Pattern Recognition, Vol. 22, No. 5, pp. 609–617, 1989
Printed in Great Britain.

0031-3203/89 $3.00 + .00
Pergamon Press plc

IMPROVEMENT OF KITTLER AND ILLINGWORTH'S MINIMUM ERROR THRESHOLDING

SUNGZOON CHO,* ROBERT HARALICK† and SEUNGKU YI‡

*Department of Computer Science, University of Maryland, College Park, MD20742, U.S.A.; †Department of Electrical Engineering, FT-10 and ‡Department of Computer Science, FR-35, University of Washington, Seattle, WA 98195, U.S.A.

(*Received* 2 *May* 1988; *in revised form* 11 *August* 1988; *received for publication* 12 *October* 1988)

Abstract—A simple modification to Kittler and Illingworth's minimum error thresholding method was made and the performance of the modified version was compared with that of the original version empirically. By correcting the biased estimates of variances of model distributions, a significant improvement in performance was found. The improvement was most outstanding among not-well-separated, but still bimodal histograms. In fact, the modification provides a more robust method. The new version is nearly computationally equivalent in complexity to the original version.

Thresholding Histogram Segmentation Minimum error Modes Bimodal distribution Unbiased estimate Image processing

1. INTRODUCTION

Thresholding is one of the most frequently used segmentation technique in image processing. It transforms a grey-level image into a binary image. After a threshold value is determined, the technique converts a pixel's brightness to 0 or 1 according to whether it is smaller than or bigger than the threshold value. There has been much work done about determining a threshold value so that after the thresholding the image consists of nicely segmented regions.[2] Among the methods solely based on histograms, one very good method is found in Ref. (1).

Here we describe a simple modification to Ref. (1), and the improvement in the performance. In the next section, the reason why the modification is needed is explained, and in Section 3, the experimental design is presented. Empirical performance data over a large number of histograms are provided and discussed in Section 4 and a conclusion is drawn in the final section.

2. MODIFICATION

A histogram is generated from a gray level image. For each brightness level, the histogram gives the number of pixels in the image with the brightness level. If the histogram is bimodal, the histogram thresholding problem is to determine a best threshold t separating two modes of the histogram from one another. Before we talk about the modification we made, a brief description about the notion of the best threshold in Kittler and Illingworth's[1] method is presented.

It assumes that the observations come from a mixture of two Normal distributions having respective mean and variances (μ_1, σ_1^2) and (μ_2, σ_2^2) and respective proportions q_1 and q_2. Then the mixture distribution reflected in the histogram takes the form

$$f(i) = \frac{q_1}{\sqrt{2\pi}\sigma_1} e^{-\frac{1}{2}(\frac{i-\mu_1}{\sigma_1})^2} + \frac{q_2}{\sqrt{2\pi}\sigma_2} e^{-\frac{1}{2}(\frac{i-\mu_2}{\sigma_2})^2}.$$

Given a brightness level t as a trial threshold, it models the two resulting pixel populations, one from those pixels whose brightness level is smaller than the threshold, and the other from those pixels whose brightness level is larger than the threshold. The two populations are modeled by Normal distributions $N(\mu_1(t), \sigma_1^2(t))$ and $N(\mu_2(t), \sigma_2^2(t))$, where $\mu_1(t)$, $\mu_2(t)$, $\sigma_1^2(t)$, and $\sigma_2^2(t)$ denote the means and the variances of the populations.

If an image has n brightness levels, we try in succession n different levels as a threshold value. Let $P(0)$, $P(1)$, …, $P(n-1)$ represent the histogram frequencies of the observed brightness values 0, 1, …, $n-1$. For each brightness value t, a fitting criterion $J(t)$ is calculated, which is defined by

$$J(t) = 1 + 2[q_1(t)\log\sigma_1(t) + q_2(t)\log\sigma_2(t)] - 2[q_1(t)\log q_1(t) + q_2(t)\log q_2(t)]$$

where

$$q_1(t) = \sum_{I=0}^{t} P(I)$$

$$q_2(t) = \sum_{I=t+1}^{n-1} P(I).$$

The better the models fit the data, the smaller the criterion. Therefore the t value which minimizes the criterion function is considered to be the best threshold value.

