Candidate Functions for a Parallel Multi-Level Thresholding Technique

L. Hayat, M. Fleury and A. F. Clark
Dept. E.S.E., Essex University, Wivenhoe Park, Colchester, CO4 4SQ, U.K

Abstract

A variety of functions which have been used to find thresholds for images are compared in terms of the actual thresholds returned. All the functions considered are suitable for a novel parallel technique involving an exhaustive search. The comparative computational overhead is taken into consideration. Details are given of the preprocessing employed to find the n modes in the global gray-level histogram. A fuzzy-entropic thresholding method is included by way of comparison.

1 Introduction

Global gray-level image histograms are a convenient method for arriving at an image segmentation which, one hopes, will also reflect a meaningful identification of the objects present in an image. Alternatively, taking regionalized histograms may suffer from skewed estimates due to the reduced pixel sample size. (Local block histograms may be used when the contrast varies across a bi-modal image [3] or if parallelization is a goal.) A typical use would be as a first step in the identification of objects on an industrial conveyor belt [4], where there will also be a time constraint to the identification process. Unsupervised thresholding will be necessary so as to avoid interrupting the flow. Quantization, prior to compression, is *not* the aim of this paper, though the results have a bearing upon such an application. Thresholding is but one technique to perform image segmentation; in [1], segmentation techniques are classified into: thresholding or clustering; edge detection; and region extraction. For further consideration of the latter category *inter alia*, which includes techniques such as region growing and region extraction, see [2].

For well-behaved gray-level distributions, the modal pattern of the distribution can be used to identify objects. (Difficulties arise when there are discontinuities arising from sharp peaks in close proximity.) For uni-modal histograms, which are common in scientific applications, thresholding will result in a 'salt and pepper' image [5] and is best avoided. A variety of methods exist for bi-modal images, especially a between-the-means method [6]. (Iterative methods are not of concern in this paper as they do not fit the computational model that is applied; and in fact the between-the-means method is equivalent [7] to the variation-based method discussed in Section 4). For other applications such as robot vision [8] in an industrial setting, the underlying distribution may be multi-modal so that a practical system will attempt to capture this structure by applying multi-level thresholds. (The underlying distribution implies that the captured image with any noise represents a sample from an ensemble of possible representations of the actual image). The advantage of multi-level thresholding

over simply separating background from foreground is that occluded objects may more easily be identified.

Care should be taken that the thresholding mechanism does not ignore significant histogram modes in the form of small peaks which have been contributed by relatively smaller objects. An industrial setting is assumed in that the scene is correctly lit, so that strong shadows do not contribute spurious peaks, but it is possible for changes in contrast or other effects to shade the returned gray-level values across a common surface. Other approaches are possible which avoid the identification of modes and/or use of the histogram but do not require iterative passes or pre-selection of edge transition pixels. For a critical review of thresholding methods and a description of an alternative approach, not involving excess computation or large data-storage, see [9]. Debate on choice of a thresholding function, if computation is not an issue, revolves around the question of 'disputed' valleys in the global histogram. The debate implies the presence of an underlying image with ground truth. The transition zone between object and background or between object and object can take various forms in the captured image [10]. Several spatially-distinct surfaces may contribute to the same histogram peak and the valleys may be polluted by noise.

The present study developed in the following way. In a paper by Kapur *et al.* [11], a partitioned maximum entropy function was used to find a suitable threshold point for a bi-modal image by maximizing a given function. A suggested extension to multi-modal images runs into the problem of an exponential rise in computation, with the time complexity given by $O(n^r/r!)$ where n is the number of gray levels and r is the number of threshold points [12]. An added difficulty is that the particular entropy function used involves the repeated calculation of natural logarithms, even if certain optimizations are made. It is possible to reduce the time penalty, though not the time complexity, by using a less computationally intensive partitioning function. Where the number of modes is small, this may be adequate if one also adds a parallel algorithm to further reduce the computation time. As it turns out, a particular parallel decomposition is possible, which is brought forward in this paper. Investigation revealed that there are indeed a number of candidate partitioning functions which have a similar algorithmic structure, making them suitable for parallelization. In this study, the competing claims for each of these functions are tested by the actual thresholds that occur, but *not* by analysis of the validity of the function. For other comparative studies with an analytical flavor see [13, 14]. The reader should bear in mind that different thresholding techniques will be appropriate for different image types. In this correspondence, a set of faces is used for testing because people are generally tuned to evaluating such images. However, bearing in mind that the end-user is likely to be robotic, objective measures of assessment are also used.

The rest of the paper is organized as follows. Section 2 describes the preliminary processing necessary, including experience in identification of histogram peaks. Section 3 is a résumé of a parallel algorithm described in a previous paper [12], where the common structure required of suitable partitioning functions is indicated. In Section 4 a description of the thresholding functions is given and the thresholds returned by each function are collated. Relative timings are also to be found in Section 4. Finally, in Section 5 some conclusions are brought together.

2 Preprocessing

The aim of the complete process is to divide an image into $k+1$ homogeneous segments given k threshold points, by the following scheme:

$$g(k:x,y) = \begin{cases} kc, & \text{if} \quad f(x,y) > T(k) \\ (k-1)c, & \quad T(k-1) < f(x,y) \leq T(k) \\ \ldots & \quad \ldots \\ c, & \quad T(0) < f(x,y) \leq T(1) \\ 0, & \quad otherwise. \end{cases} \tag{1}$$

Here $T(k)$ is the gray-tone threshold, $f(x,y)$ is the image intensity function, g is the output image and c is a constant conveniently set as $\lfloor N/k \rfloor$ for N gray levels. Thus, T is an arbitrary monotonic function, with positive integer domain and codomain. For all functions, the histogram is first smoothed to remove noise by means of

$$H'(i) = (H(i-2) + 4H(i-1) + 6H(i) + 4H(i+1) + H(i+2))/16, \tag{2}$$

where $i = 2,\ 3, \ldots, N-2$, and $H(.)$ is the global histogram before smoothing, while $H'(.)$ is the histogram after smoothing. (Terminal values are set to zero.) The advantage of the particular coefficients used for this Gaussian window is that a number of them can be achieved by shifts. The low-pass filter removes troughs which would otherwise give rise to an excessive response, though the use of the filter does assume that smooth gray-tone transitions are reflected in the histogram, which may be a warranted assumption in the case of natural images. An example of a histogram before and after smoothing is shown in Figures 1 & 2. As will be seen from the formulae given in Section 4.1, normalization is used to avoid dominance by large peaks. By the same token, a few pixels outside the gray level range of the bulk of pixels can have a disproportionate effect. Clipping to remove all histogram returns below say 5% of the maximum histogram value becomes a useful precaution which will also remove spurious responses. Another preprocessing technique, not used here, is to use local averaging before taking the histogram, so as to cope with textured images [15]. It is convenient to work subsequently from the observed probability distribution formed by normalizing the gray level histogram, specifically

$$p_i = H'(i)/T,\ \ i = 0,\ 1, \ldots, N-1, \tag{3}$$

where $T = \sum_{i=0}^{N-1} H'(i)$ and the $\{p_i\}$ form the ordinates of a probability mass function (p.m.f.).

It is important to identify correctly the number of modes present in the image, as otherwise it is perfectly possible for the partitioning algorithm to find spurious thresholds, which could arise if the method is arbitrary [16]. Two approaches are possible for identification of the peaks in the global histogram: either assume that there is an underlying number of modes or impose a limit to the number of peaks and work towards that by a peak detecting routine. Because the number of candidate peaks (or, equivalently, valleys) is likely to be high in the first approach, relaxation algorithms are used, such as the Metropolis algorithm in a paper by Brunelli [17], where the minima from all valleys were used as candidate thresholds. It is also perfectly possible to use another optimization technique such as a genetic algorithm [18].

Time considerations led to the use of the second approach, when originally a 'clipping' routine [12] was employed to find the peaks. The problem with that approach for peak detection, as opposed to its use in contrast stretching [19], is that if the peaks do not all lie at

a common level then it is possible to miss a peak. In the histogram of Figure 2, for example, the rightmost large peak would be omitted. Though there exist a variety of peak detectors, (for example, see [20]), it was found that the following method was effective and speedy. In the work of Boukharouba *et al.* [21], the smoothed cumulative distribution function (c.d.f.) formed from the $\{p_i\}$ is used to indicate changes in curvature in the original histogram. In Sezan's paper [22], the procedure is streamlined by using a box-function instead of Chebyshev polynomials for smoothing purposes and the curvature function of the c.d.f is replaced by its differential.

$$r(k) = c(k) - \overline{c}(k),\ k = 0,\ 1, \ldots, N-1. \tag{4}$$

$c(k) = \sum_{i=0}^{k} p(i)$ is the c.d.f.. $\overline{c}(k) = \sum_{i=k-\lfloor L/2 \rfloor}^{k+\lfloor L/2 \rfloor} c(i)w(k-i) = c(k) * w$, with $w(i) = 1/L,\ i = -\lfloor L/2 \rfloor, \ldots, +\lfloor L/2 \rfloor$ ($*$ signifies convolution). $r(k)$ is the peak detected signal. It was found that this differential operator was effective in locating each peak start, maximum, and end by respectively the negative-going zero-crossing, the positive-going zero-crossing and the maximum before the next negative-going zero-crossing in the graph of $r(k)$. L governs the number of peaks found, which was set to five on an heuristic basis. A limit to the number of peaks was set. If too many peaks were found, the process was iterated by

$$\begin{aligned} r_0 &= c_0 - \overline{c_0},\ c_0 = c \\ r_{n+1} &= c_{n+1} - \overline{c_0},\ c_{n+1} = c_n * w,\ n = 0,\ 1,\ 2, \ldots \end{aligned} \tag{5}$$

which avoids the use of a sensitivity parameter. The value of this procedure is illustrated in Figures 3–6, where the alignment of peak start, maximum and end can be checked from the histogram.

3 The Parallel Algorithm

In this section, the main features of a parallelization of the computations is developed. First, consider a sequential search for the maximum of a partitioning function. For simplicity, in Figure 7 the case of two potential threshold points, A and B, and hence three segments is illustrated. The gray-level range is from R_i to R_j; in this example the standard 8-bit range is used. A will be initially fixed at $R_i = 0$ while B is incremented to $R_j - 1$. At each position of B the partition function is calculated and the result is stored. When B reaches R_{j-1}, A is incremented to gray-level one and B returns to gray-level two. The procedure repeats until A has reached $R_j - 2$. The maximum value of the function is used to indicate the optimum position for A and B. It is found that the computation can be reduced considerably by not performing complete summations for each position of B. In Figure 8 a new value is uncovered to add to partition one, which can be found by simply taking the value passed over from partition two. This 'running' method may also be used for the outer calculation loop as A is advanced. The actual 'addition' calculations will differ, depending on the partitioning function.

