An Optimization-Based Approach to Image Binarization

Liju Dong[1,2] and Ge Yu[1]
[1]School of Information Science and Engineering, Northeastern University
Shenyang 110004, China
[2]School of Information Engineering, Shengyang University
Shenyang 110044, China
E-mail: dong_liju@yahoo.com.cn, yuge@mail.neu.edu.cn

Abstract

Image binarization is one of the main techniques for image segmentation. It segments an image into foreground and background. The foreground contains interested objects. Usually, the binarization is carried out with a threshold found from the histogram of an image automatically. It has many applications in pattern recognition, computer vision, and image and video understanding. This paper formulates the binarization as an optimization problem: finding the best threshold that minimizes a weighted sum-of-squared-error function. A fast iterative optimization algorithm is given to reach this goal. Our algorithm is also compared with a classic commonly-used binarization method. The experiments show that the two algorithms yield the same segmentation results but our algorithm is more efficient.

1. Introduction

Image segmentation plays a very important role in many tasks of pattern recognition, computer vision, and image and video retrieval. Many approaches have been proposed in the literature [1–5]. Image binarization is one of the most important techniques for image segmentation. Its goal is to automatically find a threshold from the histogram of the image under study. The threshold divides the image into two regions each with similar gray levels. Among many binarization techniques, the Otsu's method [6] is considered as the most commonly-used one in the survey papers in [1–5]. It is also ranked as the best and fastest global binarization technique in [2], [3] and [7].

In applications such as real-time recognition, video surveillance, and tracking systems, it is desirable to develop as fast algorithms as possible while keeping necessary processing quality for a task. This paper proposes an efficient approach to image binarization. We formulate the binarization as a discrete optimization problem: finding the best threshold that minimizes a weighted sum-of-squared-error objective function. A fast iterative optimization algorithm is proposed to reach this goal based on the histogram. We also compare our method with the Otsu's. Both theoretic analysis and experiments show that the two methods yield the same segmentation results but our algorithm is much faster.

The rest of this paper is organized as follows. The formulation and algorithm of the new method are given in Section 2. The comparison between the two methods is presented in Section 3. The experimental results are shown in Section 4. Finally, Section 5 concludes this paper.

2. The proposed approach

In this section, we first formulate image binarization as a discrete optimization problem based on the histogram of the image, and then give the new iterative algorithm for finding the optimal threshold.

2.1. Formulation of the problem

The goal of image binarization is to divide the pixels of an image into two regions with similar gray levels. This is similar to data clustering where data are partitioned into clusters with similar properties. Therefore, the widely used sum-of-squared-error criterion in data clustering [8, 9] is modified in this paper to be the objective function for our application.

Suppose that there are L gray levels $\{0,1,...,L-1\}$ in an image. Let n_l denote the number of pixels at level l. If an image contains two different objects each

Proceedings of the Fourth International Conference on Computer and Information Technology (CIT'04)
0-7695-2216-5/04 $20.00 © 2004 **IEEE**

with exactly the same gray level (an ideal case), there will be only two non-zero n_l, $l \in \{0,1,...,L-1\}$, in the histogram of this image (see Fig. 1(a)). However, the practical histogram of a real image with two objects always has much more non-zero n_l on it. These gray levels spread on the histogram in a wide range, as shown in Fig. 1(b). Thus we formulate the image segmentation as finding the two clusters on the histogram such that the total deviation of the gray levels from their corresponding cluster centers (centroids) is minimized (see Fig. 1(c)). More formally, we give the following formulation.