2.1. *Biased means and variances*

The measured means $\mu_1(t)$ and $\mu_2(t)$ and variances $\sigma_1^2(t)$ and $\sigma_2^2(t)$ come from the distributions whose tails are truncated by the threshold value. The distribution on the left has its right tail truncated and the one on the right has its left tail gone. However the distributions in the model are not modeled as being truncated in Kittler and Illingworth's method. The truncation by partitioning does bias the means and the variances we measure and use as the parameters of the models. In fact, the assumption of well separated modes implies that if t is the threshold which separates the modes, the mean and the variance measured from $P(1), \ldots, P(t)$ are close estimates of the true mean and variance for the model. Likewise, so are the mean and variance measured from $P(t+1), \ldots, P(n)$. However, we face a large number of images in practice whose histograms do not have two *well separated modes.* For the rest of this section, we show the way to compute better estimates of the true means and variances. Here only the case of a left side distribution whose right tail is truncated is shown. The case for the right side distribution is similar. We provide only the final result for that.

We assume that a histogram consists of two Normal distributions. The probability distribution function of the left distribution (N, σ^2) whose right tail is truncated at t is given by

$$f(x) = \frac{\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}}{F(t)},$$

where

$$F(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}\,dx.$$

Let the original distribution be $N(\mu, \sigma^2)$ and the truncated distribution be $N(\mu_t, \sigma_t^2)$. Then we get the following equations.

$$\mu = \mu_t + \sigma\frac{\phi'(z)}{\phi(z)} \tag{1}$$

$$\sigma_t^2 = \sigma^2 \mathscr{H} \tag{2}$$

where

$$z = \frac{t-\mu}{\sigma}$$

$$\phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}}\,dx^2$$

$$\mathscr{H} = 1 - \frac{\phi'(z)}{\phi(z)}\left(z + \frac{\phi'(z)}{\phi(z)}\right).$$

Now we want to find μ, σ given t, μ_t and σ_t. By simply replacing z by $z_t = \frac{t-\mu_t}{\sigma_t}$ in (1), we get the following linear equation.

$$\mu = \mu_t + \sigma A \tag{3}$$

where

$$A = \frac{\phi'\left(\frac{t-\mu_t}{\sigma_t}\right)}{\phi\left(\frac{t-\mu_t}{\sigma_t}\right)}.$$

In (2), the inverse square root of $\mathscr{H}$ is nicely approximated to a linear function of μ. By so doing, we get

$$\sigma = \sigma_t(k_0 + k_1\mu) \tag{4}$$

where

$$k_0 = \frac{1}{h} - \mu_t k_1,$$

$$k_1 = \frac{A}{2\sigma_t h^3}(2z_t^2 + 5z_t A + 2A^2 - 1),$$

and

$$h = \sqrt{1 - A(z_t + A)}.$$

If (3) is inserted into (4), and is solved for σ, we have a closed form solution for the updated variance.

$$\sigma^2 = \sigma_t^2 \frac{1}{h^2(1 - \sigma_t A k_1)^2}. \tag{5}$$

For the distribution on the right, we have the following solution. In this case, it is the left tail which is truncated.

$$\sigma^2 = \sigma_t^2 \frac{1}{g^2(1 + \sigma_t B p_1)^2}, \tag{6}$$

where

$$g = \sqrt{1 + B(z_t + B)}$$

$$B = \frac{\phi'\left(\frac{t-\mu_t}{\sigma_t}\right)}{1 - \phi\left(\frac{t-\mu_t}{\sigma_t}\right)}$$

$$p_1 = -\frac{B}{2\sigma_t g^3}(2z_t^2 - 5z_t B + 2B^2 - 1).$$

2.2. *Use of both measured and updated variances*

In the middle of the histogram, where two distributions overlap heavily, we use the measured means and variances. In other words, the truncation tends not to bias the statistics since both of them lose some area by truncation but at the same time get some area by overlapping. When the two distributions have the same variance and are in equal proportion, the

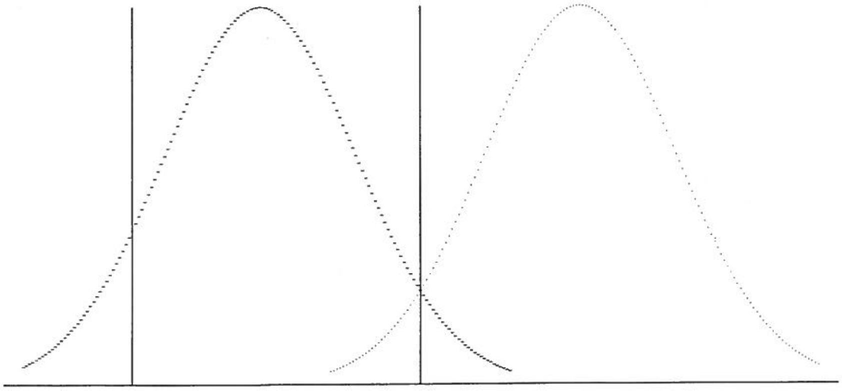

Fig. 1. The effects of truncations.

truncation bias is negligible. This is clearly shown in the Fig. 1.

