Gray Scale Image Segmentation using OTSU Thresholding Optimal Approach

Patil Priyanka Vijay
Department of Electronics & Telecommunication Engineering
D. N. Patel college of engineering, Shahada, MP-India

Prof. N. C. Patil
Department of Electronics & Telecommunication Engineering
D. N. Patel college of engineering, Shahada, MP-India

Abstract

Image segmentation is often used to distinguish the foreground from the background. Image segmentation is one of the difficult research problems in the machine vision industry and pattern recognition. Thresholding is a simple but effective method to separate objects from the background. A commonly used method, the Otsu method, improves the image segmentation effect obviously. It can be implemented by two different approaches: Iteration approach and Custom approach. In this paper both approaches have been implemented on MATLAB and give the comparison of them and show that both has given almost the same threshold value for segmenting image but the custom approach requires less computations. So if this method will be implemented on hardware in an optimized way then custom approach is the best option.

Keywords: OTSU, Image Segmentation, Thresholding

I. INTRODUCTION

Otsu’s method [1][2] is a very popular global automatic thresholding technique, which can be applied to a wide range of applications. These applications demand real-time performance and a hardware implementation is essential to increase the computational efficiency of the Otsu’s procedure. Many improved algorithms have been given till date. Xiaolu Yang has given an improved median-based Algorithm [3], WANG Hongzhi and DONG Ying has proposed new method for selection of optimal threshold value for defect detection[4], Ningbo Zhu1 and Gang Wang have proposed fast algorithm based on improved histogram[5].

In this paper, two different approaches have been implemented on MATLAB and given the comparison. The rest of this paper is organized as follows. Section 2 gives a brief review of Otsu method. In section 3, we describe the implementation of iteration approach and custom approach. Then, section 4 presents experimental results and analysis. Finally, the paper is summarized and conclusions are drawn in section 5 and section 6 gives future work.

II. OTSU METHOD

Otsu’s method is based on the principle that the gray-level for which the between-class variance is maximum or within class variance is minimum is selected as the threshold [7]. Let the pixels of a given picture be represented in $L$ gray levels $[0, 1, \ldots, L-1]$. The number of pixels at level $i$ is denoted by $n_i$ and the total number of pixels by $N = n_0 + n_1 + \ldots + n_{L-1}$. In order to simplify the discussion, the gray-level histogram is normalized and regarded as a probability distribution:

$$p_i = \frac{n_i}{N}, p_i \geq 0, \quad \sum_{i=0}^{L-1} p_i = 1 \quad (1)$$

Now suppose that we dichotomize the pixels into two classes $C_0$ and $C_1$ (background and objects, or vice versa) by a threshold at level $t$: $C_0$ denotes pixels with levels $[0, \ldots, t]$, and $C_1$ denotes pixels with levels $[t+1, \ldots, L-1]$. Then the probabilities of class occurrence and the class mean levels, respectively, are given by

$$\omega_0 = \Pr(C_0) = \sum_{i=0}^{t} p_i = \omega(t) \quad (2)$$

$$\omega_1 = \Pr(C_1) = \sum_{i=t+1}^{L-1} p_i = 1 - \omega(t) \quad (3)$$

And

$$\mu_0 = \sum_{i=0}^{t} i \cdot \Pr(i | C_0)$$
Gray Scale Image Segmentation using OTSU Thresholding Optimal Approach

Where \( \omega(t) = \sum_{i=0}^{t} p_i \) (6)

And

\[ \mu(t) = \sum_{i=0}^{t} ip_i \] (7)

are the zeroth- and the first-order cumulative moments of the histogram up to the \( t \) th level, respectively, and

\[ \mu_T = \mu(L-1) = \sum_{i=0}^{L-1} ip_i \] (8)

is the total mean level of the original picture. We can easily verify the following relation for any choice of \( t \):

\[ \omega_b\mu_0 + \omega_1\mu_1 = \mu_T, \quad \omega_b + \omega_1 = 1 \] (9)

