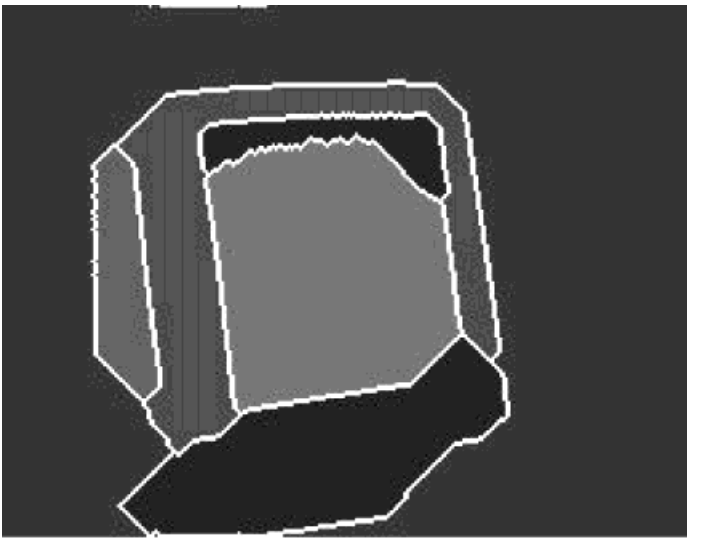

Figure 22. The segmented noisy *CRT* image ($\sigma_n^2 = 25$) resulting from MCA.

Figure 23. The segmented noisy *CRT* image ($\sigma_n^2 = 25$) resulting from the RCA.

Figure 24. The segmented noisy *CRT* image ($\sigma_n^2 = 100$) resulting from SA.

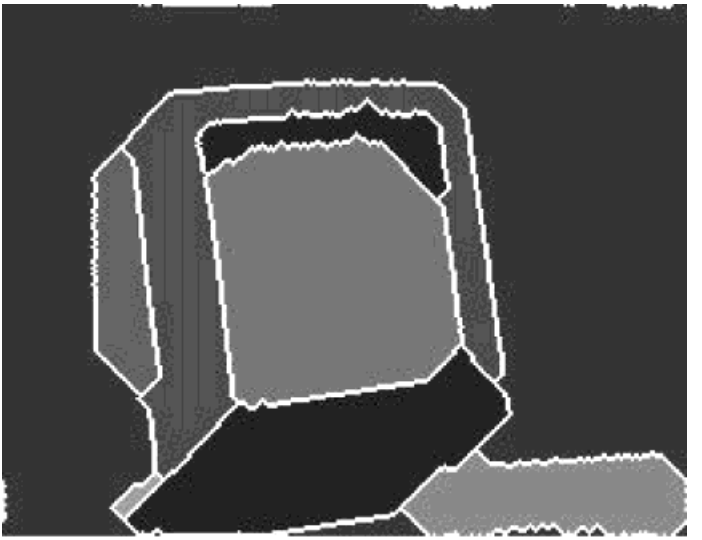

Figure 25. The segmented noisy *CRT* image ($\sigma_n^2 = 100$) resulting from MCA.

Figure 26. The segmented noisy *CRT* image ($\sigma_n^2 = 100$) resulting from the RCA.

Figure 27. The segmented noisy *Boeing* image ($\sigma_n^2 = 25$) resulting from SA.

Figure 28. The segmented noisy *Boeing* image ($\sigma_n^2 = 25$) resulting from MCA.

Figure 29. The segmented noisy *Boeing* image ($\sigma_n^2 = 25$) resulting from the RCA.

Figure 30. The segmented noisy *Boeing* image ($\sigma_n^2 = 100$) resulting from SA.

Figure 31. The segmented noisy *Boeing* image ($\sigma_n^2 = 100$) resulting from MCA.

Figure 32. The segmented noisy *Boeing* image ($\sigma_n^2 = 100$) resulting from the RCA.