Let area A represent the portion of the right distribution which is to the left of t_2. Let area B represent that of the left distribution being to the right of t_2. When truncation occurs at t_2, the variance of the population to the left of t_2 is underestimated by the amount of area B, but at the same time, overestimated by the amount of area A. Thus, truncation does not have much impact on the correctness of the mean and the variance. However, when we have t_1 as a truncating line, we do get the biased statistics. The left distribution does not include enough portion of right distribution to offset the truncated portion of itself. Therefore, we have underestimated variances.

Since the measured variances are very close to the true variances in heavily overlapped areas, we use a weighted sum of the updated variances and the measured variances. The less overlapped the area is, the more weight is given to the updated variances. By so doing, we can effectively approximate the correct means and the correct variances.

Now consider the weight parameter. For which t do we use solely the measured variances? For which t do we use solely the updated variances? It is clear that at the left-end and at the right-end, we should use only the updated variances according to the above argument. It is also apparent that we should use only the measured variances at the point where most overlapping occurs. This point corresponds to the exact threshold we are looking for. The problem here is that we do not know this value, but we need to know. The best we can do is to estimate it with some other method like, for example, Otsu's method[3] or even the original Kittler and Illingworth method itself. We call this estimated threshold value a *cutoff point* from now on. At the cutoff point the measured variances are used. The following relation shows what is actually used as a variance for the case of the left distribution,

$$\sigma_{\text{usd}}^2 = \sigma_{\text{msr}}^2 + (\sigma^2 - \sigma_{\text{msr}}^2)\frac{cp - t}{cp}$$

σ_{usd}^2 is the used variance, σ_{msr}^2 is the measured variance, σ^2 is the updated variance from (5), cp is the cutoff point, and t is the truncating point.

For the right distribution, the following is used.

$$\sigma_{\text{usd}}^2 = \sigma_{\text{msr}}^2 + (\sigma^2 - \sigma_{\text{msr}}^2)\frac{cp - t}{n - cp}$$

where n is the number of brightness levels in the image, and σ^2 is the updated variance from (6).

3. EXPERIMENTAL DESIGN

Instead of arguing that a method is better than the other in several real images, we generate more than two thousand histograms with parameters of the composite distributions carefully controlled. It is fair to say that one is better than the other only after they are tested on a large variety of histograms. It is important to clearly state on which data an experiment is done. In this section, we describe what kind of histograms we generate, how to generate them, and finally the performance measure we use.

3.1. *Histogram types*

We only consider the histograms which are expected to appear with a reasonable frequency in practice. This factor is influenced not only by the types of distributions involved but also by the parameters and the relative weight between the left and the right distributions.

Each histogram consists of two histograms, the one on the left, and the other one on the right. According to the composite distributions, we choose 9 different types of histograms, which are common in practice. In addition, we choose the parameters of histograms in such a way that most of the resulting histograms have clear bimodality and about 5–10% of them are rather vaguely bimodal or unimodal. In other words, most of the histograms do show the bimodality but not to the extent where every technique can come up with the perfect threshold value. We need the histograms which show clear boundaries and also those which do not.

The different values for each of the parameters of the distributions are shown below.

- (Gamma, Normal)

  Gamma: $\alpha(6, 8, 10)$, $\beta(4, 6, 8)$
  Normal: $\mu(140, 170, 200)$, $\sigma(20, 30, 40)$.

- (Gamma, Cauchy)

  Gamma: $\alpha(6, 8, 10)$, $\beta(4, 6, 8)$
  Cauchy: $\alpha(140, 170, 200)$, $\beta(20, 30, 40)$.

- (Gamma, Slash)

  Gamma: $\alpha(6, 8, 10)$, $\beta(4, 6, 8)$
  Slash: $\alpha(140, 170, 200)$, $\beta(10, 15, 20)$.

- (Normal, Normal)

  Normal: $\mu(60, 80, 100)$, $\sigma(20, 30, 40)$
  Normal: $\mu(160, 180, 200)$, $\sigma(20, 30, 40)$.

- (Normal, Cauchy)

  Normal: $\mu(60, 80, 100)$, $\sigma(20, 30, 40)$
  Cauchy: $\alpha(160, 180, 200)$, $\beta(20, 30, 40)$.

- (Normal, Slash)

  Normal: $\mu(60, 80, 100)$, $\sigma(20, 30, 40)$
  Slash: $\alpha(160, 180, 200)$, $\beta(10, 15, 20)$.

- (Cauchy, Cauchy)

  Cauchy: $\alpha(60, 80, 100)$, $\beta(20, 30, 40)$
  Cauchy: $\alpha(160, 180, 200)$, $\beta(20, 30, 40)$.