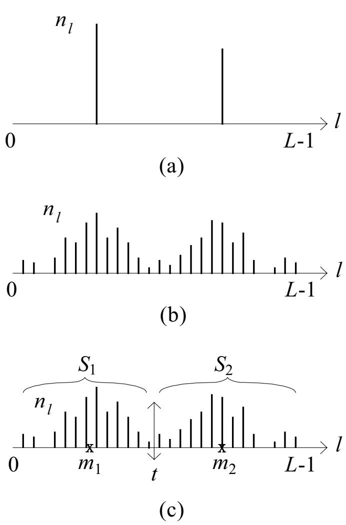


Fig. 1. (a) An ideal histogram with only two non-zero gray levels. (b) A practical histogram with two clusters. (c) One partition of the two clusters of the gray levels, S_1 and S_2, where the centroids of the clusters are m_1 and m_2, and the threshold is t.

Definition 1. *Suppose that the histogram of an image is divided into two clusters (disjoint subsets)* S_1 *and* S_2*, as shown in Fig. 1(c). Let* m_1 *and* m_2 *be the centroids of the clusters. The image binarization problem is to search for the partition* S_1 *and* S_2 *such that the objective function*

$$f(m_1, m_2) = \sum_{i=1}^{2} \sum_{l \in S_i} n_l (l - m_i)^2 \tag{1}$$

is minimized. The threshold t *can be obtained from the final partition* S_1 *and* S_2.

We call $f(m_1, m_2)$ a weighted sum-of-squared-error function, where n_l serves as a weighting factor. The centroids are calculated by

$$m_i = \frac{1}{d_i} \sum_{l \in S_i} n_l l, \; i = 1,2, \tag{2}$$

where

$$d_i = \sum_{l \in S_i} n_l, \; i = 1,2. \tag{3}$$

For convenience, let

$$f_i = \sum_{l \in S_i} n_l (l - m_i)^2, \tag{4}$$

which is the weighted sum of the squared errors in cluster S_i.

Now we derive some equations that are useful for developing an algorithm to find the partition of clusters. The algorithm uses iterative improvement to minimize $f(m_1, m_2)$.

Assume that a gray level k currently in cluster S_i is tentatively moved to cluster S_j. Then m_j changes to m_j^*, where

$$\begin{aligned} m_j^* &= \frac{1}{d_j + n_k}\left(n_k k + \sum_{l \in S_j} n_l l\right) = \frac{n_k k + m_j d_j}{d_j + n_k} \\ &= m_j + \frac{(k - m_j) n_k}{d_j + n_k}, \end{aligned} \tag{5}$$

and f_j increases to

$$\begin{aligned} f_j^* &= n_k (k - m_j^*)^2 + \sum_{l \in S_j} n_l (l - m_j^*)^2 \\ &= n_k \left(k - m_j - \frac{(k - m_j) n_k}{d_j + n_k}\right)^2 + \\ &\quad \sum_{l \in S_j} n_l \left(l - m_j - \frac{(k - m_j) n_k}{d_j + n_k}\right)^2 \\ &= \frac{n_k (k - m_j)^2 d_j^2}{(d_j + n_k)^2} + \sum_{l \in S_j} n_l (l - m_j)^2 + \\ &\quad \sum_{l \in S_j} n_l \frac{(k - m_j)^2 n_k^2}{(d_j + n_k)^2} - 2 \sum_{l \in S_j} n_l (l - m_j) \frac{(k - m_j) n_k}{d_j + n_k} \\ &= \frac{n_k (k - m_j)^2 d_j^2}{(d_j + n_k)^2} + f_j + \frac{d_j (k - m_j)^2 n_k^2}{(d_j + n_k)^2} - \\ &\quad \frac{2(k - m_j) n_k}{d_j + n_k}\left(\sum_{l \in S_j} n_l l - m_j \sum_{l \in S_j} n_l\right). \end{aligned} \tag{6}$$