If the running method is to be retained, an equal-range parallel decomposition is possible, which is shown in Figure 9 for four tasks and three threshold points. The first partition point, $A_i,\ i = 0,\ i, \ldots t-1$ with t the number of tasks, is only incremented to just before the starting position for the next task. In our previous paper [12], a load-balancing formula is given to avoid the extra calculations performed by tasks further away from $R_j - 1$. However,

that formula can be simplified by reversing the numbering of the gray-level range and by equating directly the work performed by each task. In Figure 10, the boundaries of the gray level ranges are indicated by $\{k_i\}$. Using the formula for the sum of consecutive integers, in an example for two threshold points and four tasks, the $\{k_i\}$ are related by

$$\begin{aligned} (k_1)k_1/2 &= (k_2 - k_1)(k_2 + k_1)/2 \quad (6) \\ (k_3 - k_2)(k_3 + k_2)/2 &= ((N-1) - k_3)((N-1) + k_3)/2 \quad (7) \end{aligned}$$

Simplifying, by using the difference of two squares, gives

$$\begin{aligned} k_2^2 &= 2k_1^2 \quad (8) \\ k_3^2 &= 3k_1^2 \quad (9) \\ (N-1)^2 &= 4k_1, \quad (10) \end{aligned}$$

from which, knowing N, the number of gray-levels, it is easy to find the range given to each task. (In practice the result will be approximate because integer range values are needed and because the formula is exact only for even-valued ranges.) A generalization for all r, where r is the number of threshold points, is given by

$$(k_{i+1}^{r-1} + k_i^{r-1})(k_{i+1} - k_i) = (k_{i+2}^{r-1} + k_{i+1}^{r-1})(k_{i+2} - k_{i+1}),\ i = 1,\ 2, \ldots, r-2. \quad (11)$$

4 Evaluation of the Thresholding Functions

4.1 The Function Definitions

Four different functions were used on the same image set in order to compare the thresholds found. In all cases the function value was computed at all potential threshold points. The maximum value of the function was taken as the chosen solution.
The entropy function is given for the bi-modal case as

$$Entp(s) = -(E_a + E_b), \quad s \in (0, N-1) \quad (12)$$

where s is a potential threshold point and N is the number of different gray levels. The value of the two partitions is given by

$$\begin{aligned} E_a &= \textstyle\sum_{i=0}^{s} (\frac{p_i}{P_s}) \ln(\frac{p_i}{P_s}) \\ E_b &= \textstyle\sum_{i=s+1}^{N} (\frac{p_i}{\acute{P}_s}) \ln(\frac{p_i}{\acute{P}_s}). \end{aligned} \quad (13)$$

$P_s = \sum_{i=0}^{s} p_i$ and $\acute{P}_s = \sum_{i=s+1}^{n} p_i$. The method is extended to multi-modal images by defining $Entp[s] = -\sum_{i=1}^{k} E_i$ in the natural way. Since the natural logarithm is a non-linear function, it is impossible to avoid recalculation of the natural logarithms in an inner loop. A computational form of Eqs. 13 is

$$\begin{aligned} E_a &= -\ln(P_s) + H_s/P_s,\ H_s = \textstyle\sum_{i=0}^{s} p_i \ln(p_i) \\ E_b &= -\ln(\acute{P}_s) + \acute{H}_s/\acute{P}_s,\ \acute{H}_s = \textstyle\sum_{i=s+1}^{N-1} p_i \ln(p_i) \end{aligned} \quad (14)$$

The useful running relationship is present by reason of

$$\begin{aligned} &P_{s+1} = P_s + p_{s+1},\ \acute{P}_{s+1} = \acute{P}_s - p_{s+1} \\ &H_{s+1} = H_s + p_{s+1}\ln(p_{s+1}),\ \acute{H}_{s+1} = \acute{H}_{s+1} = \acute{H}_s - p_{s+1}\ln(p_{s+1}) \end{aligned} \tag{15}$$

For computational efficiency, it is possible place the logarithm in the outer loop of the calculation [23], where the new measure is called a correlation function:

$$Corr(s) = -(C_a + C_b), \quad s \in (0, N-1) \tag{16}$$

when

$$\begin{aligned} &C_a = \ln(\tfrac{H_s}{P_s^2}), H_s = \textstyle\sum_{i=0}^{s} p_i^2, P_s = \sum_{i=0}^{s} p_i \\ &C_b = \ln(\tfrac{\acute{H}_s}{\acute{P}_s^2}), \acute{H}_s = \textstyle\sum_{i=s+1}^{N} \acute{p}_i^2, \acute{P}_s = \sum_{i=s+1}^{N} \acute{p}_i. \end{aligned} \tag{17}$$

The running method now works by

$$\begin{aligned} &H_{s+1} = H_s + p_{s+1}^2 \\ &P_{s+1} = P_s + p_{s+1} \\ &\acute{H}_{s+1} = \acute{H}_s + \acute{p}_{s+1}^2 \\ &\acute{P}_{s+1} = \acute{P}_s + \acute{p}_{s+1}. \end{aligned} \tag{18}$$

In a paper by Otsu [24], a variance-type measure was introduced that has been generally well received, for instance in [9]. In statistics, a similar discriminant or bias analysis is used to identify suitable class sizes for histogram bins. The function to maximize is:

$$\sigma^2(s) = W_0 W_1 (M_1 - M_0)^2, \quad s \in (0, N-1) \tag{19}$$

where

$$\begin{aligned} &W_0(s) = \textstyle\sum_{i=0}^{s} p_i,\ M_0(s) = -\frac{H_0(s)}{W_0},\ H_0(s) = \sum_{i=0}^{s} i p_i \\ &W_1(s) = \textstyle\sum_{i=s+1}^{N} p_i,\ M_1(s) = -\frac{H_1(s)}{W_1}, H_1(s) = \sum_{i=s+1}^{N} i p_i. \end{aligned} \tag{20}$$

To labour a point, the running method structure can also be used because

$$\begin{aligned} &W_0(s+1) = W_0(s) + p_{s+1},\ W_1(s+1) = W_1(s+1) - p_{s+1} \\ &H_0(s+1) = H_0(s) + (s+1)p_{s+1},\ H_1(s+1) = H_1(s) - (s+1)p_{s+1} \end{aligned} \tag{21}$$

This function is extended to the k-modal case by

$$\sigma^2(s_1, \ldots, s_{k-1}) = W_0 W_1 W_2 \ldots W_k \textstyle\sum_{i=1}^{k} \sum_{j=0, j\neq i}^{k} (M_i - M_j)^2, \tag{22}$$

with $\{s_i\}$ being the threshold point set, when the extension to four or more partitions may lose accuracy.

In a contribution by Reddi *et al.* [25] it is proven, by equating to zero the derivative with respect to the threshold set $\{k_i\}$, that the condition for a unique maximum is

$$M_i + M_{i+1} = 2s_i,\ i = 1,\ 2, \ldots, k-1 \tag{23}$$

which reduces the computation but more importantly means that fast search methods can be used. The method given by Reddi *et al.* [25], which is iterative, is not amenable to parallelization. If the search method is quick, then it can become preferable to replicate the search between parallel tasks, reserving the parallelization to the formation of the global histogram and the final segmentation of the image. In Figure 11, the form of the variance function is plotted for two threshold points, when only every tenth value is displayed for practical reasons. Examination of the data reveals that there is a unique maximum, without local maxima. However, it should not be assumed that there are no local maxima in all cases. In research by Kittler *et al.* [7], an example is given where two maxima are present for a bi-modal image and, more significantly, the global maximum is incorrect. It is assumed that images have been screened to avoid this aberration, which occurs when one histogram peak dominates another in a bi-modal image. There are local maxima for the equivalent plots of the other three functions, as displayed in Figures 12–14, making the functions unsuitable for fast searches. Clipping would remove the oscillatory excursions seen at the high end of the range in the plots.

A function originally used in the context of a simulated annealing algorithm [17] depends on partitioning the area of the peaks. In this study, the initial number of candidate threshold points is sharply reduced, which allows the complete range of gray-levels to be scanned in the search. The function to maximize is

$$Area(s) = A_a + A_b, \quad s \in (0, N-1) \tag{24}$$

when

$$\begin{aligned} A_a &= \frac{\min(|P_a - p_0|, |P_a - p_s|)}{P_a}(1 - Q_a/(P_a W_a)) \\ A_b &= \frac{\min(|P_b - p_s|, |P_b - p_{N-1}|)}{P_b}(1 - Q_b/(P_b W_b)), \end{aligned} \tag{25}$$

where $P_a = \max(p_i),\ i = 0, 1, \ldots s,\ P_b = \max(p_i),\ i = s, 1, \ldots N,\ W_a = s$ and $W_b = N - s$ (the widths of the gray-level ranges on either side of the prospective threshold point s), $Q_a = \sum_{i=0}^{s} p_i$ and $Q_b = \sum_{i=s} N - 1 p_i$ (the total areas of the p.d.f. on either side of the threshold point). The function is extended to the multi-modal case in a similar way to the maximum entropy and correlation functions. Unlike the previous functions considered, the complete gray level range is not used for the area-based function. A set of concavities is first selected by finding all points where $p_{i-1} > p_i < p_{i+1},\ i = 1,\ 2, \ldots, N-2$ (Figure 15). The prospective partitioning points are now stepped through the concavity points. There is no longer a need to pass incrementally through each gray level. The original p.m.f. is still used for calculation of the function quantities. The running method of calculation is retained, except that it is necessary to find again the maximum value in the appropriate segments. The possibility of reducing the search space is an advantage over the maximum entropic, correlation function and variation-based methods. Recalculation of the partitioned areas can also be assisted by using a c.d.f..

4.2 Results

The first image (Face 1), before thresholding, is shown in Figure 16, as are the results of the various thresholding techniques. Further results are given in Figure 17. The histogram for this image is given in Figure 5, from which it will be seen that there are broadly three

valleys. However, this was determined automatically at the peak detection stage, whereupon the four partitioning functions defined in Section 4.1 were applied. As a point of comparison, a binary segmentation has been applied to the image by taking simply the arithmetic mean of the gray-level distribution as the threshold.

Another interesting comparison is with a threshold resulting from a fuzzy-entropic measure. The method is a variant of the approach of Pal *et al.* [26] in which a fuzzy window of width given by an heuristically-derived bandwidth is passed across the global histogram. (The locations of the thresholds are taken to be the minima of the output.) Instead, the fuzzy membership function, an S-function, is distorted [27] within a range bounded by adjacent peaks in order to find a minimum for each peak pair. This method is described briefly in a previous paper [12], where the generalized S-function is given as

$$S(x,a,b,c) = \begin{cases} 0, & x \le a \\ k(\frac{x-a}{c-a})^2, & a \le x \le b, k = \frac{c-a}{b-a} \\ 1 - k(\frac{x-c}{c-a})^2, & b \le x \le c, k = \frac{c-a}{c-b} \\ 1, & x \ge c \end{cases} \tag{26}$$

Here we give examples of its successful application and make comparisons with other methods. This function is symmetrical only when the cross-over point b lies exactly in the middle of the fuzzy region (a,c) (*i.e.*, $b = \frac{a+c}{2}$, bandwidth $\Delta b = b - a = c - b$ and $k = 2$). The result of thresholding Face 1 by this mechanism is shown in Figure 17.

The same set of functions was also applied to two other images, Face 2 and Lena (with histogram in Figure 1). The results of these experiments are recorded in Figures 18–22, so that the reader can form an opinion. It is also possible to make objective, quantitive assessments of the success of the chosen thresholds. In a paper by Levine and Nasif [28], an automatic iterative approach to thresholding was introduced, whereby the strength of the threshold depends on normalized statistical measurements. Developing from this [14], thresholding methods were separated under two criteria: shape and uniformity. In a previous paper [12], the two measures are extended to the multi-level case and here a third discrepancy measure is added. The result of applying these measures to the three face images is tabulated in Tables 1–3. For convenience, the shape and uniformity measures are repeated here:

$$\text{Uniformity} = 1 - \frac{\sum_{x,y\,\epsilon\,R}(f(x,y) - \mu)^2}{B}, \tag{27}$$

where R is the thresholded region of concern, f returns the gray-level, μ is the mean gray level within the region and B is a normalizing factor based on the region area and its gray-level range. Uniformity is a least-squares measure of the departure of each region selected by a putative threshold from its mean gray-level value. The region may not be topologically compact.

$$\text{Shape} = 1 - \frac{\sum_{i\,\epsilon\,R}(G + I)}{C}, \tag{28}$$

where R is the thresholded region of concern, G is a neighborhood gradient measure, I is the absolute difference between the gray-level of the pixel of concern and its 8-neighborhood gray-level mean and C is a normalizing factor based on the region area and its gray-level

range. The boundary of the image is omitted in this measure as are pixels not in the interior of the thresholded region. The neighborhood gradient measure is given by

$$G = \sum_{k=1}^{4} D_k^2, \tag{29}$$

with

$$\begin{aligned} D_1 &= f(x+1,y) - f(x-1,y) && (30)\\ D_2 &= f(x,y+1) - f(x,y-1) && (31)\\ D_3 &= f(x-1,y-1) - f(x+1,y+1) && (32)\\ D_4 &= f(x+1,y-1) - f(x-1,y+1). && (33) \end{aligned}$$

In summary, the shape characteristic is again a form of least-squares measure, indicating the local homogeneity and gray-level rate of change of each region selected by the thresholding procedure. The discrepancy function for t thresholds is defined as

$$\mathrm{dis}(k) = \begin{cases} \sum_{i=0}^{s_{k+1}-1} (i-\mu_k)^2 p_i, & k = 0 \\ \sum_{i=s_k}^{s_{k+1}-1} (i-\mu_k)^2 p_i, & k = 1,\ 2, \ldots, t-1 \\ \sum_{i=s_k}^{N-1} (i-\mu_k)^2 p_i, & k = t \end{cases} \tag{34}$$

$\{s_k\}$ are the chosen threshold points. Eq. 34 is taken from the paper by Yen *et al.* [23], where it was included in a cost function which was used with the purpose of finding the optimum number of thresholds. Discrepancy gives a measure of the difference between the thresholded image and the original by means of a central moments measure on the gray-level histogram partitioning. A minimum cost is sought.