- (Cauchy, Slash)

  Cauchy: $\alpha(60, 80, 100)$, $\beta(20, 30, 40)$
  Slash: $\alpha(160, 180, 200)$, $\beta(10, 15, 20)$.

- (Slash, Slash)

  Slash: $\alpha(60, 80, 100)$, $\beta(20, 30, 40)$
  Slash: $\alpha(160, 180, 200)$, $\beta(10, 15, 20)$.

(Gamma, Normal) means that we use a Gamma distribution on the left, and Normal distribution on the right.

The final factor to be considered is the relative weight of the left and the right distributions. Even when the means of two distributions are placed far away from each other, the resulting histogram would be almost unimodal if the relative weight were close to 0 or 1 since the small sized distribution is dominated by the big one so that the existence of the former hardly appears. A close examination of the resulting histograms from other possibilities led us to the following numbers, 0.4, 0.5, and 0.7. In total, we have 9 types of distributions with 9 by 9 different parameters and 3 different proportions so that 2187 distinct histograms are generated.

3.2. *Generation of random variates*

The modules generating Normal, Gamma, Cauchy, and Slash variates are described in this section. One thing common to all the modules is that when we get a variate falling beyond the meaningful range, i.e. [0–255], we simply throw it away and generate another one. By so doing, it is guaranteed that all the variates used are in the range [0–255].

3.2.1. *Normal variates.* There are several algorithms known for generating pseudo random Normal variates. What we use here is a modified Box–Mueller algorithm. The algorithm consists of the following steps.

1. Generate two uniform random numbers, U_1 and U_2.
2. Let $V_1 = 2 * U_1 - 1$, and $V_2 = 2 * U_2 - 1$.
3. Let $S = V_1^2 + V_2^2$.
4. If $S \geqslant 1$, Go back to step 1.
5. Else, let $T = \sqrt{-2 * \log(S)/S}$.
6. Let $X = V_1 * T$.

X is a random variate from the normal distribution.

3.2.2. *Gamma variates.* The generation of the Gamma variates is straightforward in that we generate α uniform variates, multiply them, take the logarithm of the result, and multiply by $-\beta$, which results in a gamma variate.[4]

3.2.3. *Cauchy variates.* Cauchy variates are generated in a more trivial way. We induce the inverse cumulative distribution function, generate a uniform variate from the range of [0–1], and calculate the corresponding value, which is the Cauchy variate.

3.2.4. *Slash variates.* Slash distribution is defined as $N(0,1)/U(0,1)$. However, in order to control the position in the histogram and the variance, we used $N(0,\beta)/U(0,1) + \alpha$. Employing the same method as in the Normal variate generation, we get $N(0,1)$ variates. We multiply by β and divide by a uniform variate from [0–1], and finally add α to it.

3.2.5. *Uniform variates.* The basis of all the above variates is the uniform variates out of [0–1]. The 4.3 BSD C language provides a uniform random number generation function which returns a number in the range of 0–2147483647. The latter is the maximum number representable in a 32 bit machine, i.e. $2^{31} - 1$.

3.3. *Performance measure*

We call the performance measure the *error ratio*, which is the percentage of misclassified pixels which are not misclassified in the imaginary perfect method.

The smaller the *error ratio* of a method, the better the method. Note that even the perfect method has misclassification errors.

First we need to know the threshold provided by the perfect method. We can compute it since we know the composite histograms and their respective parameters and the ratio. We first describe the way to get the exact threshold value, and the validity of the error ratio we measure as a performance index.

3.3.1. *Finding the exact threshold.* The exact threshold value is obtained by solving the following equation. This is called *minimum error threshold.* This is only possible when we know the exact distributions in the histogram.

$$qf(x) = (1 - q)g(x) \tag{7}$$

where f and g are the probability density functions of the left and the right distributions and q is the ratio of the pixels from the left distributions and the total pixels.

Suppose that for instance a histogram consisting of Gamma and Cauchy distributions is given. Then we should solve equation (2) for x to find the exact threshold value.

$$q\frac{x^{\alpha-1}e^{-\frac{x}{\beta}}}{\Gamma(\alpha)\beta^{\alpha}} = (1 - q)\frac{1}{\pi\beta_1\left(1 + \left(\frac{(x - \alpha_1)}{\beta_1}\right)^2\right)} \tag{8}$$

where α, β, α_1, β_1, and q are given constants. It is clear that there is no easy closed form solution for the above equation. Instead, all the probability density values within a interval (between modes) are evaluated and the point that has minimum difference in terms of the evaluated function values was chosen as the exact threshold value. This is because the exact threshold value, in general, lies between the modes of two density functions and the modes are expressed in terms of given parameters.