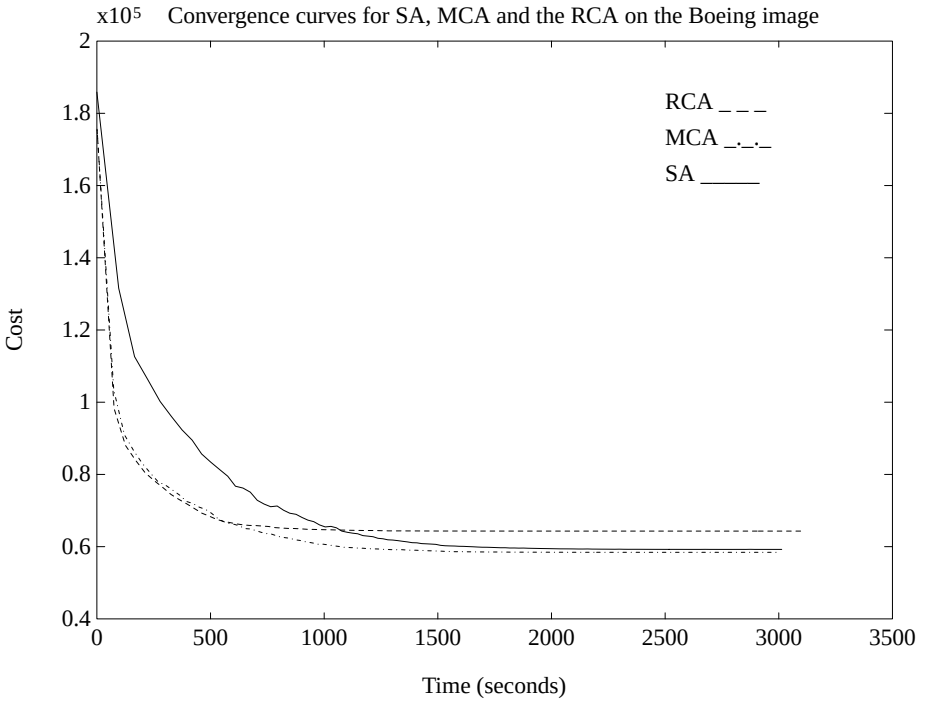

Figure 33. Convergence curves of SA, MCA, and the RCA for the *Boeing* image.

The segmentation results (Figs. 9–32), the convergence curves (Figs. 33–36), and the performance statistics (Table III) prompt us to make the following observations.

(1) On simple images such as the *CRT* image, all the three algorithms, SA, MCA, and the RCA, gave comparable results. On more complex images such as the *Computer* and the *Boeing* images, the differences between the three algorithms were more pronounced. Overall, the RCA was the fastest and SA the slowest in terms of CPU time. This can be attributed to the fact that SA entails the computation of an exponential function [Eqs. (3) and (4)] at each iteration whereas in the case of MCA and the RCA only simple addition or subtraction at each iteration is needed. MCA is slower than the RCA because of the overhead involved in updating the demon energies and perturbing the demon lattice.

(2) In terms of the quality (i.e., cost) of the final solution SA performed the best and the RCA the worst in all the cases. This implies that the Metropolis decision function [Eq. (3)] or the Boltzmann decision function [Eq. (4)] provides the SA with better stochastic hill-climbing ability compared to MCA or the RCA. On the whole, MCA was seen to offer a good compromise between the CPU time and the quality of the final solution.

