ICEE2006 14th Iranian Conference on Electrical Engineering

MULTI–LAYER REPRESENTATION OF GRAYSCALE IMAGES AND ITS GENERALIZATION

Arash Abadpour

Sharif University of Technology
Mathematics Science Department
P.O. Box 11365-9517, Tehran, Iran
abadpour@math.sharif.edu

Shohreh Kasaei

Sharif University of Technology
Computer Engineering Department
P.O. Box 11365-9517, Tehran, Iran
skasaei@sharif.edu

ABSTRACT

Thresholding is one of the earliest concepts developed in the image processing community. Many researchers have worked on the generalization of the thresholding problem as multithresholding. However, multithresholding only increases the number of threshold values used in the process. Here, we define multithresholding as computing layers which add up together to give better approximations of the given image. The paper contains comprehensive mathematical formulation of the problem and a proper solution to it. Simulation results show the efficiency of the proposed algorithm.

Keywords : Image Segmentation, multi–layer segmentation, Adaptive Thresholding.

1. INTRODUCTION

Thresholding is one of the most fundamental primitives of grayscale image processing. Accordingly, different approaches for *threshold selection* already exist in the literature [1, 2]. Using a priori knowledge about the share of each class in the image [3] or finding the deepest valley in the histogram [4] are two simple examples from the sixties. More recent approaches incorporate the edge information to the threshold–decider [5] and fit parametric curves to the histogram [6].

There are a few approaches that concern multilevel thresholding. As the generalization of single–level thresholding, the multilevel thresholding approaches work on finding a set of thresholds for converting an image into an index array. Two representative examples of these approaches are [7, 8].

In this paper we propose a novel method for multilevel thresholding of an image which results in a multi–layer representation of an image. The proposed method extracts layers from an image which are combined step by step to give a better approximation of the given image. To the best knowledge of the authors, this approach is novel.

2. PROPOSED METHOD

Assume that n realizations of the arbitrary random variable X are given as $S_X = \{x_1, \cdots, x_n\}$, $x_i \in R$. A multi–layer segmentation of S_X is a set of binary planes and corresponding multipliers. These parameters should be selected in a way that they give a proper approximation of the original signal. Below, we will discuss this definition in full details.

Lets work on the single–layer segmentation. For the sequence S_X, the single layer segmentation is a same–sized binary sequence $G_{1X} = [g_{11}, g_{12}, \cdots, g_{1n}]$ and two real values r_1 and s_1. Where, r_1 and s_1 are two real values. As such, the first approximation of x_i is denoted by $\tilde{x}_{1i}$ and is defined as,

$$\tilde{x}_{1i} = g_{1i} r_1 + (1 - g_{1i}) s_1. \tag{1}$$

Equation (1) shows that in the first layer, the sequence is quantized into two levels. Each quantization level is then assigned by a single representative. The above single–layer segmentation should be performed in a way that results in the least possible distortion. Hence, to obtain proper values of r_1, s_1, and g_{1i}, one should minimize the objective function of,

$$\Delta_1 = \sum_{i=1}^{n} (x_i - \tilde{x}_{1i})^2 = \sum_{i=1}^{n} \big(x_i - g_{1i} r_1 - (1 - g_{1i}) s_1\big)^2. \tag{2}$$

Computing the derivative of Δ_1 in terms of r_1 gives,

$$\frac{\partial \Delta_1}{\partial r_1} = -2 \sum_{i=1}^{n} g_{1i} \big(x_i - g_{1i} r_1 - (1 - g_{1i}) s_1\big). \tag{3}$$

Using the fact that $g_{1i}(1 - g_{1i})$ is always zero and $g_{1i}^2 = g_{1i}$, we get,

$$r_1 = E_{g_{1i}=1}\{x_i\}. \tag{4}$$

Similarly, computing $\frac{\partial \Delta_1}{\partial s_1}$ gives,

$$s_1 = E_{g_{1i}=0}\{x_i\}. \tag{5}$$

Here, $E_{c_i}\{x_i\}$ denotes the expectation of x_i for all those i that satisfy c_i, where c_i is a boolean condition.