3.3.2. *Performance index: error rate.* Unless the intersection of the domains of two distributions contributing a histogram is empty, i.e. they do not overlap at all, even the correct threshold value misclassifies a certain number of pixels. This was implied when we adopted the minimum error threshold as the correct threshold. From now on, we refer to the imaginary thresholding method which finds the correct threshold value as perfect method.

Let us look at the thresholding problem in terms of a decision problem.[5] Given two sets of pixels, called Left and Right, we have to decide the point which classifies the whole data into two sets. Naturally, there are two types of errors, i.e. classifying a pixel from Left as one from Right, and classifying a pixel from Right as one from Left. Let $P(L|L)$, $P(L|R)$, $P(R|L)$, and $P(R|R)$ denote the probability that a Left pixel is classified as from the Left, that a Right pixel is classified as from the Left, that a Left pixel as from the Right, and that a Right pixel as from the Right respectively. The sum of them is equal to 1.

We want to reduce the classification error, or $P(L|R) + P(R|L)$ as much as possible. Actually it is the correct threshold value that minimizes the error. If we regard the classification error as a function of the threshold value, then the error associated with the correct threshold value is the minimum, i.e. lower bound. Now what we want to measure as the error of a method given a histogram is how far the classification error is from the lower bound. Thus we measure the difference bdtween the two and take it as an error index.

Suppose that c and f denote the correct threshold value and the calculated value respectively. Also let's assume that $F(x)$, $G(x)$, denote the cumulative distribution functions of the left, and the right distributions respectively. The classification error is $P(L|R) + P(R|L)$. Since $P(L|R) = G(f)$ and $P(R|L) = 1 - F(f)$, the misclassification error of the method considered is $1 - F(f) + G(f)$, and the misclassification error of the perfect method is $1 - F(c) + G(c)$, the method error is expressed as $1 + G(f) - F(f) - (1 + G(c) - F(c))$, which is simplified as $(F(c) - F(f)) + (G(f) - G(c))$. What we calculate as an error measure is the number of pixels which lie in the range between the correct threshold value and the calculated threshold value minus the number of the misclassified pixels even by the correct threshold. Therefore the probability that a pixel is counted as falling in the error range is $|(F(c) - F(f)) - (G(c) - G(f))|$, which is exactly the one we derived above.

4. RESULTS AND DISCUSSIONS

We measured the performance of the original and 3 modified Kittler and Illingworth methods, namely *Exact*, *Combine*, and 2*Phase*. The 3 modified methods differ only in the way of getting the cutoff point. *Exact* uses the exact threshold. This is not known in practice. The reason we include this is to see the limit of the modification we make since it will give the best result. In *Combine*, the cutoff point is computed using Otsu's method.[3] 2*Phase* uses Kittler and Illingworth's original method to estimate the cutoff point. Concerning the name, we regard the process of finding the cutoff point as the phase 1 and that of finding the final threshold value as the phase 2. When the original fails to give any reasonable estimate, we simply pick the middle point in the brightness level spectrum, i.e. 128 in our experiment, as the cutoff point.

Tables 1–10 show the error rates. Before we discuss the results, a brief explanation about what the table entries mean is provided. Each table corresponds to each combination of composite distributions in histograms. For instance, Table 1 is about the histograms made up of two Cauchy distributions. Table 10 accounts for all 2187 cases while others account for 243 histograms each. The entries *Mean* and *Std* represent the means and the variance of the error

Table 1. Error rate in percentage — Cauchy and Cauchy

Method	Mean	Std	MIN	25%	Median	75%	95%	MAX
Original	2.683	8.700	0.000	0.055	0.224	0.637	18.785	52.231
Exact	0.923	5.556	0.000	0.011	0.026	0.070	0.576	49.161
Combine	1.297	5.536	0.000	0.061	0.221	0.647	2.466	49.161
2Phase	1.287	5.365	0.000	0.051	0.228	0.543	2.230	49.161

Table 2. Error rate in percentage — Cauchy and Slash

Method	Mean	Std	MIN	25%	Median	75%	95%	MAX
Original	2.603	8.425	0.000	0.055	0.218	0.773	20.710	51.190
Exact	0.808	5.031	0.000	0.011	0.023	0.151	0.825	52.521
Combine	1.214	5.033	0.000	0.050	0.228	0.780	3.199	52.521
2Phase	1.261	4.748	0.000	0.048	0.191	0.650	2.501	48.940

Table 3. Error rate in percentage — Gamma and Cauchy

Method	Mean	Std	MIN	25%	Median	75%	95%	MAX
Original	1.119	5.844	0.000	0.021	0.069	0.171	0.737	45.119
Exact	0.073	0.092	0.000	0.015	0.045	0.092	0.248	0.817
Combine	0.187	0.734	0.000	0.014	0.057	0.156	0.618	10.748
2Phase	1.110	5.856	0.000	0.018	0.063	0.158	0.578	45.163