The class variances are given by

\[ \sigma_b^2 = \sum_{i=0}^{t} \frac{(i - \mu_b)^2}{\omega_0} \Pr(i|C_b) = \sum_{i=0}^{t} \frac{(i - \mu_b)^2 p_i}{\omega_b} \] (10)

\[ \sigma_1^2 = \sum_{i=t+1}^{L} \frac{(i - \mu_1)^2}{\omega_1} \Pr(i|C_1) = \sum_{i=t+1}^{L} \frac{(i - \mu_1)^2 p_i}{\omega_1} \] (11)

These require second-order cumulative moments (statistics). In order to evaluate the "goodness" of the threshold (at level \( t \)), we shall take the following discriminant criterion measures (or measures of class separability) used in the discriminant analysis as shown in (12)

\[ \sigma_g^2 = \omega_b(\mu_1 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \] (12)

Then our problem is reduced to an optimization problem to search for a threshold \( t \) that maximizes the object functions (the criterion measures) in (12).

This standpoint is motivated by a conjecture that between class variance is a measurement to the difference between two parts. The greater the between-class variance is, the greater difference between the two parts. When we make a mistake between the objects and the background, between-class variance will decrease. As a result, making the between-class variance maximum means making the probability of error minimum, a threshold giving the best separation of classes in gray levels would be the best threshold. That is the Otsu algorithm.

The optimal threshold \( t^* \) that maximizes \( \sigma_g^2 \), is selected in the following sequential search for the values of \( t \) from 0 to \( L-1 \) by using the simple cumulative quantities (6) and (7), or explicitly using (2)-(5):

\[ \sigma_g^2(t) = \frac{\mu_T - \mu(t)}{\omega(t)} \] (13)

and the optimal threshold \( t^* \) is

\[ \sigma_g^2(t^*) = \Delta \tau \max_{0 \leq t \leq L-1} \left[ \sigma_g^2(t) \right] \] (14)

III. IMPLEMENTATION

A. Iteration Approach

Iteration Approach is exact representation of Otsu's method. The within-class variance is simply the sum of the two variances multiplied by their associated weights.

\[ \sigma_{within}^2(t) = \omega_b(t)\sigma_b^2 + \omega_1(t)\sigma_1^2 \] (15)

Where,

\[ \omega_b(t) = \sum_{i=0}^{t} p_i, \quad \omega_1(t) = \sum_{i=t+1}^{L} p_i \]

\( \sigma_b^2 \) = The variance of the pixels in the background (below threshold)
\( \sigma_1^2 \) = The variance of the pixels in the foreground (above threshold)
It has lot of computations.
B. Custom Approach

It’s Easier using simple recurrence relations. The threshold with the maximum between class variance also has the minimum within class variance. So it can also be used for finding the best threshold and therefore due to being simpler is a much better approach to use. Threshold is calculated by optimized formula of Otsu’s method.

\[
\begin{align*}
\omega_0(t+1) &= \omega_0(t) + p(t) \\
\omega_1(t+1) &= \omega_1(t) - p(t) \\
\mu_0(t+1) &= \frac{\mu_0(t) + p(t) \mu(t)}{\omega_0(t+1)} \\
\mu_1(t+1) &= \frac{\mu_1(t) - p(t) \mu(t)}{\omega_1(t+1)}
\end{align*}
\]  

(Where \( \mu = \omega_0 \mu_0 + \omega_1 \mu_1 \))

Between Class Variance

\[
\sigma^2 = \sigma^2_{\omega_0} - \sigma^2_{\omega_1} = \omega_0 (\mu - \mu)^2 + \omega_1 (\mu - \mu)^2
\]

\[
= \omega_0 \omega_1 (\mu - \mu)^2
\]

Implemented both approaches in MATLAB. The simulation results in MATLAB software for different images are shown in fig. (a), (b) and (c).