Note that the computation of r_1 and s_1, does not solve the problem. Because these values depend essentially on the layer definition (G_{1X}). To solve this recursion, select an arbitrary value of i. The share of i in Δ_1 is δ_{1i} defined as,

$$\delta_{1i} = \big(x_i - g_{1i}r_1 - (1-g_{1i})s_1\big)^2. \tag{6}$$

Hence, g_{1i} should be selected in a way that δ_{1i} gets minimum. As $g_{1i} \in [0,1]$ we have to only check whether,

$$g_{1i} = 1 \leftrightarrow (x_i - r_1)^2 \leq (x_i - s_1)^2. \tag{7}$$

or,

$$g_{1i} = 1 \leftrightarrow x_i \leq \frac{1}{2}(r_1 + s_1). \tag{8}$$

Assuming that $r_1 < s_1$. Substituting (8) in (4) and (5) gives,

$$E_{x_i \leq \frac{1}{2}(r_1+s_1)}\{x_i\} = r_1, \tag{9}$$

$$E_{x_i > \frac{1}{2}(r_1+s_1)}\{x_i\} = s_1, \tag{10}$$

which should be solved simultaneously. We combine (9) and (10) to have,

$$E_{x_i \leq \frac{1}{2}(r_1+s_1)}\{x_i\} + E_{x_i > \frac{1}{2}(r_1+s_1)}\{x_i\} = r_1 + s_1. \tag{11}$$

Now, substituting $\theta = \frac{1}{2}(r_1 + s_1)$ in (11), we have,

$$E_{x_i \leq \theta}\{x_i\} + E_{x_i > \theta}\{x_i\} = 2\theta. \tag{12}$$

Hence, we are seeking for the zero point of,

$$f(\theta) = \frac{1}{2}\Big(E_{x_i \leq \theta}\{x_i\} + E_{x_i > \theta}\{x_i\}\Big) - \theta. \tag{13}$$

Note that for an arbitrary sequence of realizations of a random variable we have,

$$\lim_{x \to \min\{X\}} E_{x_i \leq x}\{x_i\} = \min\{X\}, \tag{14}$$

and

$$\lim_{x \to \max\{X\}} E_{x_i > x}\{x_i\} = \max\{X\}. \tag{15}$$

Hence,

$$f\Big(\min\{X\}\Big) = \frac{1}{2}\Big(\bar{X} - \min\{X\}\Big) \geq 0. \tag{16}$$

Here, $\bar{X}$ denotes the expectation of the sequence of values of X. Also, we have,

$$f\Big(\max\{X\}\Big) = -\frac{1}{2}\Big(\max\{X\} - \bar{X}\Big) \leq 0. \tag{17}$$

Note that (16) and (17) turn into equalities when and only when all the values in the sequence X are identical. Neglecting this impractical situation which results in $f(\theta) \equiv 0$, we have,

$$f\Big(\max\{X\}\Big) f\Big(\min\{X\}\Big) < 0. \tag{18}$$

Hence, working in the interval $[\min\{X\}, \max\{X\}]$, to which θ should be a member of, bi–section gives a proper approximation of the value of θ giving $f(\theta) \simeq 0$. Note that in (13), $f : R \to R$ is a continuous function.

Just to perform a brief review, the method of bi–section for finding a zero of the continuous function $f : R \to R$ in the interval $[a, b]$ given that $f(a)f(b) < 0$ is as follows; First, $c = \frac{1}{2}(a+b)$ is computed. If $f(c)$ is zero, the goal is reached. Otherwise, one and only one of the two inequalities $f(c)f(a) < 0$ and $f(c)f(b) < 0$ should hold. Now, the search is repeated in the interval $[a, c]$ or $[c, b]$. The algorithm stops when the length of the working interval is less than a preselected threshold. Here, we set the minimum interval length, which determines the precision of θ, equal to 2^{-r} multiplied by the original range of the sequence. A nominal value of r is 10 which results in a precision of approximately 0.1% of the interval length. The method of bi–section is proved to linearly converge, as the error in each stage is half the error in the last stage. Assuming that the length of the beginning search interval equals α, after j iterations, it would equal $2^{-j}\alpha$. Hence, using the minimum threshold of $2^{-r}\alpha$ for convergence, the algorithm will stop not after r iterations. With the algorithm computing $f(\theta)$ only for interval margins (one computation of f at each iteration), the total number of times the function f is computed equals $r + 2$. For the nominal value, this is only twelve times computation of f.