By (2) and (3), the last term in (6) vanishes. Thus f_j^* has a compact expression

$$f_j^* = f_j + \frac{d_j n_k (k - m_j)^2}{d_j + n_k}. \qquad (7)$$

Under the assumption that S_i has other gray levels in addition to gray level k, implying that $d_i \neq n_k$, we can show in a similar way that m_i changes to

$$m_i^* = m_i - \frac{(k - m_i) n_k}{d_i - n_k}, \qquad (8)$$

and f_i decreases to

$$f_i^* = f_i - \frac{d_i n_k (k - m_i)^2}{d_i - n_k}. \qquad (9)$$

From (7) and (9), we see that the transfer of gray level k from cluster S_i to S_j can reduce $f(m_1, m_2)$ if

$$\frac{d_i n_k (k - m_i)^2}{d_i - n_k} > \frac{d_j n_k (k - m_j)^2}{d_j + n_k}. \qquad (10)$$

The above equations and analysis lead to the algorithm presented in the next section.

2.2. The binarization algorithm

Here the algorithm is listed first. Some discussion and analysis are given later.

1. Select an initial partition of two clusters S_1 and S_2 of the L gray levels on the histogram and calculate m_1, m_2, d_1 and d_2 using (2) and (3)
2. *changed* $\leftarrow$ No
3. **for** $k = 0, 1, ..., L-1$ **do**
4. **begin**
5.   **if** $d_i \neq n_k$ (suppose $k \in S_i$ currently) **then**
6.   **begin**
7. $r_i \leftarrow \frac{d_i n_k (k - m_i)^2}{d_i - n_k}$, $r_j \leftarrow \frac{d_j n_k (k - m_j)^2}{d_j + n_k}$
8.   **if** $r_j < r_i$ **then** (move k to S_j)
9.   **begin**
10.   Update m_j and m_i with (5) and (8)
11. $d_i \leftarrow d_i - n_k$; $d_j \leftarrow d_j + n_k$
12.   *changed* $\leftarrow$ Yes
13.   **end**
14.   **end**
15. **end**
16. **if** *changed* = Yes **goto** Step 2
17. **else** Find threshold t from the final S_1 and S_2
18. Return t and stop

This algorithm reflects the idea of iteratively improvement in minimizing the objective function $f(m_1, m_2)$ as described in Section 2.1. The optimization procedure repeats until no further improvement is obtained. A good initial partition can reduce the number of iterations. Let the smallest and largest non-zero gray levels on the histogram be $l_{\min}$ and $l_{\max}$, respectively. A good initial partition can be obtained by equally dividing $[l_{\min}, l_{\max}]$ into two clusters.

It is not difficult to find the computational complexity of the algorithm. Step 1 or Step 17 can be computed in $O(L)$ time. The computation from Step 3 to Step 15 also requires $O(L)$ time. Therefore, the algorithm runs in $O(LQ)$ time with Q being the number of iterations, which is the number of times Step 2 is visited. From our experiments, we find that $Q \leq 10$ in general.

3. Comparison with the Otsu's method

As mentioned in Section 1, the Otsu's method [6] is considered as the best and fastest global binarization technique in the surveys [2] and [3]. In this section, we will show that the proposed method is equivalent to the Otsu's method but our algorithm is more efficient. We briefly describe the method at first.

Otsu proposed his binarization method from a statistical point of view. Suppose that there are N pixels and L gray levels $\{0, 1, ..., L-1\}$ in an image. Let n_l denote the number of pixels at level l. Then $N = \sum_{l=0}^{L-1} n_l$. The histogram of an image can be normalized as a probability distribution by

$$p_l = \frac{n_l}{N}, \quad \sum_{l=0}^{L-1} p_l = 1. \qquad (11)$$

Assume that a threshold t divides the gray levels into two clusters:

$$S_1 = \{0, 1, ..., t\} \text{ and } S_2 = \{t+1, t+2, ..., L-1\}.$$

Let $\sigma_w^2(t)$, $\sigma_B^2(t)$, and σ_T^2 be the within-class variance, between-class variance, and the total variance of the gray levels. Then the optimal threshold t is determined by maximizing one of the three criteria:

$$\lambda=\frac{\sigma_B^2(t)}{\sigma_w^2(t)},\quad \kappa=\frac{\sigma_T^2}{\sigma_w^2(t)},\quad \eta=\frac{\sigma_B^2(t)}{\sigma_T^2}. \tag{12}$$

where

$$\sigma_T^2=\sum_{l=0}^{L-1}(l-\mu_T)^2 p_l\ ,\ \mu_T=\sum_{l=0}^{L-1} l p_l\ ,$$

$$\sigma_B^2(t)=\sum_{i=1}^{2}\omega_i(\mu_i-\mu_T)^2\ ,\ \omega_1=\sum_{l=0}^{t}p_l\ ,\ \omega_2=\sum_{l=t+1}^{L-1}p_l\ ,$$

$$\mu_1=\sum_{l=0}^{t}\frac{lp_l}{\omega_1},\ \mu_2=\sum_{l=t+1}^{L-1}\frac{lp_l}{\omega_2},\ \sigma_w^2(t)=\sum_{i=1}^{2}\omega_i\sigma_i^2\ ,$$

$$\sigma_1^2=\sum_{l=0}^{t}\frac{(l-\mu_1)^2 p_l}{\omega_1},\ \sigma_2^2=\sum_{l=t+1}^{L-1}\frac{(l-\mu_2)^2 p_l}{\omega_2}.$$

The three criteria in (12) are equivalent, and σ_T^2 is not a function of t. Therefore, since η is the simplest to compute, the optimal threshold t^* is obtained by

$$t^*=\arg\max_{0\le t<L-1}\sigma_B^2(t)\ . \tag{13}$$

Otsu used an exhaustive way to search for the optimal threshold [6]. The computational complexity of his algorithm is $O(L^2)$. Obviously, our algorithm with complexity $O(LQ)$ presented in Section 2.2 is more efficient because $L=255$ and $Q\le 10$ in general. Next, we proof that the two methods are equivalent in essence.

Theorem 1. *If $\sigma_B^2(t)$ in (13) is maximized by a partition S_1 and S_2, the objective function $f(m_1,m_2)$ in (1) is minimized by the same partition, and vice versa.*

Proof. It is not difficult to find the relation $\sigma_T^2=\sigma_W^2(t)+\sigma_B^2(t)$ from the above equations. Thus the partition that maximizes $\sigma_B^2(t)$ will minimize $\sigma_W^2(t)$. Now what we need to verify is to show that the same partition minimizes $\sigma_W^2(t)$ and $f(m_1,m_2)$ simultaneously. By rewriting $\sigma_W^2(t)$, we have

$$\sigma_w^2(t)=\omega_1\sigma_1^2+\omega_2\sigma_2^2=\sum_{l=0}^{t}(l-\mu_1)^2 p_l+\sum_{l=t+1}^{L-1}(l-\mu_2)^2 p_l$$

$$=\frac{1}{N}\sum_{i=1}^{2}\sum_{l\in S_i}(l-\mu_i)^2 n_l\ . \tag{14}$$

Since

$$\mu_1=\sum_{l=0}^{t}\frac{lp_l}{\omega_1}=\frac{1}{\omega_1 N}\sum_{l\in S_1}n_l l\ ,\ \mu_2=\sum_{l=t+1}^{L-1}\frac{lp_l}{\omega_2}=\frac{1}{\omega_2 N}\sum_{l\in S_2}n_l l \tag{15}$$

and

$$\omega_1 N=\sum_{l\in S_1}n_l\ ,\ \omega_2 N=\sum_{l\in S_2}n_l\ , \tag{16}$$

we have $\mu_i=m_i$, $i=1,2$, by comparing (15) and (16) with (2) and (3). Therefore, from (1) and (14), we obtain the relation

$$N\sigma_W^2(t)=f(m_1,m_2)\ , \tag{17}$$

which means that for each partition S_1 and S_2, the values of $N\sigma_W^2(t)$ and $f(m_1,m_2)$ are the same. Thus, the partition that minimizes one also minimizes the other. □

Although Otsu proposed his binarization method from a statistical point of view, while we develop our method from the viewpoint of data clustering, Theorem 1 reveals that they are equivalent. However, the above computational complexity analysis of the two algorithms indicates that our algorithm is more efficient.