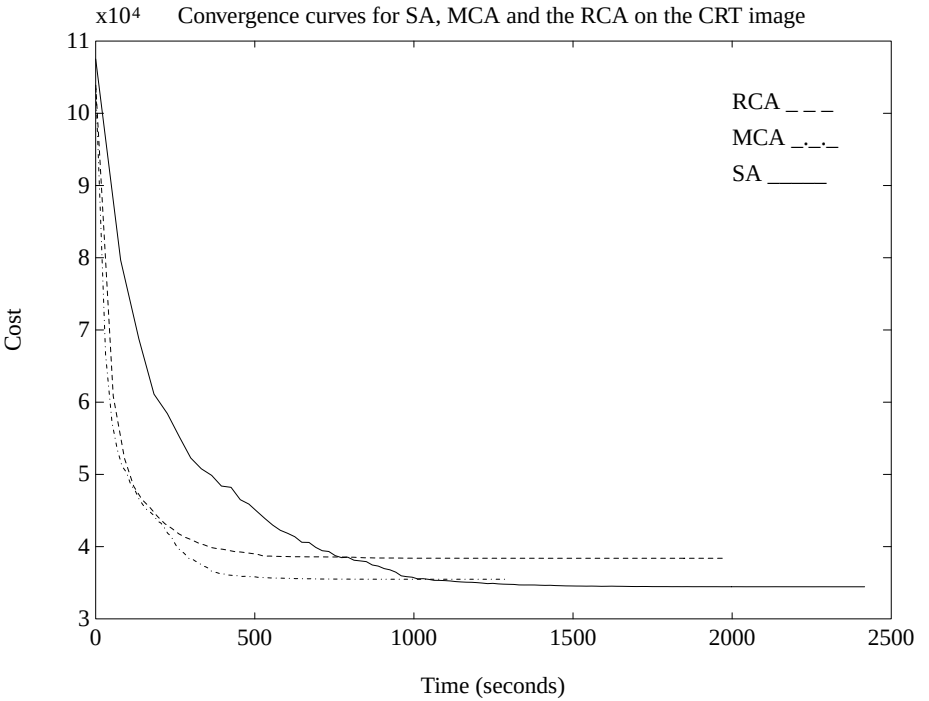

Figure 34. Convergence curves of SA, MCA, and the RCA for the *CRT* image.

(3) From Table III it can be seen that in terms of tightness of the error bounds on the performance parameters, SA performed the best whereas the RCA performed the worst. The tightness of the error bounds can be looked upon as a measure of robustness or predictability of performance. In this context, SA is the most robust of the three algorithms and the RCA the least. Once again, MCA was seen to offer a good compromise between the RCA and SA.

(4) In terms of noise tolerance, the three algorithms exhibit gradual degradation with increasing noise variance value in terms of both, the quality (cost) of the final solution and tightness of the error bounds associated with the performance parameters. The degradation is the most severe in the case of the RCA and the least in the case of SA. This shows that SA is more robust to noise that either MCA or the RCA with MCA offering a compromise between the RCA and SA.

For the sake of comparison, the results of the edge-based segmentation algorithm[41] on the gray-scale *Computer*2 image (Fig. 37) and its noisy version (Fig. 38) are presented in Figures 39 and 40. The *Computer*2 is similar to the *Computer* image except for a change in viewpoint and has the same level of detail. The segmentation algorithm uses SA to minimize the edge-based cost

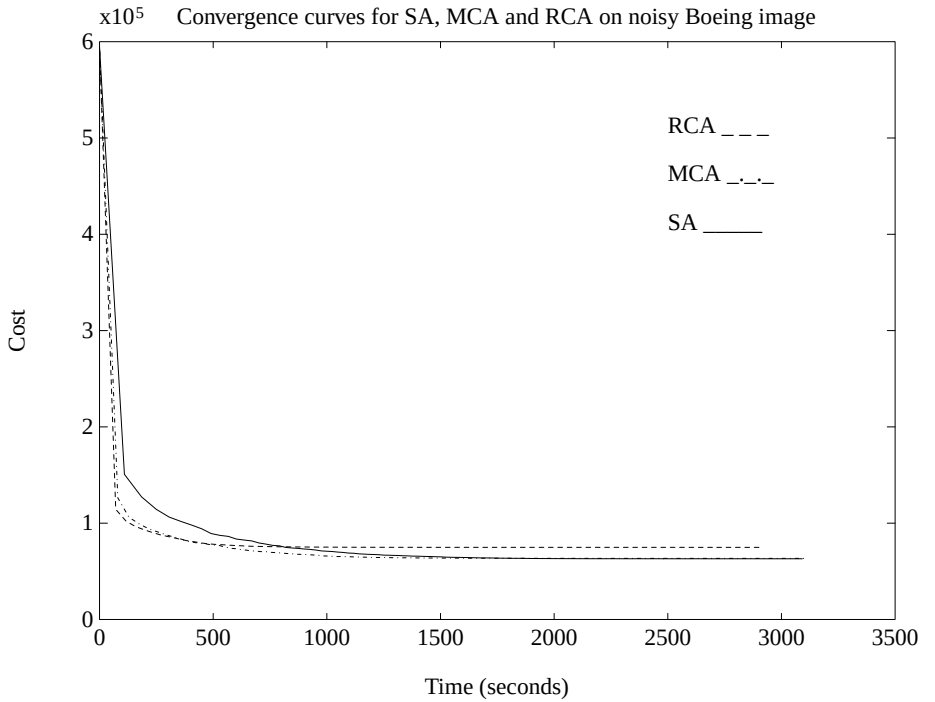

Figure 35. Convergence curves of SA, MCA, and the RCA for the noisy *Boeing* image, $\sigma_n^2 = 100$.

function C_e [Eq. (7)]. As can be seen from the results, a purely edge-based approach does not result in closed regions with well-defined contours due to the fragmentation of the edges. This shows the advantage of combining both, region and edge information in the cost function.