Returning to the original problem, each times computation of the function f in our case, means performing n comparisons and n additions. Hence, the price of finding θ by the bi–section approach equals $2n(r+2)$ flops (6.3 Mega flops for a 512×512 image while r equals 10, elapsing 6ms on a one–Giga–flops processor). Here, we have assumed that the original image is used for computing θ. Though, the wise idea would be to use a down–sampled version. Now, assume that we have down–sampled the original signal by a factor of λ before feeding it to the procedure (which finds the optimum θ). It is clear that reducing the number of data realizations may only (rarely) decrease the number of iterations passed before convergence. Hence, after down–sampling the cost of finding the optimum θ equals, $\frac{2}{\lambda}n(r+2)$. As such, by increasing λ and decreasing r, one can decrease the computational cost, drastically. In Section 3, we will perform a through investigation to obtain practical values for λ and r. We emphasize that this down–sampling scheme is not useful in the areas where the realizations of the random variable are not correlated in the spatial (or time) domain. It means that for a set of data points gathered from the values of a (computer generated) random sequence, performing the down–sampling will result in data–loss. Though, the reader should be aware that images are highly correlated in the spatial domain and hence act perfectly in the proposed method.

There is another interpretation for the process performed at the above. Note that PSNR depends on the amount of distortion, computed in terms of the mean square difference between the original and the approximated signal. Hence, the one–layer segmentation produced here, is the *best* one–bit representation of the given image in the mean square sense.

Now that the mathematics is developed, we turn into the original problem of multi–layer segmentation. Having the sequence $X = [x_1, \cdots, x_n]$, the m–layer segmentation is the set of n binary planes $G_m = [g_{11}, \cdots, g_{1n}], \cdots, [g_{m1}, \cdots, g_{mn}]$ plus $2n$ real values of $R_m = [r_1, \cdots, r_m]$ and $S_m = [s_1, \cdots, s_m]$. We define,

$$\tilde{x}_{m,i} = \sum_{j=1}^{m} \Big(g_{j,i}r_j + (1-g_{j,i})s_j\Big), \tag{19}$$

as the m–layer representation of x_i. Note that (1) is a special case of (19) with $m = 1$. Now, the objective function for fitting $x_{k,i}$s into x_is changes to,

$$\Delta_m = \sum_{i=1}^{n} \Big(x_i - \sum_{j=1}^{m} \big(g_{j,i}r_j + (1-g_{j,i})s_j\big)\Big)^2. \tag{20}$$

We call the combination of G_m, R_m, and S_m as a proper m–layer segmentation of X if for all $1 \leq k \leq m$, G_k, R_k and S_k give a proper k–layer segmentation of X (note that here X denotes a sequence of values of a random variable). This recursive definition implies that for finding the $(m+1)$–layer representation of X, one should first find its m–layer representation. Rewriting (19) as,

$$\tilde{x}_{k,i} = x_{k-1,i} + g_{k,i} r_j + (1 - g_{k,i}) s_k, \qquad (21)$$

and (20) as,

$$\Delta_k = \sum_{i=1}^{n} \Big((x - \tilde{x}_{k-1,i}) - g_{k,i} r_k - (1 - g_{k,i}) s_k \Big)^2, \qquad (22)$$

proves that the k–th layer is essentially the one–layer segmentation of the error of the $k-1$–layer representation. From this it proceeds that the process of finding the m–layer segmentation of a sequence of n realization (call it X) of a random variable is as follows: find the one–layer representation of X and call it X_1. Set $X = X - X_1$ and find the one–layer representation of the new X. Call the result X_2. Set $X = X - X_2$ and proceed to m layers.

The computational cost of obtaining the m–layer representation of a sequence of n values using the down–sampling ratio of λ and the precision of 2^{-r} equals $m(\frac{2}{\lambda} n(r+2) + 2n)$. For nominal values of $r = 10$ and $\lambda = 200$, the multiplier $\frac{2}{\lambda}(r+2)$ is less than $\frac{1}{8}$. Hence, the computational cost of the proposed method almost equals $2nm$. Here, n is the number of data points and m is the number of layers. As such, the cost of the proposed method is linear both in terms of the number of data points and the number of layers. In an ordinary practice, we may need to compute the eight–layer representation of a 512×512 image. This operation takes 4.2 Mega flops, elapsing less than $5ms$ on a one–Giga–flop processor.