In each table, the thresholds returned are recorded. The maximum entropy and correlation functions give similar threshold values, particularly for Face 1 and Lena. As noted in Section 4.1, the correlation function entails reduced computation. Thus it is verified that the correlation function is a better choice for parallel implementation. Surprisingly, in view of the fact that the two methods involve different principles, the fuzzy-entropic method and the area-based method also return similar thresholds. However, these thresholds are not the same as the maximum-entropy ones. The variation-based method returns values which in any valleys tend to lie between the maximum-entropy ones and the fuzzy-entropic thresholds. Inspection of the histograms indicates that the maximum-entropic thresholds are inclined to lie higher up the side of the valley, on the shoulder of the peaks. A threshold on the shoulder may indicate that one peak disguised the gray-level distribution of two objects [29]. The fact that the same tendency occurs for all thresholds in this study points to bias away from the valley floor in the cases of the maximum entropy method and correlation function. Therefore, we cannot recommend the use of the maximum entropy method or the correlation function for the type of images surveyed. The variation-based method remains a good compromise solution. In the tables, an average of the region values for a particular thresholding method is given along with the ranking for that measure against the other methods. The uniformity measure does pick out a pecking order, while the shape measure is unsuitable for this purpose. The discrepancy measure may well have value in estimating the optimum number of thresholds, as originally intended, but appears to have little credence as a way of separating out different thresholding methods.

4.3 Timings

A comparison of the relative timings between the different thresholding methods was made using a sequential version of the algorithm. The tests were performed on a SPARC architecture machine with a nominal CPU speed rating of 20MHz and equipped with a Weitek 3170 floating-point unit. The measurements were repeated over 50 iterations because of the possibility of varying background load. The timings, which do not include the data dependent histogram creation and segmentation phases, are for three thresholds. The area function method was applied to all gray levels in Figure 24. Because there are local calculations at each gray level increment in order to find the maximum peaks, the area function resulted in much longer timings, though it may be possible to improve this timing somewhat by changing the peak maximum search method. Though the area function does not use logarithms, it does require a number of calculations beyond the summations, which are reduced because of the running method. There is little to choose between the variance method and the maximum entropy method in terms of timing, though the correlation function method is faster. Turning to Figure 25, the area function has been applied to just the cavities in the histogram. In the plot marked 'concavity' the peaks returned by the peak-detector are used as a basis for determining the number of thresholds. It will be seen that the simulated annealing algorithm, as originally described in Brunelli's paper [17], gives slightly slower timings on average. Both calculation algorithms give the same thresholds to within a gray-level, so one would naturally select the faster method. Comparing the timings to the fuzzy-entropic method described here, it will be seen that both area function timings are less than the fuzzy method, while it will be recalled that the area and fuzzy methods give similar thresholds. The area method therefore is to be preferred. As a point of comparison, the same test is performed in the 'standard' fuzzy method [26], when the timings are still further reduced, though the method was iterated over a number of bandwidths. As might be expected (Figure 26), the timing ordering is preserved when the program was parallelized with four processors (SUN4 workstations) using a parallel-Unix communication harness, PVM [30]. Notice that the timing figures are much reduced over the sequential version primarily because of the reduction in the number of thresholds from three to two, when the order of complexity of the exhaustive search is exponential. The area function is again used over all gray-levels as a means of comparison, but would actually be applied only to concavity points in practice. The timings vary for each run or iteration because of varying background load.

5 Conclusions

A comparison has been made of a number of multi-thresholding methods. The suggestion is that where the actual thresholds returned are virtually identical, it pays to use a method with reduced computation. It is confirmed that the maximum entropy method and the correlation function method give similar thresholds, but the correlation function is quicker. A more unexpected result is that the fuzzy-entropic method and the area method are also similar in results, though the fuzzy method has a different search procedure to the area function method. Otsu's variance method, which gives results between the two types of threshold, may be implemented by a fast search method if local maxima can be assumed not to be present, but otherwise it is as costly in computation as the maximum entropy method. A further point to note is that the simpler functions, those avoiding logarithms, will be less prone to rounding errors in distinguishing the exact maximum. The area function has a key

computational advantage in that the complete gray-level range is not used. In fact, as a result of timing studies, the area function method was found to be the fastest of all methods tabulated on this occasion.

As the number of thresholds is increased the burden of calculation escalates. A common property of three of the functions considered is that the presence of local maxima makes an exhaustive search necessary. If the number of thresholds is not too great, as will generally be the case in practice, an exhaustive search is still possible, because the theoretical time complexity may not outstrip the actual calculation times. The calculation algorithm can be improved by a running method and parallelization can also be used to reduce the time penalty. A defect of entropic methods, which is noted in the survey by Kapur *et al.* [14], is that they are not resilient to noise because they respond to the gray-level distributions. In a paper by Abutaleb [31], a remedy is proposed in which the mean intensity value of the local neighborhood is plotted against the number of pixels with a particular gray-level. It is perfectly possible to partition the resulting 2D histogram search space by means of the method proposed here. There may well be other functions in this field which will yield to a similar approach in the future.

Acknowledgement

This work was carried out as part of project IED3/1/2171 ("Parallel Reconfigurable Image Processing Systems"). We gratefully acknowledge the help given by the anonymous referees in revising this paper for publication.

References

[1] S. K. Fu, J. K. Mui, A Survey on Image Segmentation, *Pattern Recognition,* **13,** 1981, 3-16.

[2] R. M. Haralick, L. G. Shapiro, *Computer and Robot Vision,* Volume 1, Addison-Wesley, Reading, Mass. 1992.

[3] S. C. Sahasrabudhe, K. S. D. Gupta, A valley-seeking threshold selection technique, in *Computer Vision and Image Processing,* (A. Rosenfeld and L. Shapiro, Eds.), pp. 55-65, Academic Press, New York, 1992.

[4] E. R. Davies, Application of the generalised Hough transform to corner detection, *IEE Proceedings Part E,* **135(1),** 1988, 49-54.

[5] J. Kittler and J. Illingworth, Minimum error thresholding, *Pattern Recognition,* **19(1),** 1986, 41-47.

[6] T. Ridler and S. Calvard, Picture thresholding using an iterative selection method, *IEEE Transactions on Systems, Man, and Cybernetics,* **8(8),** 1978, 630-632.

[7] J. Kittler and J. Illingworth, On threshold selection using clustering criteria, *IEEE Transactions on Systems, Man, and Cybernetics,* **15(5),** 1985, 652-655.

[8] R. C. Gonzalez and R. Safabakhsh, Computer vision techniques for industrial applications and robot control, *IEEE Computer,* **15,** 1982, 17-32.

[9] J. Kittler, J. Illingworth, and J. Föglein, Threshold selection based on a simple image statistic, *Computer Vision, Graphics, and Image Processing,* **30,** 1985, 125-147.

[10] S. K. Park and R. A. Schowengerdt, Image sampling, reconstruction, and the effect of sample-scene phasing, *Journal of Applied Optics,* **21(17),** 1982, 3142-3151.

[11] 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, and Image Processing,* **29,** 1985, 273-285.

[12] M. Fleury, L. Hayat and A. F. Clark, Parallel entropic auto-thresholding, To appear in *Image and Vision Computing* 1996.

[13] S. U. Lee and S. Y. Chung and Park, R. H., A comparative performance study of several global thresholding techniques for segmentation, *Computer Vision, Graphics, and Image Processing,* **52,** 1990, 171-190.

[14] P. K. Sahoo, S. Soltani, A. K. C. Wong and Y. C. Chen, A survey of thresholding techniques, *Computer Vision and Image Processing,* **41,** 1988, 233-260.

[15] L. Davis, A. Rosenfeld and J. Weska, Region extraction by averaging and thresholding, *IEEE Transactions on Systems, Man, and Cybernetics,* **8(8),** 1975, 622-629.

[16] D-M Tsai and Y-H Chen, A fast histogram-clustering approach for multi-level thresholding, 1992, Pattern Recognition Letters, 13, 245-252.

[17] R. Brunelli, Optimal histogram partitioning using a simulated annealing technique, *Pattern Recognition Letters.* **13,** 1992, 581-586.

[18] J. R. Koza, *Genetic Programming: On the Programming of Computers by Natural Selection,* MIT Press, 1992.

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

[20] J-O Eklundh and A. Rosenfeld, Peak detection using difference operators, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **1(3),** 1979, 317-325.

[21] S. Boukharouba, J. M. Rebordao and P. L. Wendel, An amplitude segmentation method based on the distribution function of an image, *Computer Vision, Graphics and Image Processing* **29,** 1985, 47-59.

[22] M. I. Sezan, A peak detection algorithm and its application to histogram-based image data reduction, *Computer Vision, Graphics and Image Processing,* **49,** 1990, 36-51.

[23] J-C Yen, F-J Chang, S. Chang, A new criterion for automatic multilevel thresholding, *IEEE Transactions on Image Processing,* **4(3),** 1995, 370-378.

[24] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man, and Cybernetics* **9(1),** 1979, 62-66.

[25] S. S. Reddi, S. F. Rudin and H. R. Keshavan, An optimal multiple threshold scheme for image segmentation, *IEEE Transactions on Systems, Man, and Cybernetics,* **14(4),** 1984, 661-665.

[26] S. K. Pal, R. A. King and A. A. Hashim, Automatic grey level thresholding through index of fuzziness and entropy, *Pattern Recognition Letters,* **1,** 1983, 141-146.

[27] H. Li, H. and H. S. Yang, Fast and reliable image enhancement using a fuzzy relaxation technique, *IEEE Transactions on Systems, Man, and Cybernetics,* **19(5),** 1989, 1276-1281.

[28] M. D. Levine and A. M. Nazif, Dynamic measurement of computer-generated image segmentation, *IEEE Transactions in Pattern Analysis and Machine Intelligence,* **7,** 1985, 155-164.

[29] A. Rosenfeld, and P. De La Torre, Histogram concavity analysis as an aid in threshold selection, *IEEE Transactions on Systems, Man, and Cybernetics,* **13(3),** 1983, 231-235.

[30] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sunderam, *PVM 3 User's Guide and Reference Manual,* Oak Ridge National Laboratory 1993.

[31] A. S. Abutaleb, Automatic thresholding of gray-level pictures using two-dimensional thresholding, *Computer Vision, Graphics, and Image Processing,* **47,** 1989, 22-32.

Image : Face 1				
Function	Threshold range	Evaluation		
		Uniformity	Shape	Discrepancy
Averaging	0-105	0.800	0.719	0.901
	105-255	0.749	0.893	0.843
Average		0.775[4]	0.806[1]	0.873[5]
Entropy	0-74	0.810	0.710	0.971
	74-131	0.631	0.692	0.966
	131-173	0.584	0.721	0.996
	173-255	0.969	0.810	0.998
Average		0.749[6]	0.733[3]	0.983[3]
Correlation	0-78	0.802	0.714	0.963
	78-132	0.638	0.707	0.972
	132-175	0.596	0.723	0.995
	175-255	0.970	0.798	0.998
Average		0.751[5]	0.736[2]	0.982[4]
Variance	0-39	0.903	0.576	0.999
	39-103	0.702	0.591	0.958
	103-184	0.644	0.841	0.927
	184-255	0.987	0.686	0.999
Average		0.783[3]	0.673[6]	0.971[6]
Fuzzy	0-52	0.878	0.647	0.995
	52-101	0.715	0.501	0.981
	101-151	0.766	0.781	0.990
	151-255	0.938	0.871	0.991
Average		0.824[1]	0.700[5]	0.989[1]
Area	0-55	0.870	0.662	0.994
	55-110	0.705	0.548	0.973
	110-154	0.745	0.777	0.993
	154-255	0.943	0.867	0.992
Average		0.815[2]	0.714[4]	0.988[2]

Table 1: Selection of Optimum Threshold Points.