VI. CONCLUSIONS

In this paper, the problem of image segmentation was treated as one of combinatorial optimization. The goal of this paper was to compare the performance of stochastic optimization techniques, namely, simulated annealing (SA), microcanonical annealing (MCA), and the random cost algorithm (RCA), in the context of image segmentation. A cost function was defined for an input gray-scale image which incorporated both, edge information and region gray-scale variances. The cost function was shown to be multivariate with a solution

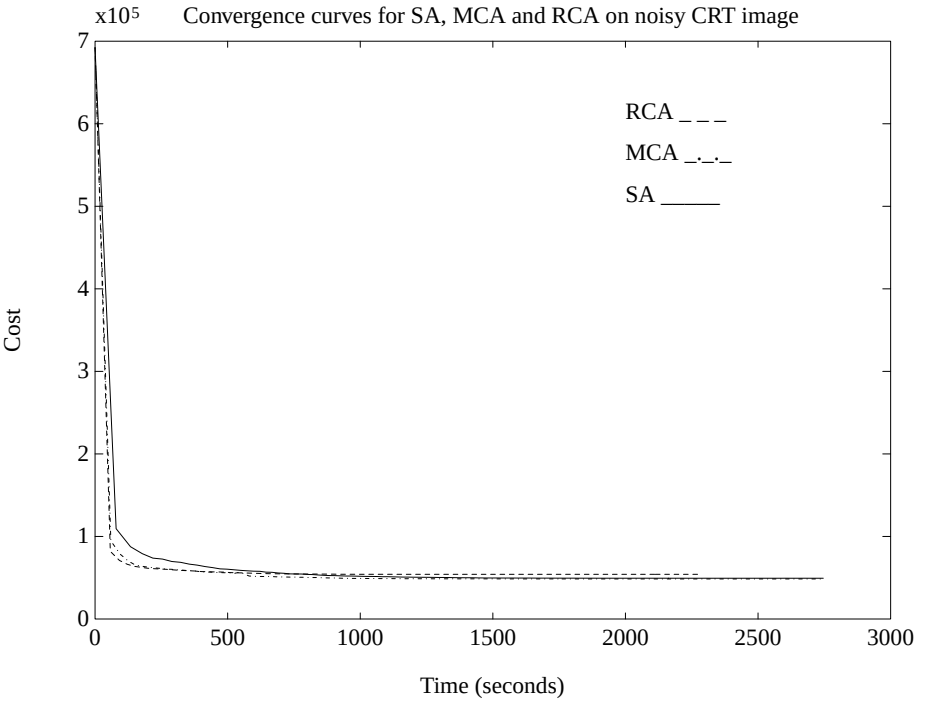

Figure 36. Convergence curves of SA, MCA, and the RCA for the noisy *CRT* image, $\sigma_n^2 = 100$.

landscape containing several local minima. Since deterministic combinatorial optimization techniques based on local search are vulnerable to the presence of local minima, stochastic optimization techniques were considered of which three, namely, SA, MCA, and the RCA, were investigated in this paper.

Experimental results on gray-scale images showed all the three techniques to perform well on noise-free images. The RCA was the fastest in terms of CPU time whereas SA was the slowest. However, the RCA exhibited the worst performance in terms of the quality (i.e., cost) of the final segmented image, tightness of the error bounds associated with the performance parameters and degradation of the segmentation quality and degradation of the tightness of the error bounds with increasing levels of additive Gaussian noise. SA exhibited the best performance in terms of the aforementioned criteria whereas MCA was seen to provide a good compromise between SA and the RCA.