Table 4. Error rate in percentage — Gamma and Normal

Method	Mean	Std	MIN	25%	Median	75%	95%	MAX
Original	0.610	4.092	0.000	0.069	0.146	0.274	0.632	50.000
Exact	0.120	0.122	0.000	0.035	0.087	0.162	0.307	0.999
Combine	0.187	0.370	0.000	0.041	0.095	0.185	0.667	4.094
2Phase	0.204	0.440	0.000	0.054	0.116	0.216	0.500	5.402

Table 5. Error rate in percentage — Gamma and Slash

Method	Mean	Std	MIN	25%	Median	75%	95%	MAX
Original	1.046	5.829	0.000	0.022	0.071	0.192	0.882	48.058
Exact	0.094	0.165	0.000	0.014	0.044	0.114	0.296	1.495
Combine	0.141	0.320	0.000	0.016	0.053	0.130	0.460	3.092
2Phase	1.021	5.836	0.000	0.019	0.066	0.171	0.729	48.083

Table 6. Error rate in percentage — Normal and Cauchy

Method	Mean	Std	MIN	25%	Median	75%	95%	MAX
Original	3.499	9.481	0.000	0.146	0.479	1.089	24.234	56.144
Exact	1.320	6.316	0.000	0.048	0.080	0.312	2.141	54.421
Combine	1.562	5.710	0.000	0.109	0.345	0.874	5.135	54.898
2Phase	1.695	5.111	0.000	0.174	0.490	0.990	5.031	54.421

rates in percentage. All the others are order statistics. In the entry MAX, we see the worst case performance of each method while in the MIN, the best performance is shown. Consider Table 4 as an example, which is for Gamma and Normal case. *Original* method gives a threshold which misclassifies 50% more pixels than the exact threshold. 2*Phase* outputs the threshold value which does 5% worse than the perfect.

All three different modifications performed better than the original method in almost all entries. There are some cases for which the modified methods worked poorly, i.e. the MAX cases. All the worst cases come from clear unimodal histograms whose images do not comprise objects and background. Thresholding techniques should not be used for such images. Therefore we will not discuss such worst cases any further.

It is the data near the 90–95 percentile that really matters since these are performance indices for the histograms which show unclear bimodality. In every table, we see a significant performance difference between the *Original* and the modified methods by as much as an order of magnitude. Even in other entries, about two thirds of the case show the mean of error rates decreased owing to the modification by more than 50%.

Next we show a specific example where *Original* fails to give a good threshold while 2*Phase* gives the one which is very close to the exact threshold. Figure 2 shows the histogram, which consists of two Cauchy distributions with 70% of the pixels from the left distribution. The vertical lines indicate the threshold values from the five different methods and the exact threshold value. 2*Phase* and *Combine* give the same value of 152 while *Exact* gives 158 which is very close to the exact threshold of 159 so that the two lines

Table 7. Error rate in percentage—Normal and Normal

Method	Mean	Std	MIN	25%	Median	75%	95%	MAX
Original	3.591	10.531	0.000	0.031	0.197	1.197	24.503	57.856
Exact	1.680	8.282	0.000	0.009	0.027	0.113	2.901	56.050
Combine	2.019	7.672	0.000	0.038	0.229	1.005	5.015	56.050
2Phase	1.934	6.978	0.000	0.018	0.181	0.875	6.531	56.050

Table 8. Error rate in percentage—Normal and Slash

Method	Mean	Std	MIN	25%	Median	75%	95%	MAX
Original	2.728	8.627	0.000	0.074	0.238	0.657	20.440	56.102
Exact	1.005	5.510	0.000	0.027	0.060	0.219	1.717	56.189
Combine	1.405	5.687	0.000	0.066	0.274	0.660	4.553	56.189
2Phase	1.262	3.889	0.000	0.098	0.248	0.635	4.861	34.664

Table 9. Error rate in percentage—Slash and Slash

Method	Mean	Std	MIN	25%	Median	75%	95%	MAX
Original	1.969	7.455	0.000	0.039	0.137	0.412	14.309	54.325
Exact	0.670	4.862	0.000	0.011	0.027	0.097	0.467	52.281
Combine	0.937	4.867	0.000	0.045	0.163	0.440	1.606	52.281
2Phase	1.094	4.710	0.000	0.030	0.157	0.370	1.743	49.458