3. EXPERIMENTAL RESULTS

Figure 1–a, b, and c show *Lena* image, its 50–bin PDF, and the values of $f(x)$, respectively. Here, the optimum θ is approximated as 104.7. Figure 1–d shows the one–layer segmentation of *Lena* using $r_1 = 72.2$ and $s_1 = 150.8$. The PSNR is about $20dB$.

Figure 2 examines 512×512, $8bpp$ *Peppers* image to find the acceptable values of the precision and λ (down–sampling ratio of the signal for reducing the cost of finding θ). Figure 2–a shows that selecting the precision of 2^{-4} is an acceptable compromise. Note that larger values of $-log_2(precision)$ tend to unwanted decrease of PSNR. Figure 2–b shows that reducing the number of data points by a factor of 1000 has no great influence on the PSNR. Hence, according to the fact that having a large number of realizations, greatly declines the performance of finding θ in the coming parts of this paper we down–sample he signal before feeding it to the routine which finds the optimum θ. Also, we let the routine to stop making θ more precise when it is working in an interval less than $\frac{1}{16}$ of the original interval. With this settings, the cost of computing the optimum θ declines to 3200 flops (about $3\mu s$ in a one–Giga–flop processor). Using non–optimized MATLAB code, it takes $28ms$ to perform the operation. From which, $22ms$ is elapsed on performing the down–sampling (on a PIV with clock frequency of $2871MHz$).

Figure 3 shows the results of applying the proposed method on a 1–D signal. This signal is actually a line of a 512×512 image. The value of SNR increases as $23dB$, $27dB$, $31dB$, $34dB$, and $39dB$ as the number of layers increases from one to five, respectively. We believe that this figure clearly depicts the scope of the proposed method.

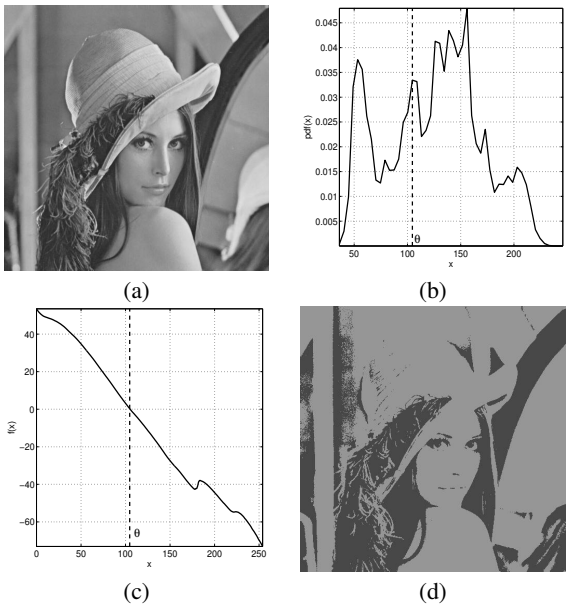
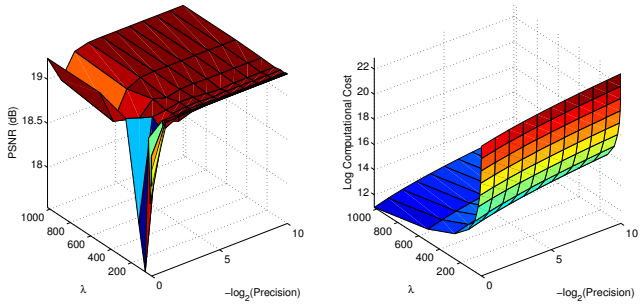


Figure 1: (a) *Lena* image. (b) PDF of (a). (c) Values of the function f defined in (13). (d) Single–layer segmentation result (PSNR=$19.8dB$).

Figure 4 shows *unr730* image. The proposed method is applied on this image to produce the number of layers that give a proper representation with $PSNR \geq 40dB$. As such, seven layers have been produced (here, we have only shown the results of five layers, the two others are not visually recognizable). Note the results of adding more and more layers. While increasing the PSNR, adding new layers also results in an image with more details. Figure 5 shows PSNR versus layer number for this image.