Table III. Performance comparison of SA, MCA, and the RCA

Image	Algorithm	Final Cost	CPU Time (min)	Annealing Steps
CRT	MCA	35,479.43 ± 154.93	45.23 ± 1.27	149.57 ± 4.18
	RCA	38,390.35 ± 216.91	37.93 ± 1.78	152.77 ± 7.15
	SA	34,648.63 ± 109.44	53.54 ± 1.02	125.62 ± 2.38
Boeing	MCA	58,429.72 ± 251.37	51.27 ± 1.51	147.33 ± 4.34
	RCA	64,310.26 ± 360.14	45.73 ± 2.12	164.14 ± 7.63
	SA	53,832.73 ± 166.27	54.92 ± 1.06	125.17 ± 2.39
Computer	MCA	70,726.25 ± 297.13	49.43 ± 1.34	149.72 ± 4.07
	RCA	88,405.11 ± 483.55	43.12 ± 2.58	162.97 ± 9.76
	SA	68,602.33 ± 206.55	59.58 ± 1.17	128.13 ± 2.51
Phone	MCA	63,393.75 ± 272.17	51.72 ± 1.46	160.43 ± 4.48
	RCA	75,394.27 ± 421.63	47.63 ± 2.37	185.33 ± 9.21
	SA	61,484.73 ± 188.37	64.74 ± 1.27	137.67 ± 2.68
CRT $\sigma_n^2 = 25$	MCA	41,523.73 ± 199.87	46.71 ± 1.41	153.11 ± 4.59
	RCA	54,107.37 ± 237.36	38.23 ± 1.95	157.41 ± 8.03
	SA	39,423.13 ± 134.57	54.63 ± 1.14	128.06 ± 2.66
CRT $\sigma_n^2 = 100$	MCA	48,403.63 ± 273.51	45.36 ± 1.65	151.27 ± 5.51
	RCA	64,803.34 ± 458.70	37.27 ± 2.21	155.33 ± 9.23
	SA	46,713.75 ± 187.41	55.61 ± 1.34	130.06 ± 3.12
Boeing $\sigma_n^2 = 25$	MCA	60,576.44 ± 287.11	49.63 ± 1.61	143.07 ± 4.62
	RCA	66,005.47 ± 407.82	44.24 ± 2.31	159.11 ± 8.34
	SA	56,147.23 ± 188.17	53.45 ± 1.12	121.74 ± 2.55
Boeing $\sigma_n^2 = 100$	MCA	63,405.22 ± 350.18	51.59 ± 1.96	148.87 ± 5.65
	RCA	73,917.23 ± 520.62	46.58 ± 2.18	168.07 ± 10.17
	SA	60,011.73 ± 244.83	53.58 ± 1.34	121.93 ± 3.06

Our future work will involve the investigation of evolutionary algorithms and hybrid algorithms that combine stochastic annealing algorithms with evolutionary algorithms, specifically the genetic algorithm (GA).[44–46] We expect that by combining the *building blocks* property of the GA with the asymptotic convergence properties of the SA, MCA, and the RCA we can design hybrid stochastic optimization algorithms that combine the advantages and alleviate the individual shortcomings of both, stochastic annealing algorithms and evolutionary algorithms.

Figure 37. Gray-scale *Computer*2 image.

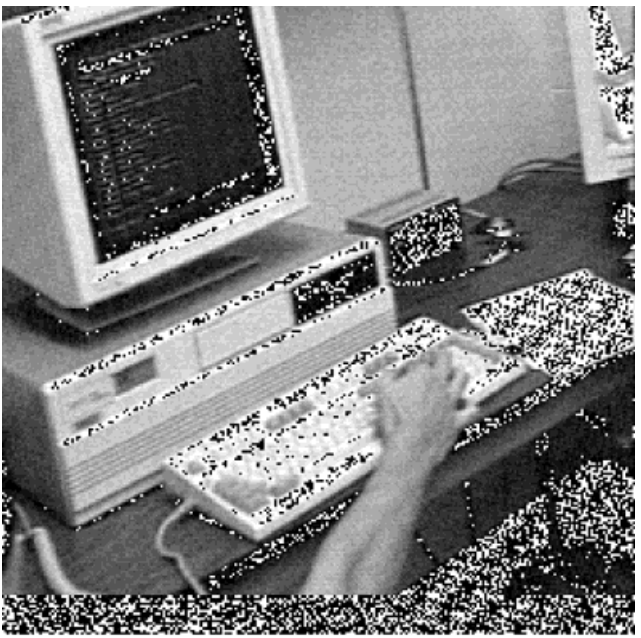

Figure 38. Noisy *Computer*2 image, ($\sigma_n^2 = 100$).