Table 10. Error rate in percentage—Total

Method	Mean	Std	MIN	25%	Median	75%	95%	MAX
Original	2.205	7.970	0.000	0.045	0.156	0.500	16.010	57.856
Exact	0.744	4.959	0.000	0.017	0.046	0.138	0.780	56.189
Combine	0.994	4.807	0.000	0.037	0.150	0.470	2.563	56.189
2Phase	1.207	5.097	0.000	0.040	0.152	0.452	3.353	56.050

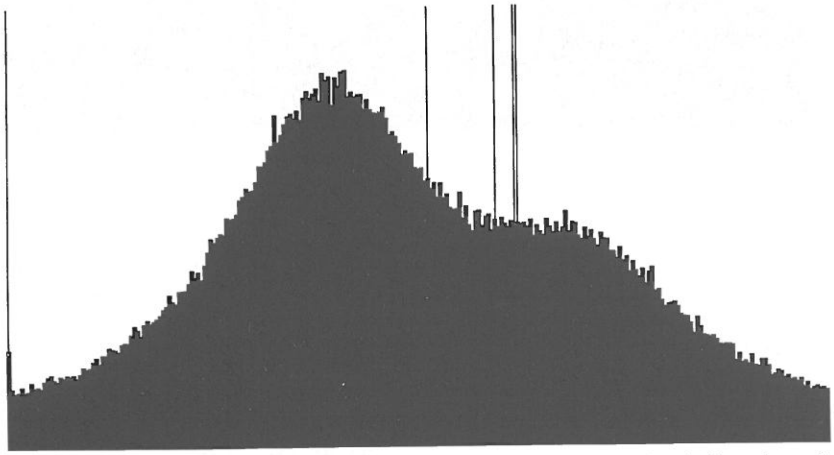

Fig. 2. Threshold values from *Original*, *Otsu*, *2Phase*, *Combine*, the exact threshold, and one from *Exact*. Histogram, Cauchy(100, 40) Cauchy(180, 40) 70%.

look like a thicker one. This is the reason why we have only 4 vertical lines in the figure.* The *Original* apparently fails to give any reasonable threshold. The reason is clear in the next figures. Figures 3 and 4 depict the respective criterion functions computed from the two methods. Note that how the shapes of the criterion functions look similar to the histogram. In Fig. 3 there is no local minimum. However, by providing more precise estimates to the model, we get the criterion function which has the local minimum and gives very good threshold values as shown in Fig. 4.

The best performance by *Exact* indirectly shows the validity of the modification and the importance of the cutoff point. The closer the cutoff point is to the exact threshold, the better performance we got from the modified methods. Since we can not use *Exact* in practice, it would be better to compare *Combine* and *2Phase*. Although there are some cases where *2Phase*'s performance is better than that of *Combine*, it should be stated that the latter provides

* Note the differences between *Exact* and the exact threshold. By *Exact* we mean a method we tested where the exact threshold is used as the cutoff point.

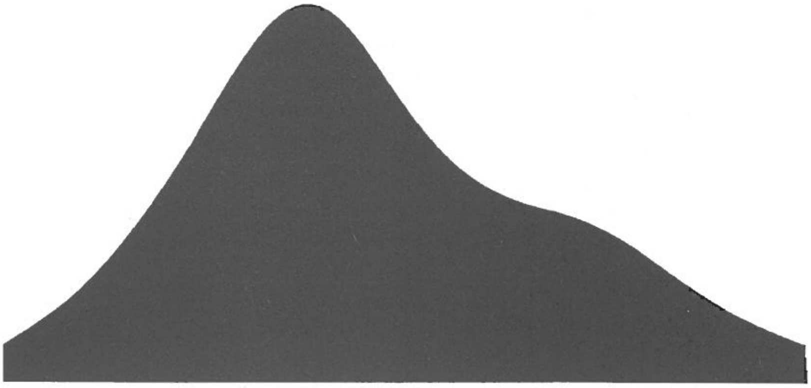

Fig. 3. Criterion function from the *Original*.

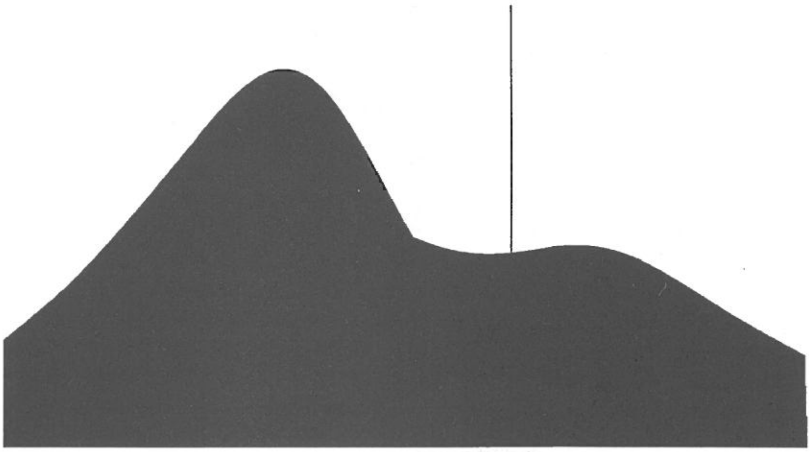

Fig. 4. Criterion function from the 2*Phase*.