Here, we compare the proposed method with one of the best available single–layer thresholding approaches. As an outstanding approach, *Otsu* proposed a method for finding the optimal threshold of an image containing a bright object on a dark background [9]. The *Otsu* method divides a double–peak histogram into two compact classes which are well–separated. The above method differs from the proposed method in many major points. While the *Otsu* method only works for a specific class of images, it focuses on the intra– and inter–specifications of the classes. In contrast, the proposed method gives a proper estimation of any arbitrary image. Also, *Otsu*'s approach results in a binary (single–layer) representation of an image, while the proposed method is able to give the representation of the given image in any number of layers. In the computational side, both the *Otsu*'s method and the proposed method rely on analytical search for a proper threshold. While the proposed method utilizes a logarithmic–order search algorithm, the *Otsu*'s approach searches an interval from the beginning bin–by–bin to find the optimum threshold.



(a) (b)

Figure 2: (a) PSNR of a single–layer segmentation of *Peppers* image with varying λ and r. (b) Logarithm of the computational cost for different values of λ and r.

4. CONCLUSIONS

A new multi–layer segmentation problem is defined and its practical meaning is described. Also, a novel approach for solving the above problem is proposed. The proposed method acts on a grayscale image to produce sequential layers. Adding these layers one after another gives successive approximations of the image with increasing qualities. The proposed method is proved to be both efficient and effective.

Acknowledgement

This work was in part supported by a grant from ITRC. The first author wishes to thank Ms. *Azadeh Yadollahi* for her encouragement and invaluable ideas.

A. GENERALIZATION

Assume that $x_1, \cdots, x_n \in R$ are given. Also, assume that we need approximate x_is using a single layer. As such, the problem is to find the values $g_1, \cdots, g_n \in \{0, 1\}$ and the two real values r and s, for which $\tilde{x}_{1i} = rg_i + s(1 - g_i)$ is an appropriate approximation. Here, appropriateness is measured in terms of the following objective function which is to minimized,

$$\Delta = \sum_{i=1}^{n} f\big(x_i - rg_i - s(1 - g_i)\big). \tag{23}$$

Where, $f : R \to R^+ \cup \{0\}$ is a continuous differentiable function which is increasing in $R^+ \cup \{0\}$. First note that $\forall x \in R, f(x) = f(|x|)$. For an arbitrary value of i we have,

$$g_i = 1 \leftrightarrow f(|x_i - r|) \leq f(|x_i - s|), \tag{24}$$

or equivalently,

$$g_i = 1 \leftrightarrow |x_i - r| \leq |x_i - s|. \tag{25}$$

Assuming $s > r$, (25) turns into,

$$g_i = 1 \leftrightarrow x_i \leq \frac{1}{2}(r + s). \tag{26}$$

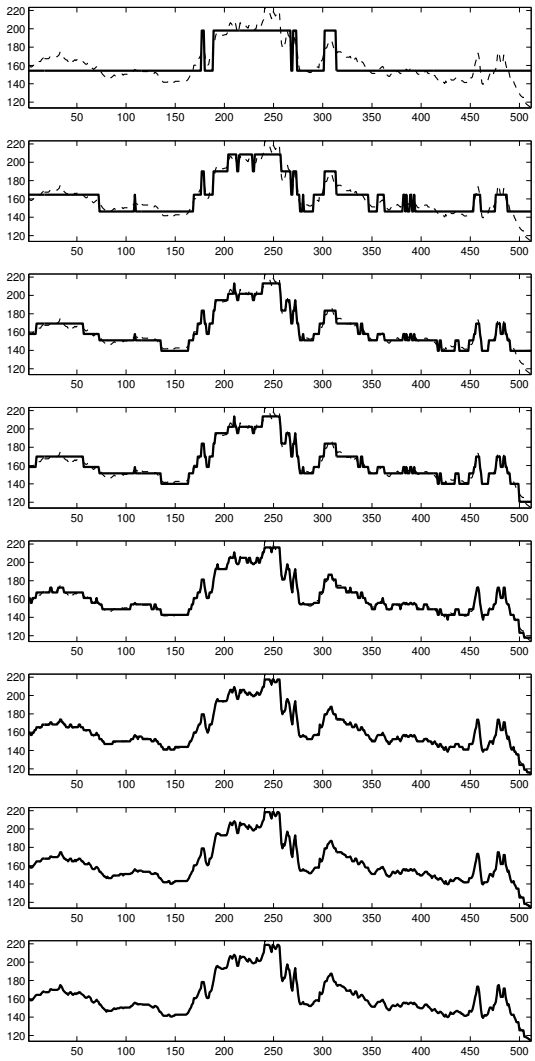


Figure 3: Results of applying the proposed method on a 1–D signal.