Figure 39. Edge-based segmentation of *Computer*2 image using SA.

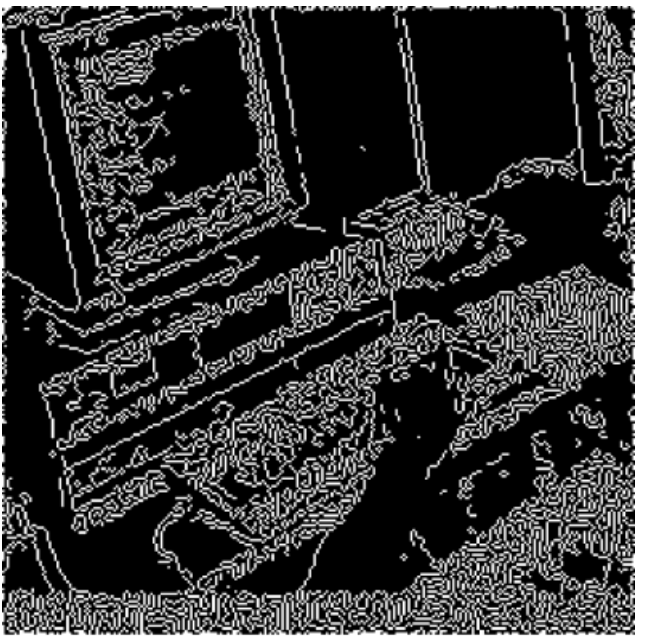

Figure 40. Edge-based segmentation of noisy *Computer*2 image using SA ($\sigma_n^2 = 100$).

References

1. Kirkpatrick S, Gelatt Jr. C, Vecchi M. Optimization by simulated annealing. Science 1983;220(4598):498–516.
2. Creutz M. Microcanonical Monte Carlo simulation. Phys Rev Lett 1983;50(19):1411–1414.
3. Berg BA. Locating global minima in optimization problems by a random-cost approach. Nature 1993;361:708–710.
4. Wang Y, Prade RA, Griffith J, Timberlake WE, Arnold J. A fast random cost algorithm for physical mapping. Proc Natl Acad Sci 1994;91:11,094–11,098.
5. Aarts EHL, Korst, K. Simulated annealing and Boltzman machines: A stochastic approach to combinatorial optimization and neural computing. New York: John Wiley & Sons; 1989.
6. Geman S, Geman D. Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. IEEE Trans Pattern Anal Machine Intell 1984;6(6):721–741.
7. Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E. Equation of state calculations by fast computing machines. J Chem Phys 1953;21:1087–1092.
8. Bhanot G, Creutz M, Neuberger H. Microcanonical simulation of Ising systems. Nucl Phys 1984;B235(FS11):417–434.
9. Fu KS, Mui JK. A survey on image segmentation. Pattern Recogn 1981;13:3–16.
10. Haralick RM, Shapiro LG. Survey: image segmentation techniques. Comput Vision Graphics Image Process 1985;29:100–132.
11. Pal NR, Pal SK. A review on image segmentation techniques. Pattern Recognit 1993;26(9):1277–1294.
12. Sahoo PK, Soltani S, Wong AKC, Chen YC. A survey of thresholding techniques. Comput Vision Graphics Image Process 1988;41:233–260.
13. Brink AB. Gray level thresholding of images using a correlation criterion. Pattern Recognit Lett 1989;9:335–341.
14. Kapur JN, Sahoo PK, Wong AKC. A new method for gray level picture thresholding using the entropy of the histogram. Comput Vision Graphics Image Proc 1985;29:273–285.
15. Mardia KV, Hainsworth TJ. A spatial thresholding method for image segmentation. IEEE Trans Pattern Anal Machine Intell 1988;10:919–927.
16. Perez A, Gonzalez RC. An iterative thresholding algorithm for image segmentation. IEEE Trans Pattern Anal Machine Intell 1987;9:742–751.
17. Pun T. A new method for gray level picture thresholding using the entropy of the histogram. Signal Process 1980;2:223–237.
18. Kittler J, Illingworth J. Relaxation labeling algorithms—a review. Image Vision Comput 1985;3(4):206–216.
19. Peleg S. A new probabilistic relaxation scheme. IEEE Trans Pattern Anal Machine Intell 1980;2:362–369.
20. Rosenfeld A, Hummel RA, Zucker SW. Scene labeling using relaxation operations. IEEE Trans Syst Man Cybern 1976;6:420–453.
21. Chellappa R, Jain AK, Eds. Markov random fields: Theory and applications. San Diego: Academic Press; 1993.
22. Derin H, Elliot H. Modeling and segmentation of noisy textured images using Gibbs random fields. IEEE Trans Pattern Anal Machine Intell 1987;9(1):39–55.
23. Dubes RC, Jain AK. Random field models in image analysis. J Appl Statist 1989;16(2):131–164.
24. Dubes RC, Jain AK, Nadabar SG, Chen CC. MRF model-based algorithms for image segmentation. Proceedings of the Tenth International Pattern Recognition; 1990. pp 808–814.
25. Geiger D, Girosi F. Parallel and deterministic algorithms from MRF's: surface reconstruction. IEEE Trans Pattern Anal Machine Intell 1991;13(5):401–412.

