An efficient iterative algorithm for image thresholding

Liju Dong a,1, Ge Yu b, Philip Ogunbona c,*, Wanqing Li c

a School of Information Science and Engineering, Shenyang University, Shenyang, PR China
b School of Information Science and Engineering, Northeastern University, Shenyang, PR China
c School of Computer Science and Software Engineering, University of Wollongong, Wollongong, Australia

Received 1 March 2007; received in revised form 17 December 2007
Available online 14 February 2008

Communicated by Y.J. Zhang

Abstract

Thresholding is a commonly used technique for image segmentation. This paper presents an efficient iterative algorithm for finding optimal thresholds that minimize a weighted sum-of-squared-error objective function. We have proven that the proposed algorithm is mathematically equivalent to the well-known Otsu’s method, but requires much less computation. The computational complexity of the proposed algorithm is linear with respect to the number of thresholds to be calculated as against the exponential complexity of the Otsu’s algorithm. Experimental results have verified the theoretical analysis and the efficiency of the proposed algorithm.

© 2008 Published by Elsevier B.V.

Keywords: Thresholding; Binarisation; Segmentation

1. Introduction

Thresholding is a simple and commonly used technique for image segmentation (Pal and Pal, 1993). It aims to group the image pixels by checking the pixel intensities against a set of thresholds. Searching for optimal thresholds with respect to a particular objective function has always been one of the fundamental problems in image processing. Numerous methods have been proposed in the past (Sankur and Sezgin, 2004; Chang et al., 1997; Papamarkos et al., 2000; Yin, 2002; Cheng et al., 2000; Sahoo, 1988). In particular, Otsu’s method (Otsu, 1979) is so far regarded as the best and most commonly employed global thresholding technique.

In Otsu’s method (Otsu, 1979), the objective function is defined using between-group variance, within-group variance or total variance of the pixel intensities and thresholds that maximizes the objective function are obtained through an exhaustive search. Due to the exhaustive search, Otsu’s algorithm bears exponential computation, \(O(c^L)\), where \(L\) is the number of intensity levels and \(c - 1\) is the number of thresholds.

In Section 2 of this paper, we cast the thresholding problem from the perspective of data clustering (Duda et al., 2001; Theodoridis and Koutroumbas, 2003) and formulate the problem so as to find the optimal thresholds that minimize a weighted sum-of-squared-error objective function. An efficient iterative algorithm is derived in Section 3. In addition, we prove that the proposed method is mathematically equivalent to Otsu’s method and its complexity is linear, \(O(cL)\). In Section 4, experimental results are presented and demonstrated that the proposed method is 200 times faster than Otsu’s method in searching for two thresholds from the histogram of an image. Furthermore it is demonstrated that the thresholds obtained from both methods are identical. Conclusions are given in Section 5.

2. Problem formulation and solution

Suppose there are \(L\) gray levels \(\{0, \ldots, L-1\}\) in an image. Let \(n_l\) denote the number of pixels at level \(l\). An ideal image containing three objects with distinct gray
levels will have only three non-zero \( n_l, l \in \{0, 1, \ldots, L-1\} \), as shown in Fig. 1a. However, the gray levels spread in a wide range for a real image, as shown in Fig. 1b. Image segmentation can therefore be formulated as finding the clusters on the histogram such that the total deviation of the gray levels from their corresponding cluster centers (centroids) is minimized, as shown in Fig. 1c.

Definition 1. Suppose that the histogram of an image is divided into \( c \) clusters (disjoint subsets) \( S_1, S_2, \ldots, S_c \). Let \( m_1, m_2, \ldots, m_c \) be the centroids of these clusters. The thresholding problem can be seen as a search for the thresholds that divide the histogram into \( c \) clusters, \( S_1, S_2, \ldots, S_c \). Let \( m_1, m_2, \ldots, m_c \) be the centroids of these clusters. The thresholding problem can be seen as a search for the centroids such that the total deviation of the gray levels from their corresponding cluster centers (centroids) is minimized, as shown in Fig. 1c.