Now, lets find the values of r and s. We have,

$$\frac{\partial \Delta}{\partial r} = -\sum_{x_i \leq \frac{r+s}{2}} f'(x_i - r) = 0, \tag{27}$$

$$\frac{\partial \Delta}{\partial r} = -\sum_{x_i > \frac{r+s}{2}} f'(x_i - s) = 0. \tag{28}$$

For an arbitrary function $g : R \to R$ we define,

$$E_g\{x\} \equiv \arg_{r \in R} \sum_{x \in X} g(x - r) = 0. \tag{29}$$

For example, for $g(x) \equiv x$, E_g turns into the conventional average. Using this notation, (27) and (28) are converted to,

$$E_{f'}\{x_i \leq \theta\} + E_{f'}\{x_i > \theta\} = 2\theta, \tag{30}$$

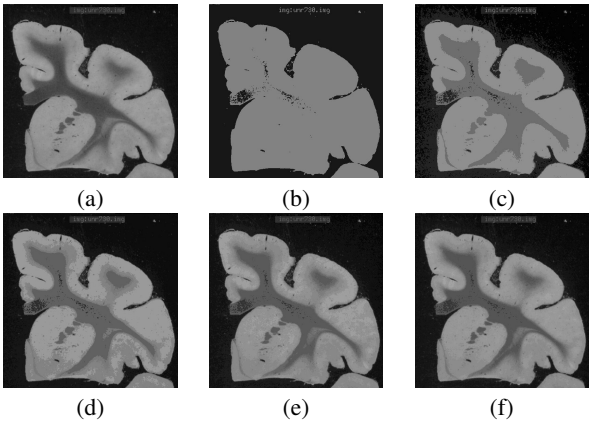


Figure 4: Results of applying the proposed method on *unr730* image. (a) Original image adopted from *http://ipagwww.med.yale.edu/∼staib*. (b) Single–layer representation. (c) Two–layer representation. (d) Three–layer representation. (e) Four–layer representation. (f) Five–layer representation.

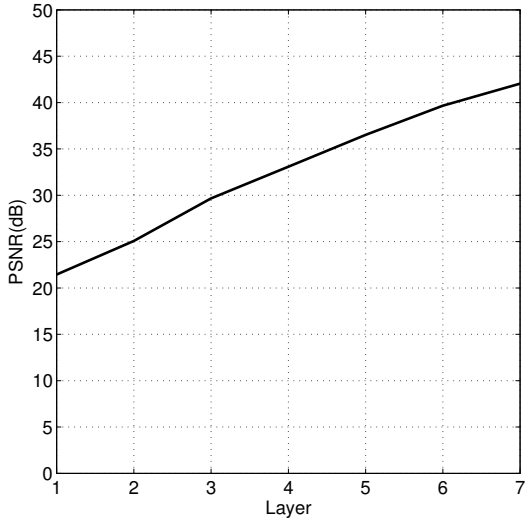


Figure 5: PSNR versus layer number for *unr730* image.

solution of what gives the appropriate $\theta = \frac{r+s}{2}$.

Lets try to compute $E_g(X)$ for an arbitrary function g and a set X. Here, we focus on the continuous functions g satisfying $g(-x) = -g(x)$ and $g \geq 0 \rightarrow g(x) \geq 0$. Lets define,

$$G_X(r) = \sum_{x \in X} g(x-r) = \sum_{x \in X, x > r} g(x-r) - \sum_{x \in X, x \leq r} g(r-x). \tag{31}$$

Now,

$$G_X\Big(\min\{X\}\Big) > 0, G_X\Big(\max\{X\}\Big) < 0. \tag{32}$$

Hence, using bi–section, the value of r satisfying $G_X(r) = 0$ is approximated. Returning to the main problem, we were faced with an odd function so,

$$f'(-x) = \lim_{\Delta x \to 0} \frac{f(-x+\Delta x) - f(-x)}{\Delta x} = \\ - \lim_{\Delta x \to 0} \frac{f(x-\Delta x) - f(x)}{-\Delta x} = -f'(x). \tag{33}$$

As for $x \geq 0$, $f'(x)$ is positive, the method of bi–section will find the value of $E_{f'}\{X\}$ for any set $X \subset R$. For the set $X \subset [a,b]$ for finding $E_{f'}\{X\}$ with the precision of $2^{-\alpha}(b-a)$, α operations would be enough. Denoting the cost of computing f' by τ, the cost equals $(\alpha+2)n\tau$.