Image : Face 2				
Function	Threshold range	Evaluation		
		Uniformity	Shape	Discrepancy
Averaging	0-114	0.837	0.821	0.877
	114-255	0.823	0.751	0.840
Average		0.830[1]	0.786[1]	0.858[6]
Entropy	0-64	0.738	0.651	0.988
	64-137	0.751	0.827	0.924
	137-209	0.695	0.657	0.950
	209-255	0.746	0.725	0.999
Average		0.732[6]	0.715[3]	0.965[4]
Correlation	0-58	0.752	0.637	0.992
	58-114	0.800	0.808	0.970
	114-164	0.633	0.652	0.977
	164-255	0.852	0.814	0.975
Average		0.759[2]	0.728[2]	0.979[1]
Variance	0-39	0.764	0.598	0.998
	39-114	0.814	0.798	0.945
	114-203	0.670	0.662	0.890
	203-255	0.726	0.773	0.998
Average		0.743[5]	0.708[5]	0.958[5]
Fuzzy	0-106	0.829	0.819	0.897
	106-135	0.655	0.630	0.996
	135-173	0.663	0.580	0.991
	173-255	0.849	0.819	0.983
Average		0.749[4]	0.712[4]	0.967[3]
Area	0-100	0.814	0.819	0.905
	100-127	0.676	0.632	0.997
	127-176	0.669	0.621	0.982
	176-220	0.798	0.711	0.995
	220-255	0.813	0.665	0.999
Average		0.754[3]	0.690[6]	0.976[2]

Table 2: Selection of Optimum Threshold Points.

Image : Lena				
Function	Threshold range	Evaluation		
		Uniformity	Shape	Discrepancy
Averaging	0-98	0.706	0.765	0.872
	98-255	0.789	0.882	0.813
Average		0.748[3]	0.824[1]	0.842[4]
Entropy	0-56	0.805	0.658	0.987
	56-99	0.639	0.697	0.984
	99-142	0.654	0.734	0.984
	142-180	0.644	0.653	0.995
	180-255	0.878	0.837	0.997
Average		0.723[5]	0.716[2]	0.989[2]
Correlation	0-58	0.797	0.660	0.985
	58-101	0.633	0.701	0.983
	101-143	0.662	0.729	0.986
	143-180	0.641	0.642	0.996
	180-255	0.878	0.837	0.997
Average		0.722[6]	0.714[3]	0.989[2]
Variance	0-24	0.880	0.556	0.999
	24-80	0.606	0.666	0.967
	80-138	0.649	0.768	0.959
	138-199	0.595	0.754	0.975
	199-255	0.911	0.824	0.999
Average		0.728[4]	0.714[3]	0.980[3]
Fuzzy	0-46	0.836	0.636	0.994
	46-80	0.700	0.628	0.994
	80-112	0.681	0.648	0.994
	112-175	0.741	0.791	0.972
	175-255	0.872	0.837	0.995
Average		0.766[1]	0.708[4]	0.990[1]
Area	0-42	0.849	0.626	0.996
	42-77	0.681	0.625	0.994
	77-117	0.679	0.686	0.988
	117-169	0.733	0.770	0.983
	169-255	0.864	0.833	0.993
Average		0.761[2]	0.708[4]	0.990[1]

Table 3: Selection of Optimum Threshold Points.