26. Hansen FR, Elliot H. Image segmentation using simple Markov random field models. Comput Graphics Image Process 1982;20:101–132.
27. Bhandarkar SM, Koh J, Suk M. Multi-scale image segmentation using a hierarchical self-organizing feature map. Neurocomputing 1997;14(3):241–272.
28. Ghosh A, Pal NR, Pal SK. Self-organization for object extraction using multi-layer neural networks and fuzziness measure. IEEE Trans Fuzzy Syst 1993;1(1):54–68.
29. Kosko B. Neural networks and fuzzy systems: A dynamical systems approach to machine intelligence. Englewood Cliffs, NJ: Prentice Hall; 1991.
30. Manjunath BS, Simchony T, Chellappa R. Stochastic and deterministic network for texture segmentation. IEEE Trans Acoustics Speech Signal Process 1990;38:1039–1049.
31. Zhou YT, Challappa, R. Artificial neural networks for computer vision. New York: Springer-Verlag; 1992.
32. Besl P, Jain R. Segmentation through variable order surface fitting. IEEE Trans Pattern Anal Machine Intell 1988;10(2):167–192.
33. Haralick RM, Watson LT, Laffey TJ. The topographic primal sketch. Int J Robot Res 1983;2(1):50–72.
34. Bezdek JC, Pal SK. Fuzzy models for pattern recognition: Methods that search for structures in data. New York: IEEE Press; 1992.
35. Jain AK, Dubes RC. Algorithms for clustering data. Englewood Cliffs, NJ: Prentice Hall; 1988.
36. Canny JF. A computational approach to edge detection. IEEE Trans Pattern Anal Machine Intell 1986;8(6):679–698.
37. Nalwa V, Binford T. On detecting edges. IEEE Trans Pattern Anal Machine Intell 1986;8(6):699–714.
38. Perona P, Malik J. Scale space and edge detection using anisotropic diffusion. IEEE Trans Pattern Anal Machine Intell 1990;12(7):629–639.
39. Bhandarkar SM, Zhang Y, Potter WD. An edge detection technique using genetic algorithm-based optimization. Pattern Recognit 1994;27(9):1159–1180.
40. Tan HL, Gelfand SB, Delp EJ. A comparative cost function approach to edge detection. IEEE Trans Syst Man Cybern 1989;19(6):1337–1349.
41. Tan HL, Gelfand SB, Delp EJ. A cost minimization approach to edge detection using simulated annealing. IEEE Trans Pattern Anal Machine Intell 1991;14(1):3–18.
42. Romeo F, Sangiovanni-Vincentelli A. A theoretical framework for simulated annealing. Algorithmica 1991;6:302–345.
43. Bhattacharyya GK, Johnson RA. Statistical concepts and methods. New York: John Wiley & Sons; 1977.
44. Davis L. Handbook of genetic algorithms. New York: Van Nostrand-Reinhold; 1991.
45. Goldberg D. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley; 1988.
46. Holland JH. Adaptation in natural and artificial systems. Ann Arbor, MI: Univ. of Michigan Press; 1975.
47. Marr D, Nishihara HK. Visual information processing: Artificial intelligence and the sensorium of sight. Technol Rev 1978;81(1):2–23.