better results than the former. The most outstanding difference can be found in Gamma–Slash and Gamma–Cauchy cases. Combine actually uses 2 different techniques although one of them is very simple and computationally cheap. By using two different methods, a bias in a solution provided by one method is offset by the other.

5. CONCLUSION

We made simple modifications to Kittler and Illingworth's thresholding method and compared performances between the original method and the modified ones on the wide variety of histograms we might face in practice. We controlled the parameters of distribution functions so that we could generate basically bimodal histograms. The empirical data show the apparently better performances by the modified methods and consequently validate the modifications we made to the original version. The improvement was outstanding especially for the histograms which show vague bimodality like the one we show in the figures. Our methods become more robust in the sense that they are less sensitive to the modality. In conclusion, the update of the biased variances is legitimate and makes the technique more powerful with only a little more computational effort.

Acknowledgement—Sungzoon Cho was sponsored by E/AE USIA Washington DC under an exchange visitor program under #G-1-0001, a Fulbright grant.

REFERENCES

1. J. Kittler and J. Illingworth, Minimum error thresholding, *Pattern Recognition* **19**, 41–47 (1986).
2. J. Kittler, J. Illingworth and J. Föglein, Threshold selection based on a simple image statistic, *Comput. Vision Graphics Image Process.* **30**, 125–147 (1985).
3. N. Otsu, A threshold selection method from grey level histograms, *IEEE Trans. Systems Man Cybernetics* **SMC-9**, 62–66 (1979).
4. J. Payne, *Introduction to Simulation*. McGraw-Hill, New York (1982).
5. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. Academic Press, New York (1976).

About the Author—SUNGZOON CHO was born in Seoul, Korea on 2 December 1960. He received the B.S. and M.S. degree in industrial engineering from the Seoul National University, Seoul, Korea in 1983 and 1985 respectively. He also received the M.S. degree in computer science from the University of Washington, Seattle in 1987. He is currently a Ph.D. student at the University of Maryland, College Park. His research interest includes pattern recognition. He is a member of IEEE and ACM connectionist modelling and artificial intelligence.

About the Author—ROBERT M. HARALICK was born in Brooklyn, NY, on 30 September 1943. He received the B.A. degree in Mathematics from the University of Kansas, Lawrence in 1964, the B.S. degree in electrical engineering in 1966, the M.S. degree in electrical engineering in 1967, and the Ph.D. degrees from the University of Kansas in 1969.

He has worked with Autonetics and IBM. In 1965 he worked for the Center for Research, University of Kansas, as a Research Engineer and in 1969 he joined the faculty of the Department of Electrical Engineering, where he served as a Professor from 1975 to 1978. In 1979 he joined the faculty of the Departments of Electrical Engineering and Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, where he was a Professor and Director of the Spatial Data Analysis Laboratory. From 1984 to 1986 he served as Vice President of Research at Machine Vision International, Ann Arbor, MI. He now holds the Boeing Clairmont Egtvedt chaired professorship in the Department of Electrical Engineering, University of Washington, Seattle. He has done research in pattern recognition, multi-image processing, remote sensing, texture analysis, data compression, clustering, artificial intelligence, and general systems theory, and has published over 180 papers. He is responsible for the development of GIPSY (General Image Processing System), a multi-image processing package which runs on a minicomputer system.

Dr Haralick is a member of the Institute of Electrical and Electronics Engineers, the Association for Computing Machinery, and the Pattern Recognition Society.

About the Author—SEUNGKU YI was born in Seoul, Korea on 1 September 1955. He received the B.S. degree in computer science and statistics from the Seoul National University, Seoul, Korea in 1979 and the M.S. degree in computer science from both Korea Advanced Institute of Science and Technology, Seoul, Korea, and the University of Washington, Seattle in 1981 and 1986 respectively.

From 1981 to 1984 he worked for Daewoo Shipbuilding and Heavy Machinery Ltd, Seoul, Korea as a systems engineer. He joined the University of Washington in 1984 and currently is a Ph.D. candidate in computer science. His research interest includes artificial intelligence, image understanding, and robot vision.

Mr Yi is a member of the Institute of Electrical and Electronics Engineers.