IV. Histogram

An image histogram is a type of histogram that acts as a graphical representation of the intensity distribution in a digital image. It plots the number of pixels for each intensity value. By looking at the histogram for a specific image a viewer will be able to judge the entire intensity distribution at a glance.

The horizontal axis of the graph represents the intensity variations, while the vertical axis represents the number of pixels at that particular intensity. The left side of the horizontal axis represents the black and dark areas, the middle represents medium gray and the right hand side represents light and pure white areas. The vertical axis represents the size of the area that is captured in each one of these zones. Thus, the histogram for a very dark image will have the majority of its data points on the left side and center of the graph. Conversely, the histogram for a very bright image with few dark areas and/or shadows will have most of its data points on the right side and center of the graph.

![Fig. 1: Histogram](image)

V. Experimental Results and Analysis

In this section, the performance of the both approach has been evaluated and compared. To evaluate the practical performance, both approaches have been implemented in MATLAB under windows 7 system. Experiments are performed on nearly 10 images to compare the results of both approaches. The few images are given below to know how the segmentation takes place.

As per the figures, figure (a) indicates the original images and figure (b) indicates the corresponding segmented images using Iteration approach and figure (c) indicates the corresponding segmented images using custom approach. d) Indicates the corresponding segmented image using MATLAB function as shown in Fig. The important point in this concept is in both the methods the obtained threshold value is nearly same, but these methods are differing in terms of computations. The iteration approach has lot of computations while custom approach has fewer computations.

VI. Simulation Result & Comparison Tables

Design Otsu algorithm in MATLAB 7.0. MATLAB is a data analysis and visualisation tool designed to make matrix manipulation as simple as possible. In addition, it has powerful graphics capabilities and its own programming language. The basic MATLAB distribution can be expanded by adding a range of toolboxes, the one relevant to this course is the image-processing toolbox (IPT). Result carry out of different Image and compare with MATLAB function. The basic distribution and all of the currently available toolboxes are available in the labs. The basic distribution plus any installed toolboxes will provide a large selection of functions, invoked via a command line interface.

Table – 1

Simulation Results
### Gray Scale Image Segmentation using OTSU Thresholding Optimal Approach

**Iteration Approach**

**Custom Approach**

**MATLAB function**

<table>
<thead>
<tr>
<th>Image 1</th>
<th>Iteration Approach</th>
<th>Custom Approach</th>
<th>MATLAB function</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Gray Image" /></td>
<td><img src="histogram1" alt="Histogram" /></td>
<td><img src="binary_histogram1" alt="Histogram of Binary Image" /></td>
<td><img src="cumulative_sum1" alt="Cumulative Sum" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image 2</th>
<th>Iteration Approach</th>
<th>Custom Approach</th>
<th>MATLAB Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image2" alt="Gray Image" /></td>
<td><img src="histogram2" alt="Histogram" /></td>
<td><img src="binary_histogram2" alt="Histogram of Binary Image" /></td>
<td><img src="cumulative_sum2" alt="Cumulative Sum" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image 3</th>
<th>Iteration Approach</th>
<th>Custom Approach</th>
<th>MATLAB function</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3" alt="Gray Image" /></td>
<td><img src="histogram3" alt="Histogram" /></td>
<td><img src="binary_histogram3" alt="Histogram of Binary Image" /></td>
<td><img src="cumulative_sum3" alt="Cumulative Sum" /></td>
</tr>
</tbody>
</table>
The Otsu method is one of the very efficient methods to threshold the gray images. Comparison of iteration approach and custom approach to implement Otsu thresholding method has been given for image segmentation. The custom approach maximizes a modified between-class variance instead of maximizing the conventional between-class variance as a criterion. For M multilevel threshold selection, we showed that maximizing the modified between-class variance has less computation than maximizing the conventional between-class variance. The formulas for ω and µ for the modified between-class variance are written in recursive form, which reduces the complexity of computation for ω and µ. Thus, for image segmentation using custom approach is better than the iteration approach which has lots of computation.

REFERENCES