Chapter 12
Image Thresholding

Abstract. The subject of this chapter is image thresholding in which we transform an input image, A, into a binary image B, where the pixel gray-levels in B are restricted to $\{0,1\}$. If a_m is the gray level of the mth pixel in A, then the corresponding value in B is

$$b_m = \begin{cases} 1 \text{ if } a_m \geq t_m \ , \\ 0 \text{ otherwise} \ , \end{cases}$$

where t_m is the threshold value for the mth pixel. The thresholds $t_m, m \in \{1,2,\ldots,M\}$, may all be equal to a global threshold t_G or they may vary locally (i. e. from pixel to pixel). In this chapter we shall concentrate on unsupervised thresholding methods. These are thresholding algorithms in which we only use information contained in the current input image to calculate t_m and t_G.

12.1 Global Thresholding

Let A denote a given input image, where a_m denotes the gray-level of the mth pixel. Then the pixels in the image are divided into two groups: "low" intensity pixels whose gray-levels are less than, or equal to, a threshold t and "high" intensity pixels whose gray-levels are greater than t. We follow [12] and categorize the thresholding methods into five groups as follows:

Histogram-based Methods. These methods analyze the shape and features of the image histogram.

Cluster-based Methods. These methods cluster the pixel gray-levels into two parts as background and foreground pixels.

Entropy-based Methods. The methods use the entropy of the histogram or cross-entropy between input image and the thresholded image to find an optimal threshold.

Object Attribute-based methods. The methods use a measure of similarity between the gray-level and the binarized images.

Spatial-based Methods. The methods use a higher-order probability distribution to model the correlation between pixels.

In some applications we find it useful to modify the input image before thresholding. The following example describes one such application involving the estimation of raindrop sizes.

Example 12.1. Histogram Modification [13]. Rainfall estimates often require an estimate of the distribution of raindrop sizes. Unfortunately image irregularities mean the size estimates are often noisy and inaccurate. In [13] the image irregularities are reduced by applying the following iterative histogram modification algorithm to the raindrop images.

The algorithm iteratively sharpens the peaks of the input image histogram by considering the number of pixels H_i having a particular gray-level i and comparing it with $\bar{H}_i$:

$$\bar{H}_i = \frac{1}{2R}\left(\sum_{k=i-R}^{i-1} H_i + \sum_{k=i+1}^{i+R} H_i\right) ,$$

where $\bar{H}_i$ is the average number of pixels in the neighboring R bins on either side of i. Whenever $H_i > \bar{H}_i$, we shift $\lfloor xH_l \rfloor$ pixels from the lth bin to the neighboring bin nearest to the ith bin, where

$$x = \frac{H_i - \bar{H}_i}{H_i} .$$

12.2 Statistical Algorithms

Many of the global thresholding algorithms are statistical in nature. In this case, the threshold t is found by analyzing the image histogram $\mathbf{H} = (H_0, H_1, \ldots, H_{K-1})^T$, or equivalently, the probability distribution $\mathbf{p} = (p_0, p_2, \ldots, p_{K-1})^T$, where H_k is the number of pixels in the input image whose gray levels are equal to k [1] and $p_k = H_k / \sum_{l=0}^{K-1} H_l$.

Fig. 12.1 shows binary images $B_k, k \in \{1, 2, \ldots, K\}$, obtained by thresholding an input image I using K different global thresholding algorithms.

[1] For an 8-bit deep input image the gray-levels are $0, 1, \ldots, 255$, i. e. $K = 256$.