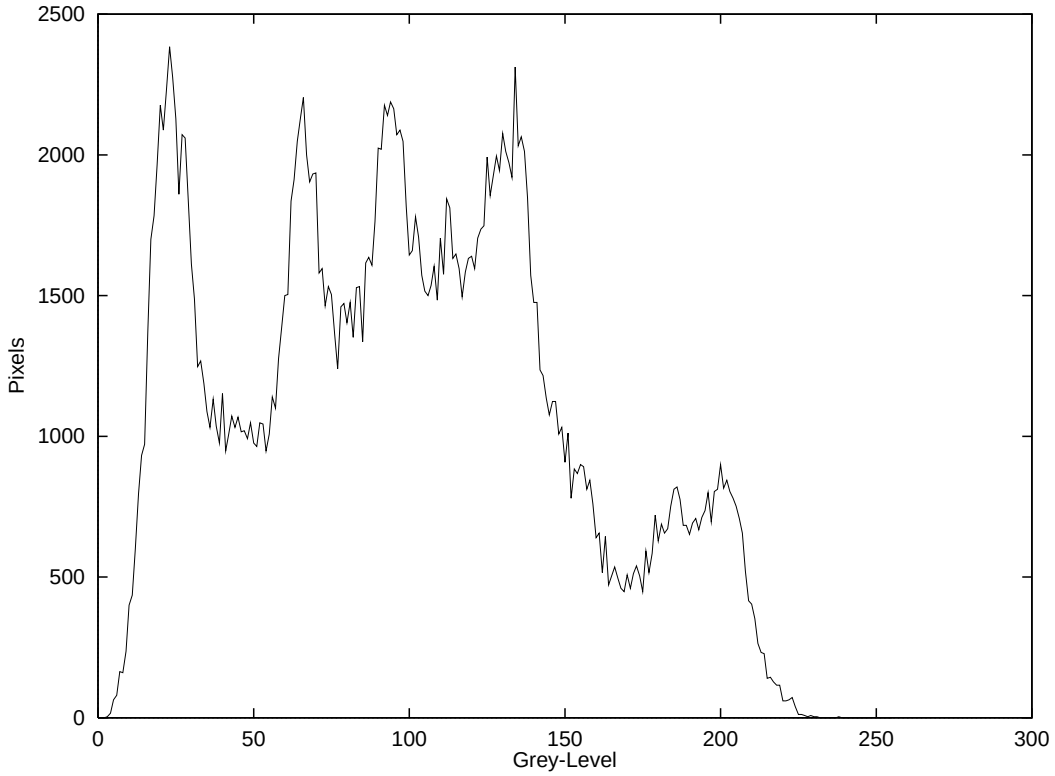

Figure 1: Gray-Level Histogram of Lena Image

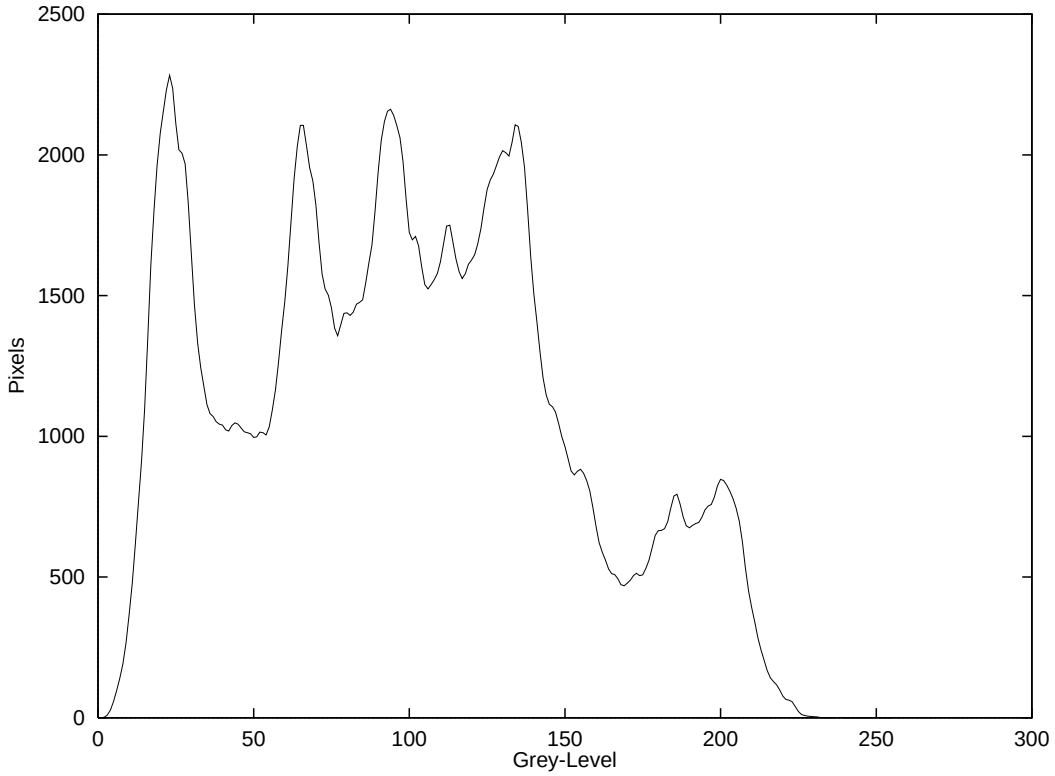

Figure 2: Gray-Level Histogram of Lena Image after Smoothing

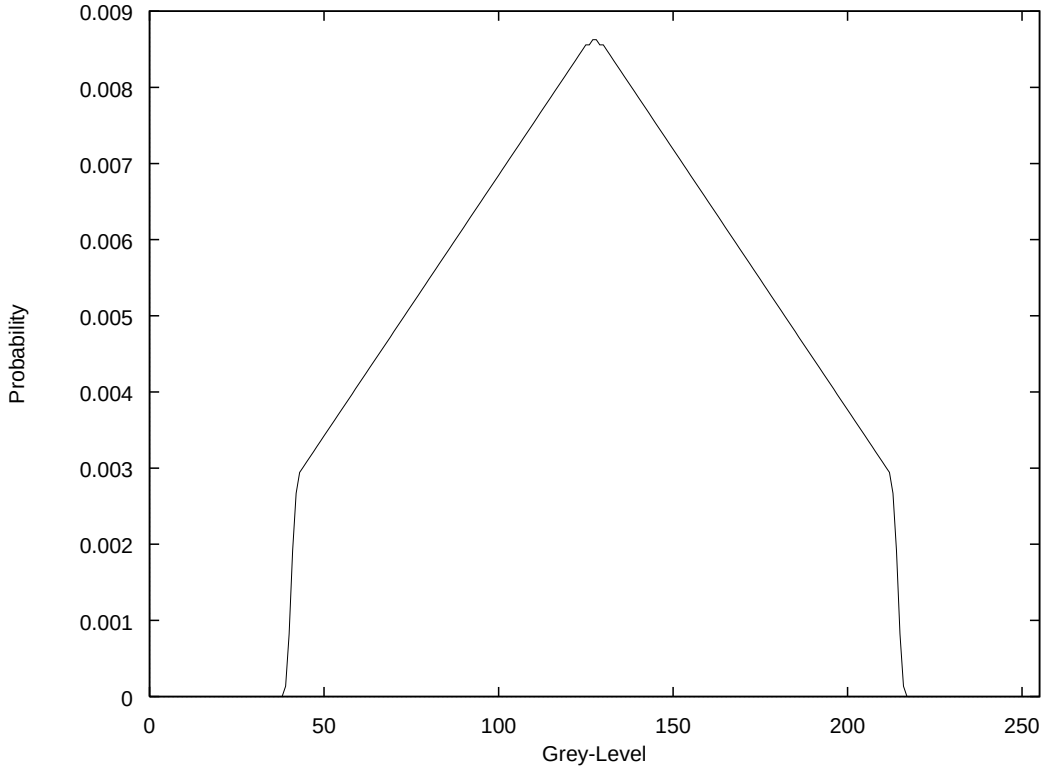

Figure 3: Synthetic Histogram

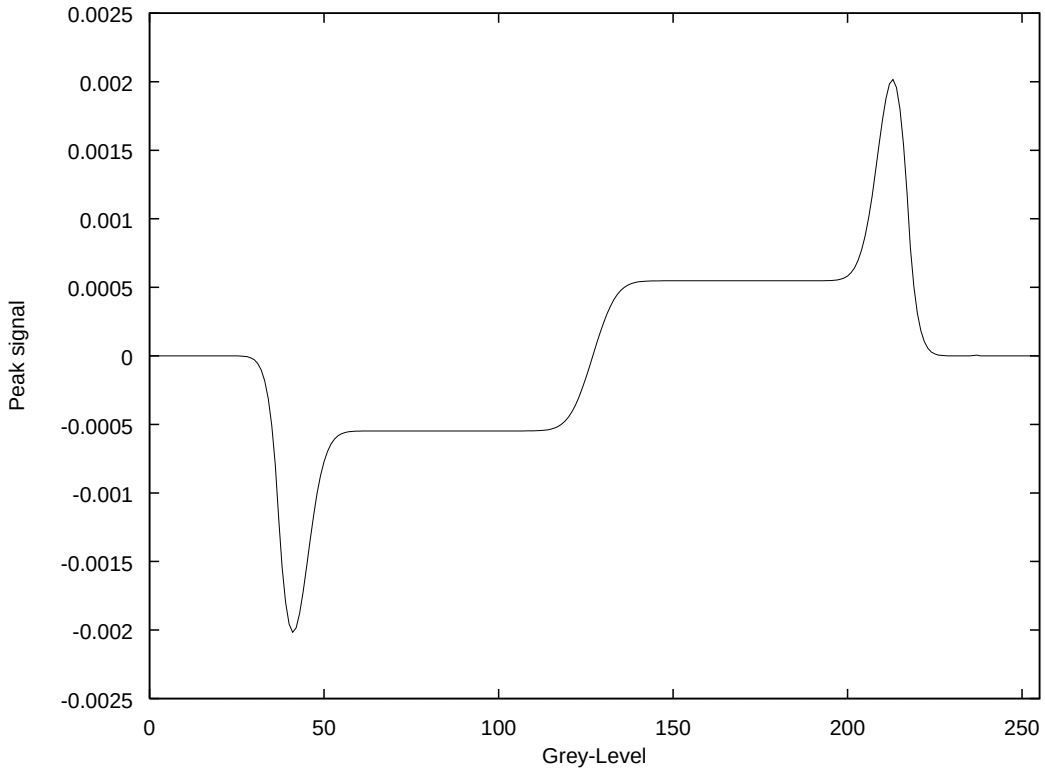

Figure 4: Peak Detection Signal for Figure 3

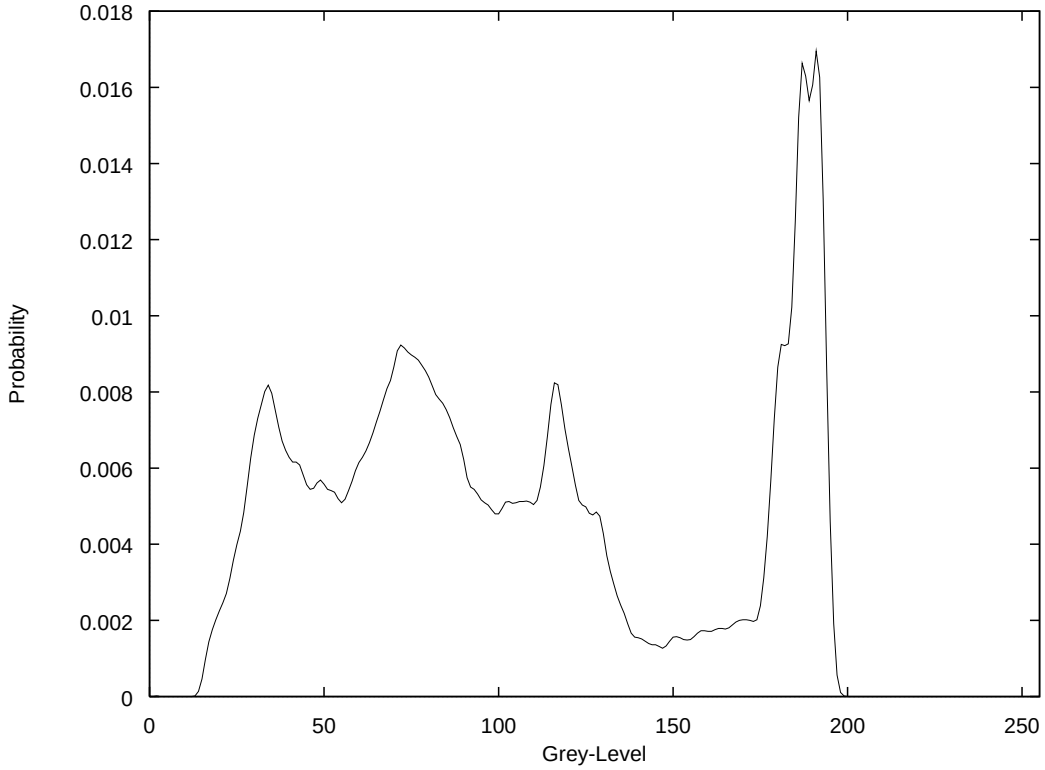

Figure 5: Example Image Histogram

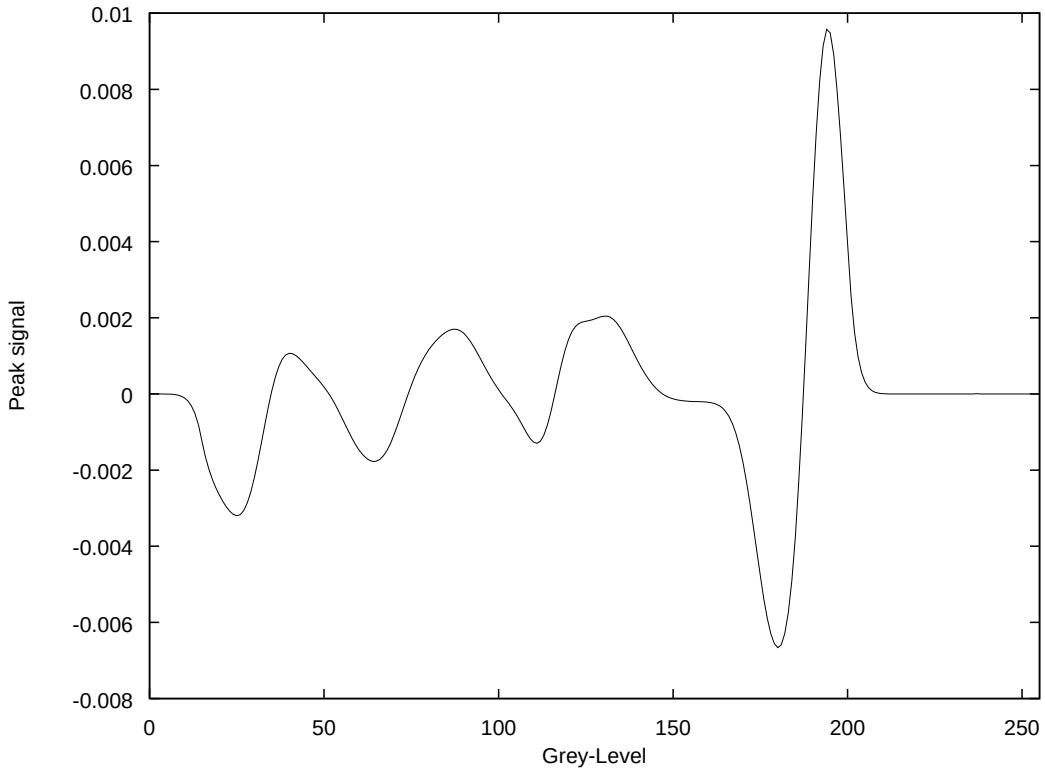

Figure 6: Peak Detection Signal for Figure 5

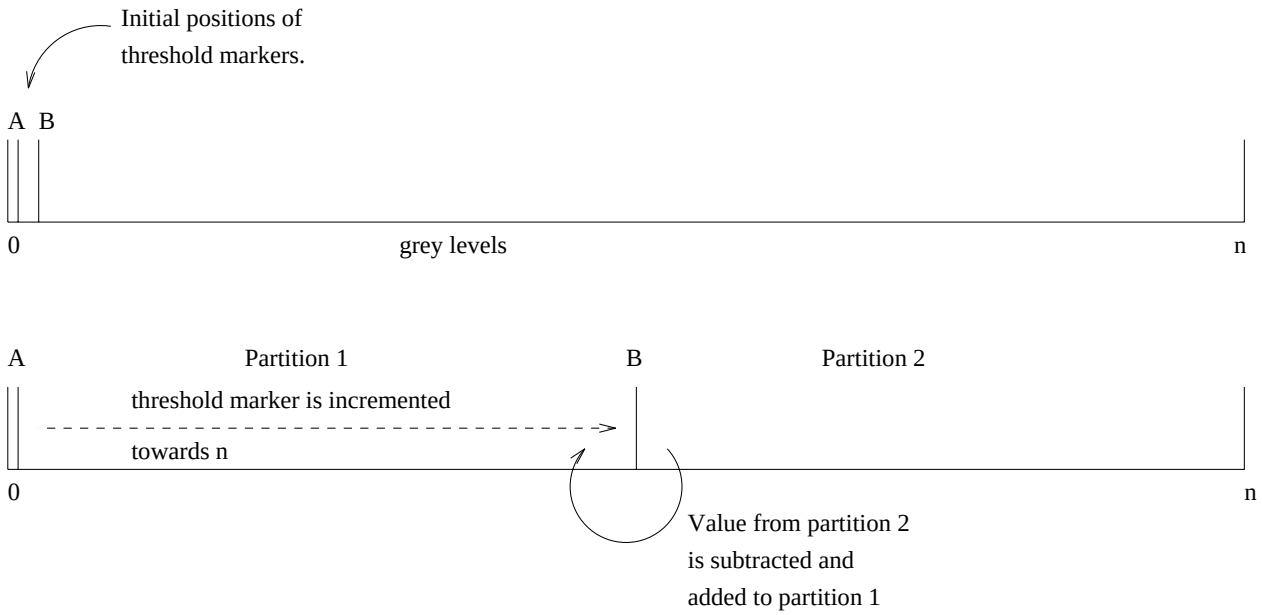

Figure 8: The Running Calculation Method

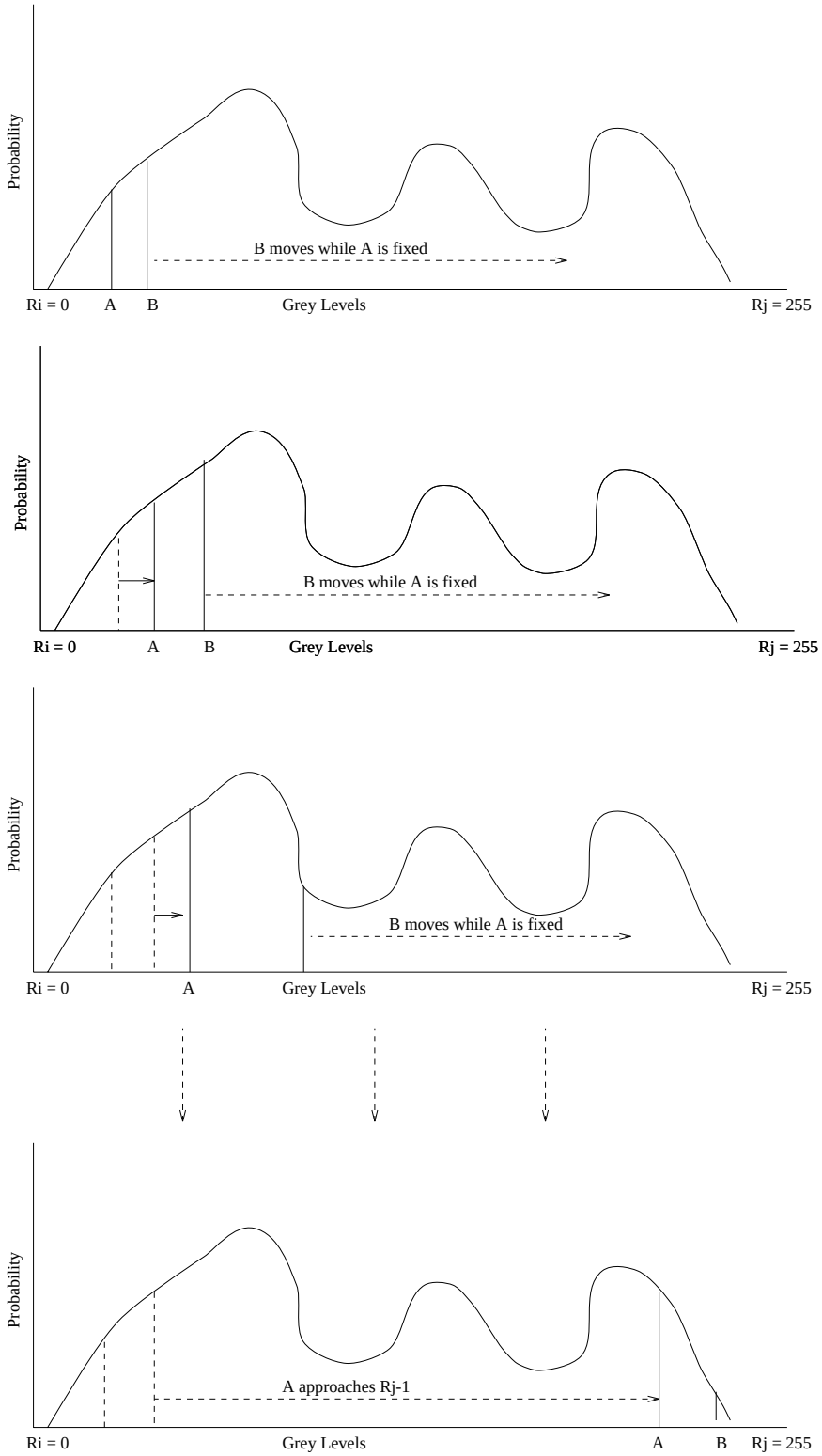

Figure 7: Finding the Optimum Partition for a Multi-Modal Thresholding Function

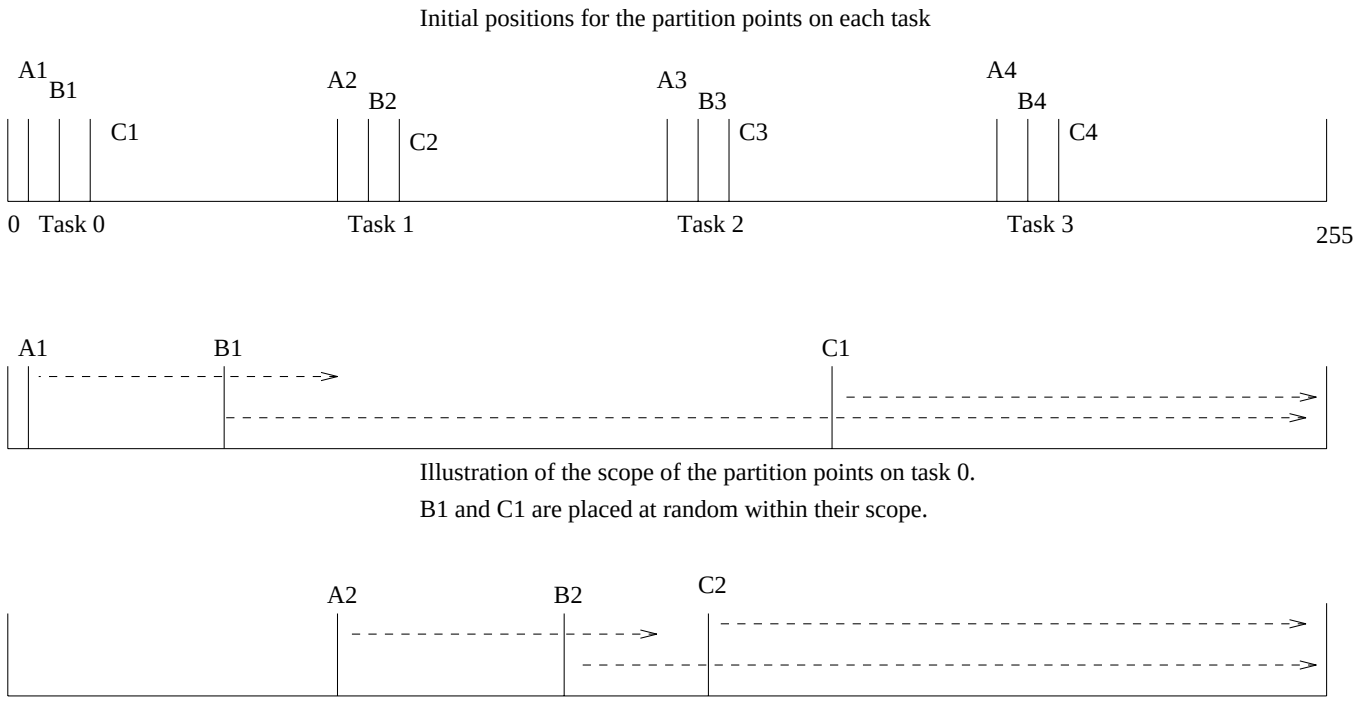