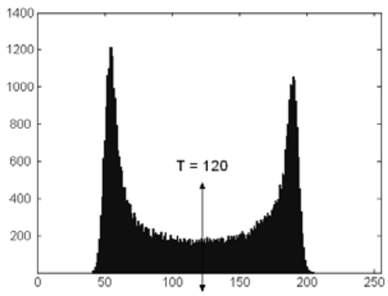
It is worth noting that both the new and the Otsu's methods can be extended to the segmentation with more than one threshold on the histogram. The extension is straightforward, and the detail can be found in [10]. Suppose $c-1$ thresholds segment the histogram into c clusters. The complexities of the new algorithm and the Otsu's algorithm are $O(cLQ)$ and $O(cL^c)$, respectively. It is clear that the new algorithm is more efficient.

4. Experimental results

The proposed algorithm is implemented in Visual C++ and runs on a 1 GHz Pentium III PC. For comparison, we also implement the Otsu's method [6]. More than 80 images have been used to test the two algorithms. Most of the images used in the experiments are chosen from web sites on the internet, such as the one in [11] where a number of public image and video databases are available. In all the experiments, the two algorithms obtain the same result for each image.

Fig. 2 shows an example of the segmentation by the new or the Otsu's algorithm. Fig. 2(a) is a fingerprint image, and Fig. 2(b) gives the satisfactory binarization result. The threshold obtained is 120 and $Q=3$. Fig. 3(a) is an infrared image. The segmentation result by the two algorithms is very good. For this example, the number of iterations $Q=3$ also in the new algorithm.

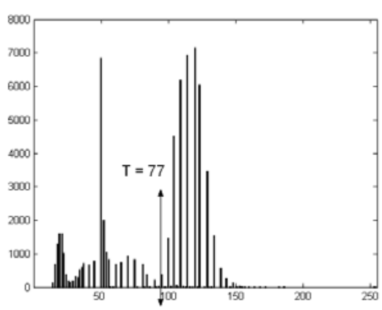
(a) (b)



(c)

Fig. 2. (a) A fingerprint image. (b) Binarization result by the two algorithms. (c) Histogram of the image in (a). The found threshold is 120.

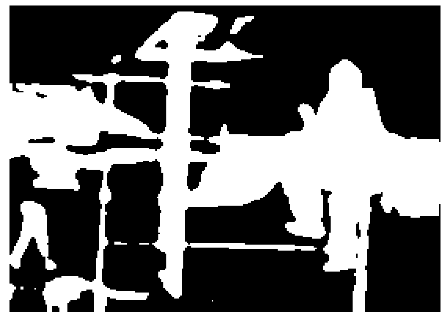
(a) (b)



(c)

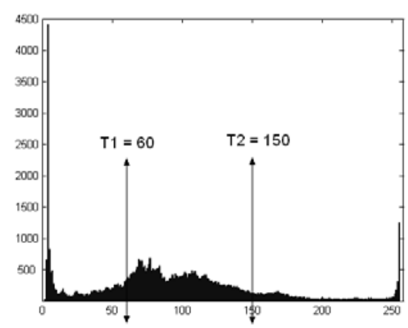
Fig. 3. (a) An infrared image. (b) Binarization result by the two algorithms. (c) The histogram of the image in (a). The found threshold is 77.

(a)



(b)

(c)



(d)

Fig. 4. (a) A thermal infrared image. (b) Bad two-cluster segmentation result. (c) Good three-cluster segmentation result where the clusters are denoted by three gray levels. (d) The histogram of the image in (a) with two thresholds 60 and 150.