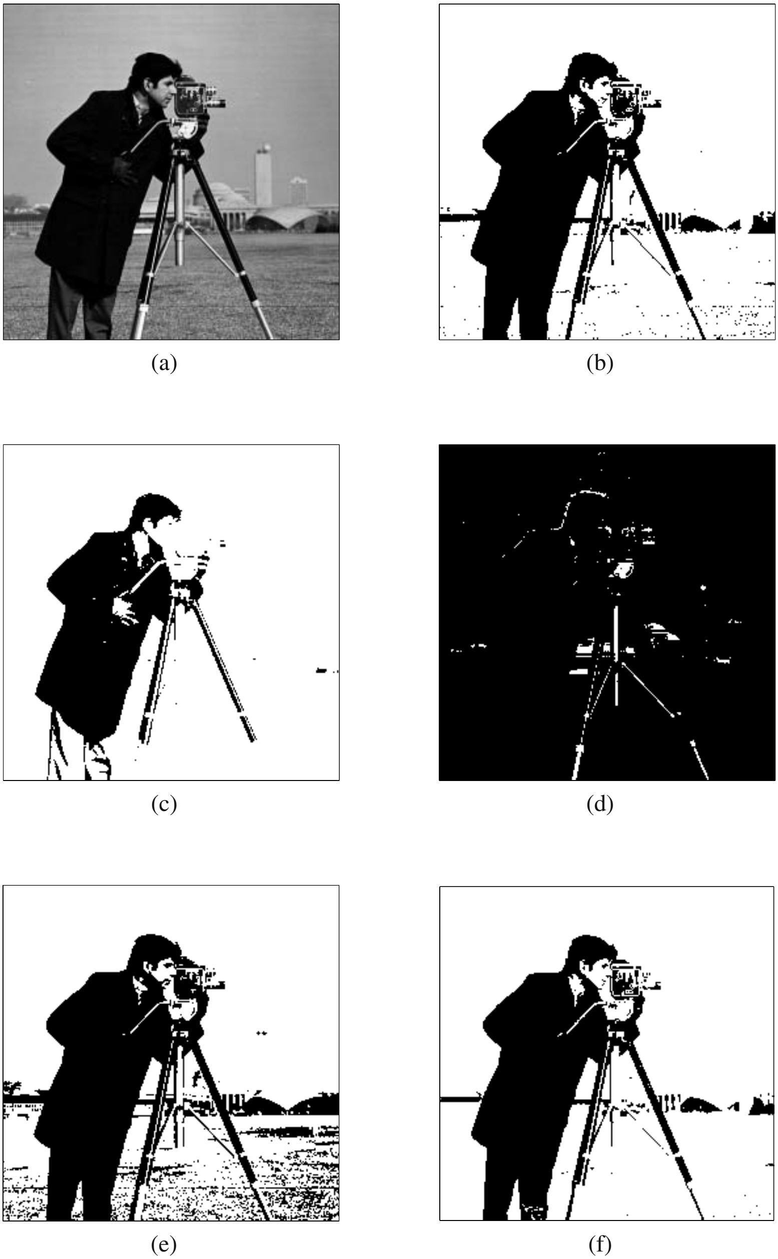

Fig. 12.1 **(a)** Shows an input image I. **(b)**-**(e)** Shows the binary image obtained by thresholding I using the Otsu, Kittler-Illingsworth, Kapur and Tsai thresholding algorithms. **(f)** Shows the binary image obtained by thresholding I using a maximum likelihood algorithm in which we assume a mixture of two Gaussian distributions [1].

12.2.1 Ridler-Calvard

Ridler and Calvard [10] describe an iterative heuristic thresholding technique. The initial threshold, t, is set equal to the mean pixel gray-level $\bar{\mu}$:

$$\bar{\mu} = \sum_{k=0}^{K} kH_k / \sum_{k=0}^{K} H_k \ .$$

Thereafter, the threshold value t_k for the kth iteration is given by:

$$t_k = \frac{\mu_0(t_k) + \mu_1(t_k)}{2} \ ,$$

where $\mu_0(t_k)$ and $\mu_1(t_k)$ are, respectively, the mean gray-levels of the pixels which lie below and above the threshold t_k.