Figure 9: Parallel Decomposition for Calculating a Partitioning Function

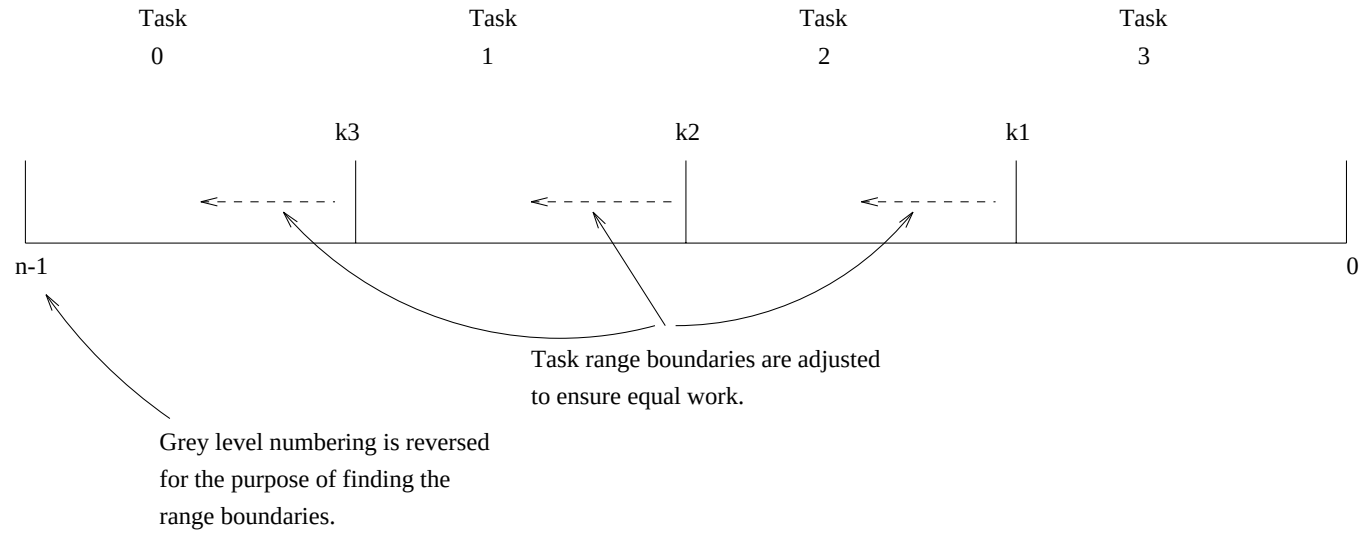

Figure 10: Layout for Load-balancing the Parallel Decomposition

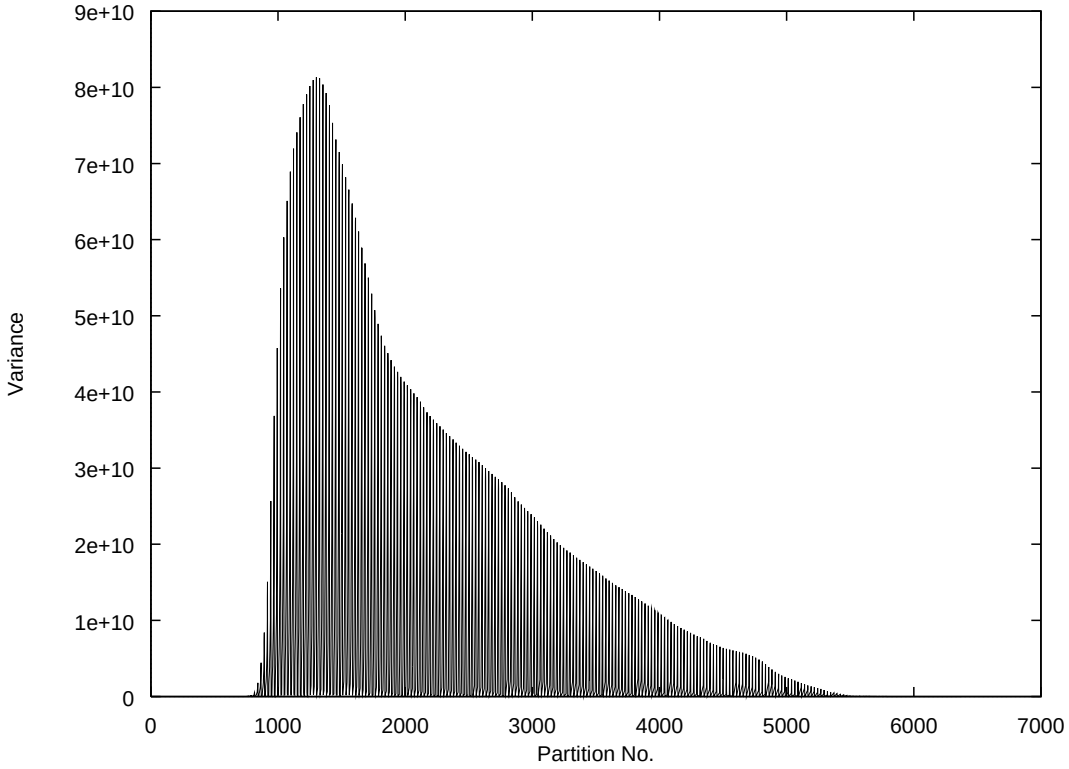

Figure 11: The Form of the Variance-based Function

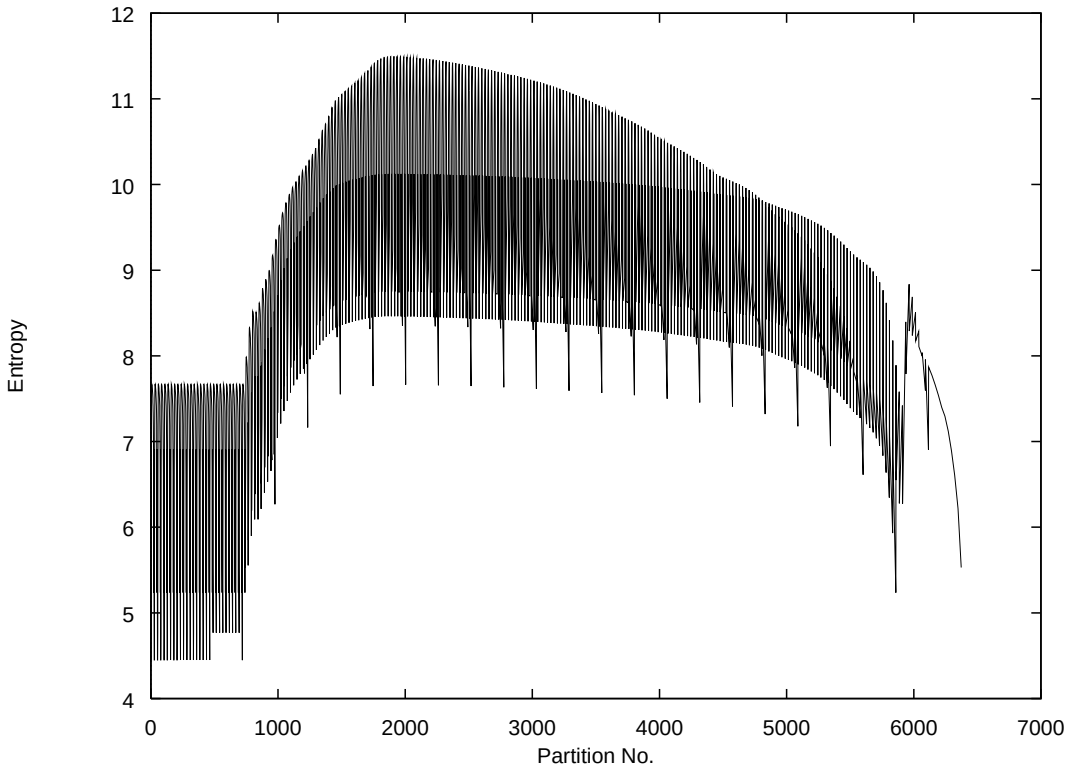

Figure 12: The Entropy Function for Three Segments in Order of Calculation

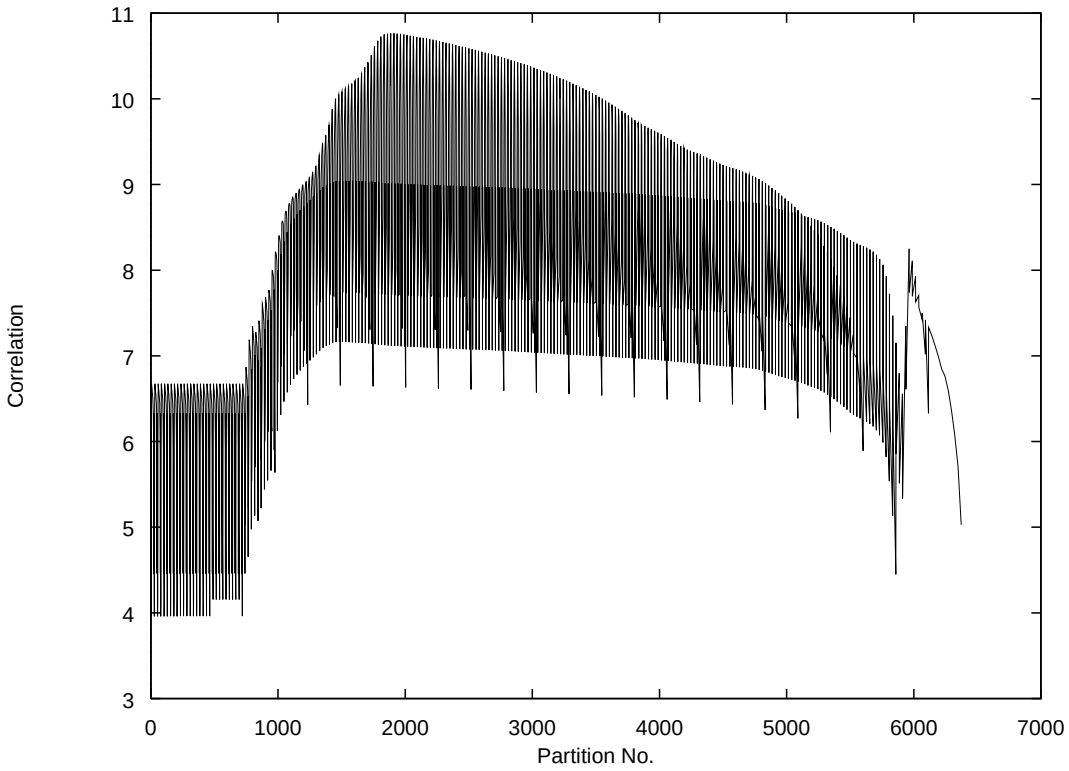

Figure 13: The Correlation Function for Three Segments in Order of Calculation

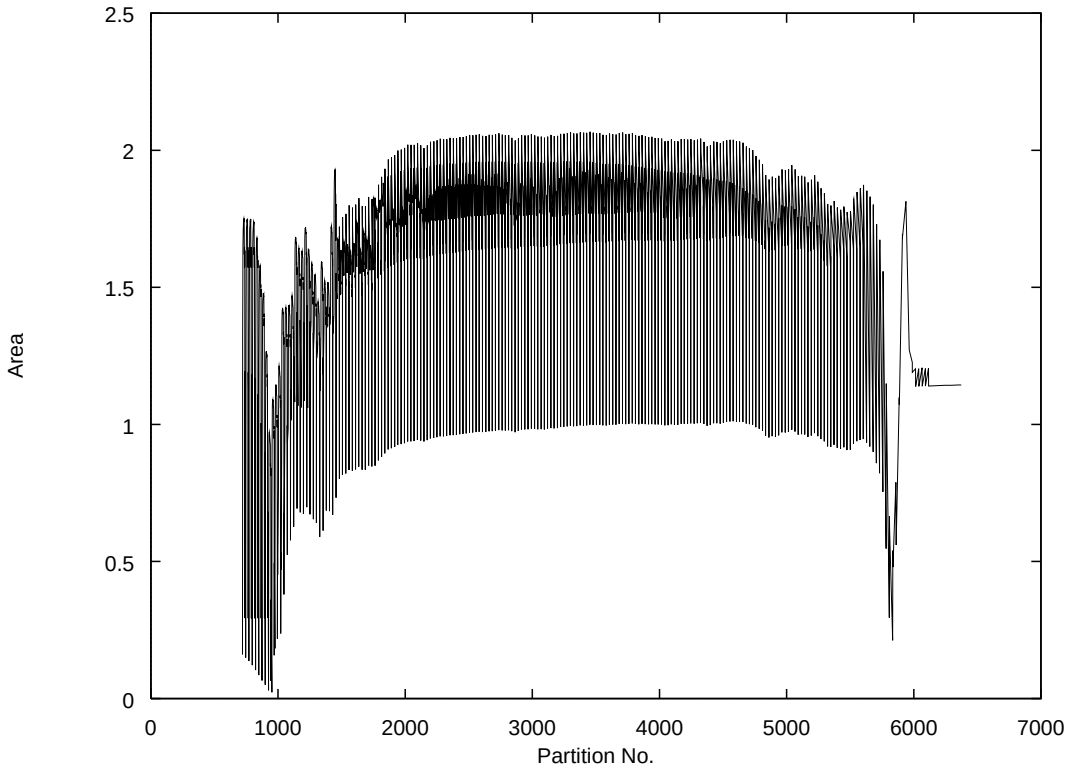

Figure 14: The Area Function for Three Segments in Order of Calculation

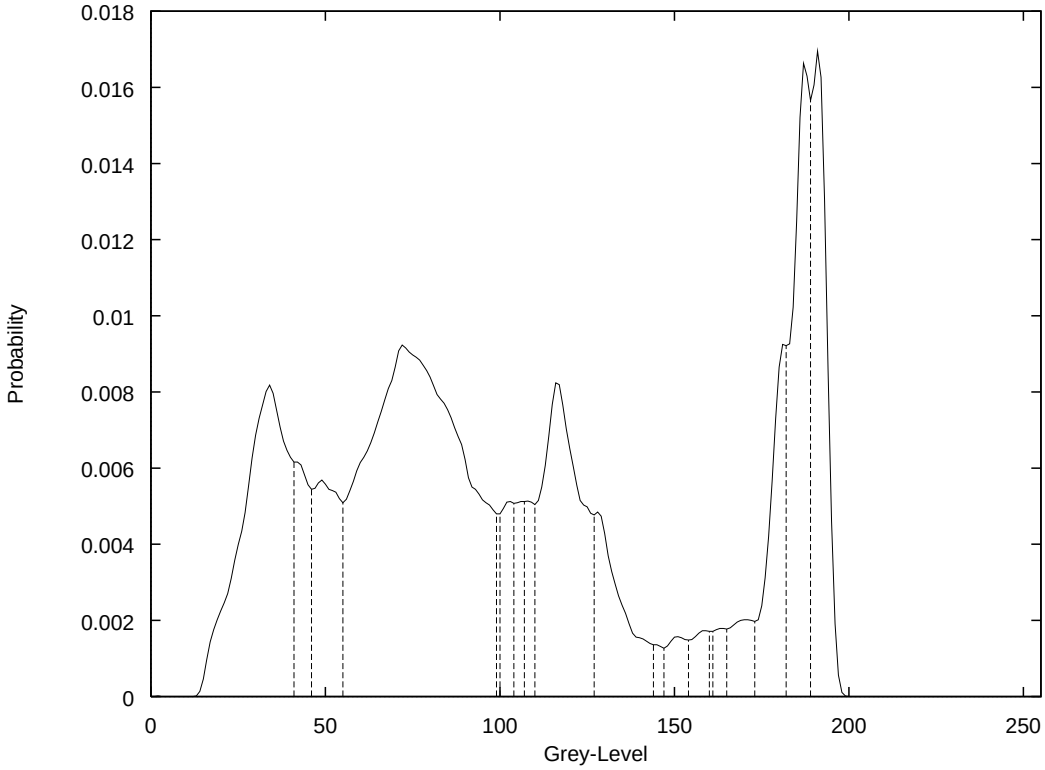

Figure 15: Histogram Concavity Set used for Area-based Function