In many segmentation tasks, one threshold may not give satisfactory results. For example, the thermal infrared image (Fig. 4(a)) from a surveillance video system cannot be handled with one threshold, as illustrated in Fig. 4(b) where the threshold found is 65. We see that the wanted object (the person) cannot be separated from the other objects. In this case, we can try segmentation with two thresholds found from the histogram. Both our algorithm and the Otsu's algorithm obtained two thresholds 60 and 150. The good segmentation result is given in Fig. 4(c) where the three clusters are represented with three different grey levels.

Now we compare the computational time taken by the two algorithms. As mentioned in the last section, the new algorithm with complexity $O(cLQ)$ is much more efficient than the Otsu's algorithm with complexity $O(cL^c)$, where c is the number of clusters and Q is the number of iterations. In the experiments of two-cluster segmentation, the new algorithm needs only 0.0001 second to handle one image, and is about one order of magnitude faster than the Otsu's algorithm. In the segmentation of three clusters, the new algorithm takes about 0.0008 second, while the Otsu's algorithm have to spend 0.17 second. The new algorithm is more than 200 times faster in this case. Therefore, our algorithm is more suitable for real-time video surveillance and tracking systems.

5. Conclusions

Image segmentation by binarization (thresholding) is the classic technique that is still used widely in many applications of pattern recognition and computer vision. The main advantage is in its simplicity and good efficiency, which is a crucial requirement in most real-time systems.

We have presented a new efficient optimization-based approach to image binarization. The algorithm iteratively minimizes a weighted sum-of-squared-error objective function, which is expected to finally generate good segmentation of gray levels on the histogram. Our approach is proved equivalent to the Otsu's method, which is popular and ranked as the best and fastest global thresholding technique in the survey papers [2] and [3]. However, our re-formulation of the problem allows us to develop an even more efficient algorithm.

A number of experiments have been carried out to test our algorithm and the Otsu's algorithm. While the two algorithms yield the same segmentation results, our algorithm is more than 10 times and 200 times faster for two-cluster and three-cluster segmentation, respectively. Therefore, our algorithm is more efficient and has more applications, especially in real-time video surveillance and tracking systems.

Acknowledgements

This work was partially supported by the Teaching and Research Award Program for Outstanding Young Teachers in Higher Education Institution, the National Research Foundation for the Doctoral Program of Higher Education of China (20030145029) of the Ministry of Education, China, and the National Natural Science Foundation of China (60273079).

References

[1] B. Sankur and M. Sezgin, "A survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, (to appear).

[2] O. D. Trier and A. K. Jain, "Goal-directed evaluation of binarization methods," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 17, pp. 1191–1201, 1995.

[3] O. D. Trier and T. Taxt, "Evaluation of binarization methods for document images," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 17, pp. 312–315, 1995.

[4] N. R. Pal and S. A Pal, "Review on image segmentation techniques," *Pattern Recognition*, Vol. 26, pp. 1277–1294, 1993.

[5] P. K. Sahoo, et al., "A survey of thresholding techniques," *Comput. Vis. Graph. Image Process.*, Vol. 41, pp. 233–260, 1988.

[6] N. A. Otsu, "Threshold selection method from grey-level histograms," *IEEE Trans. Syst., Man, Cybern.*, Vol. 8, pp. 62–66, 1979.

[7] X. Ye, M Cheriet, and C. Y. Suen, "Stroke-model-based character extraction from gray-level document images," *IEEE Trans. Image Processing*, Vol.10, pp.1152–1161, 2001.

[8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, New York, Wiley, 2001.

[9] S. Theodoridis, and K. Koutroumbas, *Pattern Recognition*, London, Academic Press, 2003.

[10] L. Dong, "An iterative algorithm for image thresholding," Technical Report #20031225, Department of Communications Engineering, Shengyang University, China, 2003.

[11] Computer vision test images, http://www-2.cs.cmu.edu/~cil/v-images.html.