12.2.2 Otsu

According to Otsu [9] we select a threshold t which minimizes the within-group variance $\sigma_W^2(t)$ of the pixels. Mathematically, $\sigma_W^2(t)$ is defined as follows:

$$\sigma_W^2(t) = P_0(t)\sigma_0^2(t) + P_1(t)\sigma_1^2(t) \ ,$$

where $P_0(t)$ and $P_1(t)$ are, respectively, the proportion of pixel gray-levels which lie below, and above, the threshold t and $\sigma_0^2(t)$ and $\sigma_1^2(t)$ are, respectively, the variance of the pixel gray-levels which lie below, and above, the threshold t. The optimum threshold may be found by an exhaustive search. In Ex. 6.6 we illustrate the action of the Otsu algorithm to segment an input image into areas of background and areas of foreground.

For an efficient method of implementating the Otsu algorithm see [4].

12.2.3 Kittler-Illingworth

According to Kittler and Illingworth [7] we select a threshold t which minimizes a criterion $J(t)$, where

$$J(t) = 1 + 2(P_0(t)\ln\sigma_0(t) + P_1\ln\sigma_1(t)) - 2(P_0(t)\ln P_0(t) + P_1(t)\ln P_1(t)) \ .$$

where $P_0(t)$ and $P_1(t)$ are, respectively, the proportion of pixel gray-levels which lie below and above the threshold t. If the below threshold and the above threshold pixels each follow a Gaussian distribution, then the criterion $J(t)$ represents the average pixel classification error rate. As the threshold t is varied, the parameters $P_0(t), P_1(t), \sigma_0^2(t)$ and $\sigma_1^2(t)$ change. The problem of minimum error threshold selection is reduced to computing $J(t)$ for all t, and finding its minimum value.

Note. Even if the assumption of a bimodal normal distribution is valid , the model parameters are biased estimates of the true values, as the tails of the overlapping distributions are truncated. Thus, the correctness of the estimated threshold relies on this overlap being small. However, the most challenging images are those where the histograms do not have two well separated modes [3].

12.2.4 Kapur

According to Kapur [6] we select a threshold t which maximizes the sum of the entropies of pixel gray-levels which lie below the threshold t and which lie above the threshold t. Mathematically, the sum of the entropies is

$$\psi(t) = \ln(P_0(t)P_1(t)) + \frac{H_t}{P_0(t)} + \frac{H_G - H_t}{P_1(t)} ,$$

where

$$H_t = -\sum_{k=0}^{t} p_k \ln p_k ,$$
$$H_G = -\sum_{k=0}^{K-1} p_k \ln p_k .$$

The discrete value of t which maximizes $\psi(t)$ is the threshold value which maximizes the information between object and background.

12.2.5 Tsai

According to Tsai [14] we select a threshold t such that the binary image has the same first three moments as the input image, where the ith moment of the image I is

$$m_i = \sum_{k=0}^{K-1} k^i p_k ,$$

and, by definition, $m_0 = 1$.

Example 12.2. Change Detection in Remotely Sensed Imagery [8]. The most common methodology to carry out an unsupervised change detection in remotely sensed imagery is to compare two spatially aligned multitemporal remote sensing images $I_1(m,n)$ and $I_2(m,n)$ taken at two different dates over the same geographical area. The result of the comparison is a binary image $B(m,n)$ where

$$B(m,n) = \begin{cases} 1 \text{ if the pixel } (m,n) \text{ is classified as ``changed''} , \\ 0 \text{ otherwise} . \end{cases} \quad (12.1)$$

One way of generating B is to threshold the difference image $D = |I_1 - I_2|$ using one of the above global thresholding algorithms. A more powerful method is to match the local gray-level distributions in the two images: For each pixel (m,n) we establish a window W of size $(2L+1) \times (2L+1)$ centered on (m,n). Let H_1 and H_2 denote, respectively, the local distributions, or histograms, of the pixel gray-levels $I_1(i,j)$ and $I_2(i,j), (i,j) \in W$. We then compare H_1 and H_2 using an appropriate similarity measure (see Chapt. 14). We often use the mutual information $MI(H_1,H_2)$ or the Kullback-Leibler distance $KL(H_1,H_2)$ for this purpose.