Face 1 Image, Size 256×256

After Maximum Entropy Thresholding

After Correlation Function Thresholding

After Variation-based Threshold

Figure 16: Results for Face 1

After Area-based Thresholding

After Mean Thresholding

After Fuzzy Entropy Thresholding

Figure 17: Further Results for Face 1

Face 2 Image, Size 720×576

After Maximum Entropy Thresholding

After Correlation Function Thresholding

After Variation-based Threshold

Figure 18: Results for Face 2

After Area-based Threshold

After Mean Thresholding

After Fuzzy Entropy Thresholding

Figure 19: Further Results for Face 2

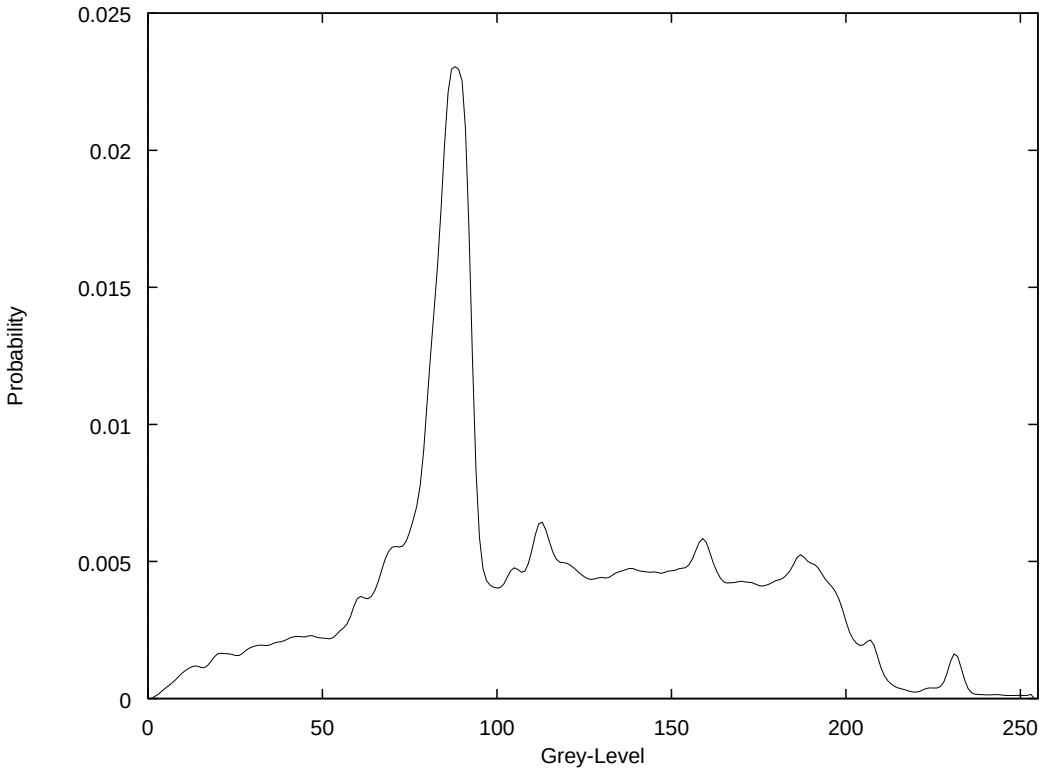

Figure 20: Histogram of Face 2

Lena Image, Size 256×256

After Maximum Entropy Thresholding

After Correlation Function Thresholding