We refer \( f(m_1, m_2, \ldots, m_c) \) as a weighted sum-of-squared-error function, where \( n_l \) serves as a weighting factor. The centroids, \( m_1, m_2, \ldots, m_c \), are calculated as

\[
m_i = \frac{1}{d_i} \sum_{l \in S_i} n_l,
\]

where \( d_i = \sum_{l \in S_i} n_l \), \( 1 \leq i \leq c \).

\[
f_i = \sum_{l \in S_i} n_l (l - m_i)^2
\]

is the weighted sum-of-squared-errors for cluster \( S_i \).

Assume that a gray level \( k \) currently in cluster \( S_i \) is tentatively moved to cluster \( S_j \). Then \( m_j \) changes to \( m_j' \), \( m_i \) changes to \( m_i' \), \( f_i \) changes to \( f_i' \), and \( f_j \) changes to \( f_j' \). After some mathematical manipulations, we have

\[
m_j' = \frac{1 + \frac{k - m_j n_k}{d_j + n_k}}{d_j + n_k} n_k + m_i d_j
\]

\[
m_i' = m_i + \frac{k - m_j n_k}{d_j + n_k}
\]

and \( f_j \) increases to

\[
f_j' = n_k (k - m_j')^2 + \sum_{l \in S_j} n_l (l - m_j')^2
\]

\[
= n_k \left( k - m_j - \frac{(k - m_j) n_k}{d_j + n_k} \right)^2
\]

\[
+ \sum_{l \in S_j} n_l \left( l - m_j - \frac{(k - m_j) n_k}{d_j + n_k} \right)^2
\]

\[
= n_k (k - m_j)^2 \frac{d_j^2}{(d_j + n_k)^2} + \sum_{l \in S_j} n_l (l - m_j)^2
\]

\[
+ \sum_{l \in S_j} n_l \frac{(k - m_j)^2 n_k^2}{(d_j + n_k)^2}
\]

\[
- 2 \sum_{l \in S_j} n_l (l - m_j) \frac{(k - m_j) n_k}{d_j + n_k}
\]

\[
= n_k (k - m_j)^2 \frac{d_j^2}{(d_j + n_k)^2} + \frac{d_j (k - m_j)^2 n_k^2}{(d_j + n_k)^2}
\]

\[
- 2 \frac{(k - m_j) n_k}{d_j + n_k} \left( \sum_{l \in S_j} n_l (l - m_j) \sum_{l \in S_j} n_l \right)
\]

\[
(5)
\]

According to (2) the last term of (6) equals zero. Therefore,

\[
f_j' = f_j + \frac{d_j n_k (k - m_j)^2}{d_j + n_k}
\]

Similarly, we have \( m_i' \) as

\[
m_i' = m_i - \frac{(k - m_j) n_k}{d_i - n_k}
\]

and \( f_i \) reduces to

\[
f_i' = f_i - \frac{d_i n_k (k - m_j)^2}{d_i - n_k}
\]

From (6) and (8), we see that the transfer of gray level \( k \) from cluster \( S_i \) to \( S_j \) can reduce \( f(m_1, m_2, \ldots, m_c) \) when

\[
d_i n_k (k - m_j)^2 > d_j n_k (k - m_j)^2
\]

The greatest reduction of \( f(m_1, m_2, \ldots, m_c) \) is achieved by choosing the cluster for which the value \( \frac{d_j n_k (k - m_j)^2}{d_j - n_k} \) is minimal. An optimal partition that minimizes the objective function \( f(m_1, m_2, \ldots, m_c) \) can be obtained through an iterative application of the reassignment for all \( L \) gray levels.

Let \( t_1, t_2, \ldots, t_{L-1} \) be \( c-1 \) thresholds that divide the histogram into \( c \) clusters, \( S_1, S_2, \ldots, S_c \), where \( 0 \leq t_1 \leq t_2 \leq \ldots \leq t_{L-1} \leq L-1 \) and the gray level, \( k \), of pixels in cluster \( S_i \) satisfies, \( k \in [t_{i-1}, t_i) \) for \( i = 0, 1, \ldots, c \).