Lets return to the main problem of solving (30). Thus, we are seeking for the zero point of the function,

$$F_X(\theta) = E_{f'}\{x_i \leq \theta\} + E_{f'}\{x_i > \theta\} - 2\theta. \tag{34}$$

Note that,

$$\lim_{\theta \to \min\{X\}^+} E_{f'}\{x \in X, x < \theta\} = \min\{X\}, \tag{35}$$

and

$$\lim_{\theta \to \max\{X\}^-} E_{f'}\{x \in X, x > \theta\} = \max\{X\}. \tag{36}$$

Hence,

$$F_X\Big(\min\{X\}\Big) = E_{f'}\{x \in X\} - \min\{X\}, \tag{37}$$

$$F_X\Big(\max\{X\}\Big) = E_{f'}\{x \in X\} - \max\{X\}, \tag{38}$$

We claim that for every function f with the conditions described above,

$$\min\{X\} \leq E_{f'}\{x \in X\} \leq \max\{X\}. \tag{39}$$

Note that if (39) is proved then $F_X\Big(\min\{X\}\Big) \geq 0$ and $F_X\Big(\max\{X\}\Big) \leq 0$ and so finding the zero of F_X would be possible using bi–section. To prove (39) note that,

$$r \leq \min\{X\} \rightarrow x_i - r \geq 0 \rightarrow f'(x_i - r) \geq 0 \rightarrow \\ \sum_{x \in X} f'(x-r) \geq 0, \tag{40}$$

$$r \geq \max\{X\} \rightarrow r - x_i \geq 0 \rightarrow f'(r - x_i) \geq 0 \rightarrow \\ \sum_{x \in X} f'(x-r) \leq 0. \tag{41}$$

Hence, the zero point of $\sum_{x \in X} f'(x-r)$ should be in the interval $[\min\{X\}, \max\{X\}]$, proving (39).

For a specific value of θ to compute (30) one should compute both $E_{f'}\{x_i \leq \theta\}$ and $E_{f'}\{x_i > \theta\}$. Assuming that $X \subset [a,b]$ and that $\theta = a\beta + b(1-\beta)$ $(0 \leq \beta \leq 1)$ finding the two above measures with the precision of $2^{-\alpha}$ multiplied by interval length elapses,

$$(\alpha+2)n\tau\beta + (\alpha+2)n\tau(1-\beta) = (\alpha+2)n\tau, \tag{42}$$

flops (assuming that the points are almost uniformly spread). For finding θ with the precision of $2^{-\alpha}(b-a)$ the total number of flops will get $(\alpha+2)^2 n\tau$.

B. REFERENCES

[1] J. S. Weska, "A survey of threshold selection techniques," *Computer Graphics and Image Processing*, vol. 7(2), pp. 259–265, 1978.

[2] B. Sankur, A. T. Abak, and U. Baris, "Assessment of thresholding algorithms for document processing," in *Proceedings of the IEEE International Conference on Image Processing*, Kobe, Japan, 1999, pp. 580–584.

[3] W. Doyle, "Operations useful for similarity–invariant pattern recognition," *Journal of Association for Computing Machinery*, vol. 9(2), pp. 259–267, 1962.

[4] J. M. S. Prewitt and M. L. Mendelsohn, "The analysis of cell images," *Annual New York Academy of Science*, vol. 128, pp. 1036–1053, 1966.

[5] J. S. Weska, R. N. Nagel, and A. Rosenfeld, "A threshold selection technique," *IEEE Transaction on Computers*, vol. C–23 (12), pp. 1322–1326, 1974.

[6] N. Papamarkos and B. Gatos, "A new approach for multilevel threshold selection," *CVGIP: Graphical Models and Image Processing*, vol. 56(5), pp. 357–370, 1994.

[7] F. Tomita, M. Yachida, and S. Tsuji, "Detection of homogenous regions by structural analysis," in *Proceedings of the Joint Conference on Artificial Intelligence*, Stanford, CA, 1973, pp. 564–571.

[8] N. Papamarkos, C. Strouthopoulos, and I. Andreadis, "Multithresholding of color and gray–level images through a neural network technique," *Image and Vision Computing*, vol. 18, pp. 213–222, 2000.

[9] N. Otsu, "A threshold selection method from gray level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC–9, pp. 62–66, 1979.