After Variation-based Threshold

Figure 21: Results for Lena Image

After Area-based Threshold

After Mean Thresholding

After Fuzzy Entropy Thresholding

Figure 22: Further Results for Lena Image

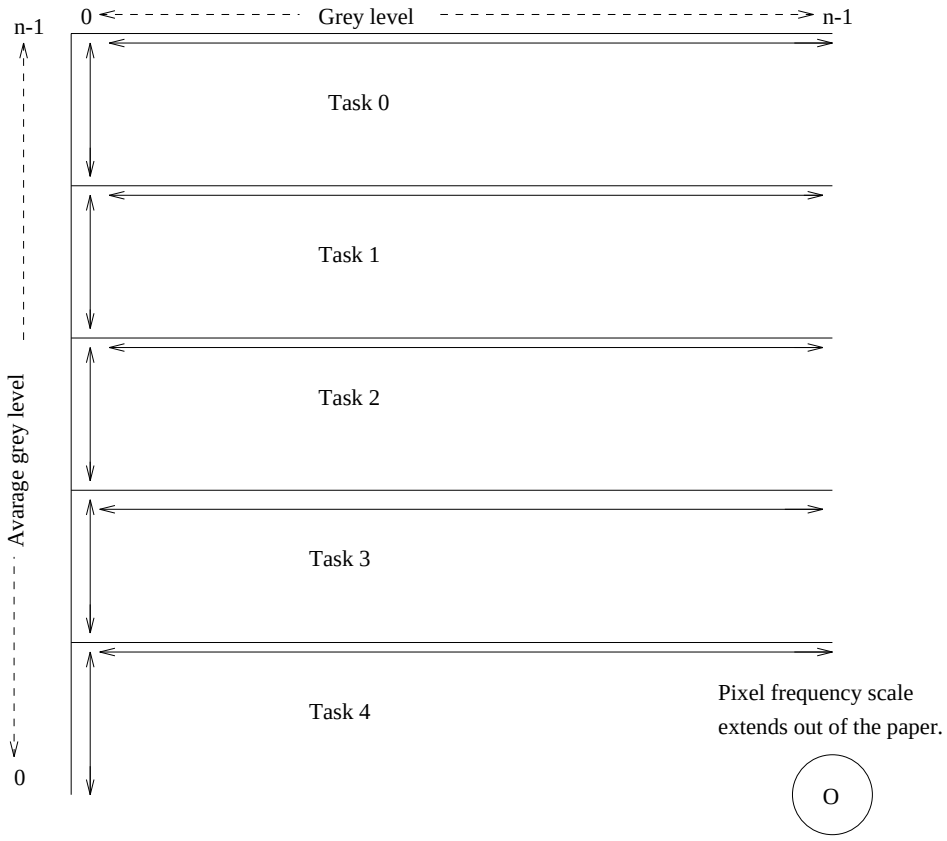

Figure 23: Equal-range Parallel Decomposition of 2D-Entropy Histogram Search for Five Tasks

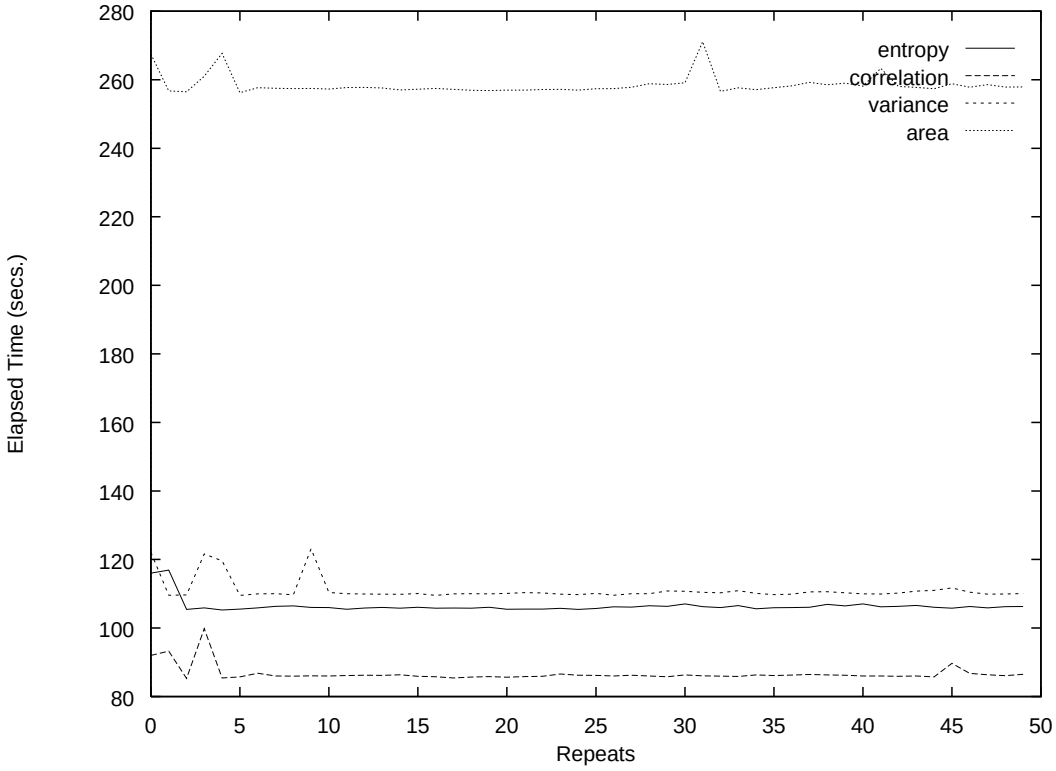

Figure 24: Repeated Timings for Different Thresholding Functions

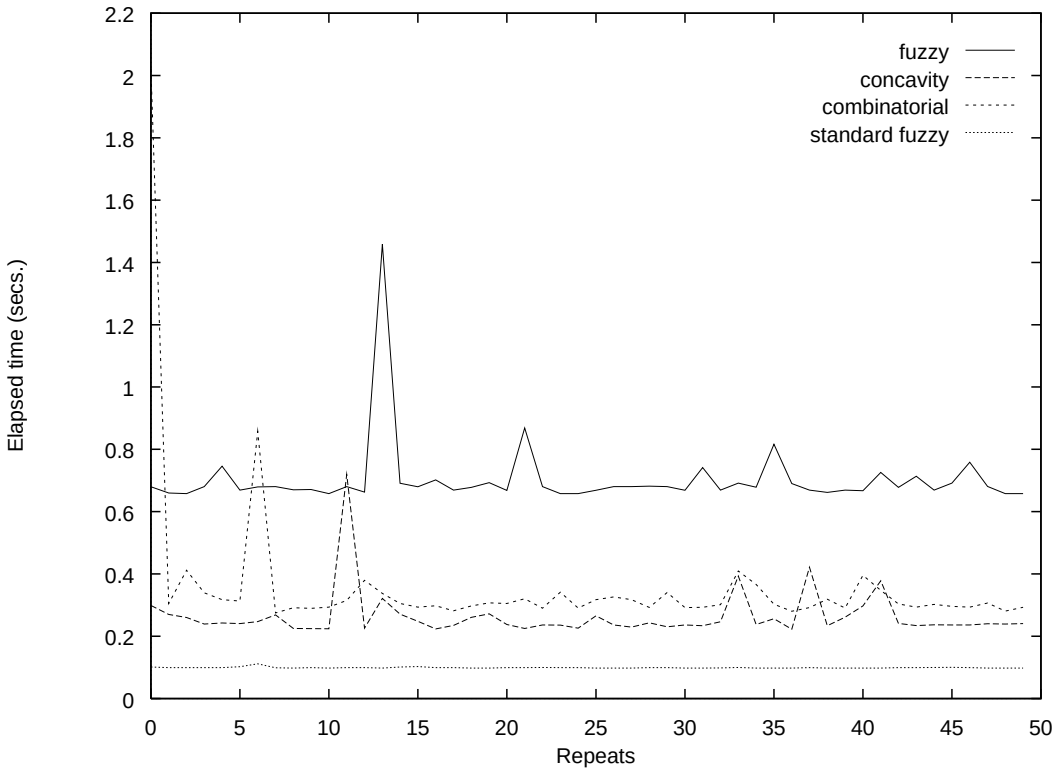

Figure 25: Repeated Timings for Two Methods of Finding the Area Function Thresholds

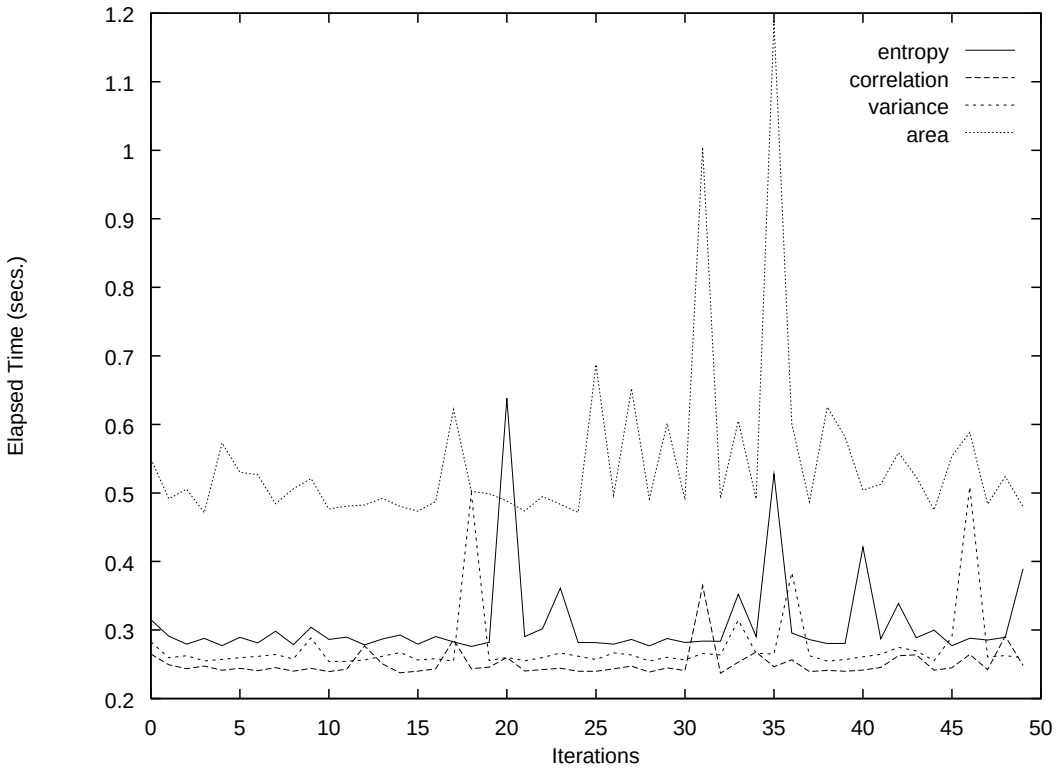

Figure 26: Repeated Timings for a PVM Implementation of Different Thresholding Functions, Using Four Processors and Two Threshold Points