Theorem 1. If the objective function \( f(m_1, m_2, \ldots, m_c) \) in (1) is minimized by a partition, \( S_1, S_2, \ldots, S_c \), then \( \sigma^2_W(t_1, t_2, \ldots, t_{L-1}) \) is minimized.
Fig. 2. Binarisation of a fingerprint image with one threshold obtained using the proposed algorithm and Otsu’s algorithm: (a) original, (b) the proposed algorithm, and (c) Histogram and threshold.
This algorithm reflects the idea of iterative improvement in minimizing the objective function $f(m_1, m_2, \ldots, m_c)$ as defined by (1). The optimization procedure repeats until no further improvement can be achieved. A good initial partition can reduce the number of iterations. Let the smallest and largest non-zero gray levels on the histogram be $l_{\text{min}}$ and $l_{\text{max}}$, respectively. A good initial partition can be obtained by equally dividing $[l_{\text{min}}, l_{\text{max}}]$ into $c$ clusters.

Fig. 3. Segmentation of a text image (bank cheque) with one and two thresholds obtained using the proposed algorithm and Otsu’s algorithm: (a) original, (b) one threshold: $T_{\text{new}} = T_{\text{Otsu}} = 170$, (c) two thresholds: $T_{\text{new}} = T_{\text{Otsu}} = 89; T_{\text{new}} = T_{\text{Otsu}} = 204$, and (d) histogram and the thresholds.

Fig. 4. Segmentation of an infrared image with one and two thresholds obtained using the proposed algorithm and Otsu’s algorithm: (a) original, (b) one threshold: $T_{\text{new}} = T_{\text{Otsu}} = 65$, (c) two thresholds: $T_{\text{new}} = T_{\text{Otsu}} = 60; T_{\text{new}} = T_{\text{Otsu}} = 150$, and (d) histogram and the thresholds.
It is not difficult to analyse the computational complexity of the algorithm. Step 1 or Step 16 can be computed in $O(L)$. The main computation is in Steps 3–12, which requires $O(cLQ)$ with $Q$ being the number of iterations, which is also the number of times that Step 2 is visited.

4. Experimental results

We applied both Otsu’s algorithm and the proposed iterative algorithm to over 100 images of different nature. All results have verified that the two algorithms are equivalent and produce identical thresholds. However, the proposed algorithm requires substantially lower computation than Otsu’s algorithm.

Fig. 2 shows the result of binarising a fingerprint image. Satisfactory threshold was obtained though the histogram did not have clear peaks. Fig. 3 illustrates the segmentation results of a text image (bank cheque) with one and two thresholds calculated by the proposed algorithm and Otsu’s algorithm. Texts in the image were well separated using the two thresholds. Results for a thermal infrared image from a video surveillance are shown in Fig. 4. In addition, an example of the segmentation of a remote sensing image is given in Fig. 5, where the image mainly includes three parts: mountain, lake, and snow.

Tables 1 and 2 summaries the complexity and average running time of the proposed algorithm and Otsu’s algorithm in the cases of one and two thresholds, where the running time was based on an implementation of the two algorithms on 1 GHz Pentium III PC. As discussed previously, the proposed algorithm is linear, $O(cLQ)$ with respect to the number of clusters, $c$, where Otsu’s algorithm is exponential, $O(cL^c)$; $L$ is the number of gray levels and $Q$ is the number of iterations required by the proposed algorithm. Although no theoretical bound for $Q$ has been identified, our empirical results on over 100 images has shown that $Q \leq 10$ for one threshold and $Q \leq 15$ for two thresholds. In addition, the running time has indicated that our proposed algorithm was more than 200 times faster than Otsu’s algorithm in the case of two thresholds.

5. Conclusion

This paper presents a linear algorithm to find optimal thresholds from image histograms. The algorithm has been theoretically proven to be equivalent to Otsu’s method, but
requires significantly lower computation, which makes it beneficial for real-time applications. In addition, the benefit of low computation becomes significant in cases where multiple thresholds are required.

References