For a method of combining multiple thresholded images using an *unsupervised* weighted majority vote rule see Ex. 10.7.

12.3 Local Thresholding

In general local thresholding algorithms work by moving a sliding window over the input image. In each window we calculate a threshold t using one of the above global thresholding algorithms. Then t is applied to the center pixel of the window.

However this approach suffers from the drawback that there is no principled method for choosing the local window size. A recent alternative method is the following "ensemble-learning" algorithm (see Sect. 10.7) which may be used to convert the global thresholding algorithms into local algorithms.

Given an input image I we randomly select K image patches from I. The size and location of the image patches is random and the patches may overlap. Thus a given pixel (m,n) may be present in L image patches, where $0 \leq L \leq K$. We threshold each image patch using a global thresholding algorithm. Then for each pixel (m,n) we calculate the number of times it has a label of one. If the number of times is greater than, or equal to, $L/2$, then we set $B(m,n)$ equal to one, otherwise $B(x,y) = 0$.

12.4 Software

HISTHRESH. HistThresh is a matlab toolbox for global image thresholding. Author is Antti Niemisto.

12.5 Further Reading

Recently the expectation-maximization (EM) algorithm has been used for image thresholding assuming the foreground and background pixels follow a given distribution. In [1] the foreground and background pixels are assumed to follow a generalized Gaussian distribution. A comprehensive survey of image thresholding methods is [12]. A survey of entropy and relative entropy thresholding methods is [2]. Two earlier reviews are: [5] and [11].

References

1. Bazi, Y., Bruzzone, L., Melgani, F.: Image thresholding based on the EM algorithm and the generalized Gaussian distribution. Patt. Recogn. 40, 619–634 (2007)
2. Chang, C.-I., Du, Y., Wang, J., Guo, S.-M., Thouin, P.D.: Survey and comparative analysis of entropy and relative entropy thresholding techniques. IEE Proc. Vis. Image Sig. Process. 153, 837–850 (2006)
3. Cho, S., Haralick, R., Yi, S.: Improvement of Kittler and Illingworth's minimum error thresholding. Patt. Recogn. 22, 609–618 (1989)
4. Dong, L., Yu, G., Ogunbona, P., Li, W.: An efficient iterative algorithm for image thresholding. Patt. Recogn. Lett. 29, 1311–1316 (2008)
5. Glasbery, C.A.: An analysis of histogram-based thresholding algorithms. Comp. Vis. Graphics Image Process. 55, 532–537 (1993)
6. Kapur, J.N., Sahoo, P.K., Wong, A.K.C.: A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram. Comp. Vis. Graphics Image Process. 29, 273–285 (1985)
7. Kittler, J., Illingworth, J.: Minimum error thresholding. Patt. Recogn. 19, 41–47 (1986)
8. Melgani, F., Bazi, Y.: Markovian fusion approach to robust unsupervised change detection in remotely sensed imagery. IEEE Geosci. Remote Sensing Lett. 3, 457–461 (2006)
9. Otsu, N.: A threshold selection method from gray-level histogram. IEEE Trans. Syst. Man Cyber. 9, 62–66 (1979)
10. Ridler, T.W., Calvard, S.: Picture thresholding using an interative selection method. IEEE Trans. Syst. Man Cyber. 8, 630–632 (1978)
11. Sahoo, P.H., Soltani, S., Wong, A.K.C.: A survey of thresholding techniques. Comp. Vis. Graphics Image Process. 41, 233–260 (1988)
12. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. J. Elect. Imag. 13, 146–165 (2004)
13. Sivasubramanian, N.A., Saylor, J.R.: Application of a histogram modification algorithm to the processing of raindrop images. Opt. Engng. 47, 037011-1–037011-10 (2008)
14. Tasi, W.: Moment-preserving thresholding: a new approach. Comp. Vis. Graphics Image Process. 29, 377–393 (1985)