An improved OTSU method based on Genetic Algorithm

Wei SHANG[1,a], Yan-fen CHENG[2,b]

[1,2]Wuhan University of Technology No.1186, Heping Boulevard, Wuchang District, Wuhan, Hubei, China

[a]1012691657@qq.com, [b]995132428@qq.com

Keywords: Image thresholding, Otsu method, Genetic Algorithm.

Abstract. Image segmentation is required to be studied in detail some particular features (areas of interest) of a digital image. It forms an important and exigent part of image processing and requires an exhaustive and robust search technique for its implementation. In the present work we have studied working of image thresholding based on genetic algorithm. The proposed variant in this paper is tested on a set of images and the results are compared with original Otsu method.

Introduction

Threshold selection plays an important role in image segmentation; relevant experts have widely studied on the threshold automatic selection. The most useful methods in practice are mainly as minimum error method, iterative method, entropy method and Otsu method. [1,2,3] The original Otsu method might be as fast as possible, but it takes exhaustive optimization method and extends the calculation time. It's not good enough to solve the situation of foreground and background crossing in gray level statistics. The Otsu method mainly include: ant colony algorithm, BP neural network, genetic algorithm, expert system, etc... The genetic algorithm [4] uses the probabilistic optimization algorithm, in which the global space is searched to find the optimal solution. In this paper, taking the advantage of genetic algorithm and Otsu method, the author put forward a method to automatic select the optimal threshold for image using improved Otsu algorithm with genetic algorithm.

Related Work

Otsu method is an adaptive threshold selection method, this method was proposed by Japanese scholar Nobuyuki Otsu in 1979. The principle of the method is that the image is divided into two classes, the target and the background by searching an optimal threshold in the range of image gray level. Its evaluation criterion is the between-class variance of the two classes. The larger the variance, the greater the difference between the target and the background and the better the segmentation results. The smaller the variance, the more probability the two classes to be divided and the worse the segmentation results. [5]

Suppose that the gray level is L(used to be 256), and the number of pixels of gray level at i is n_i, then the number of total pixels of an image is $N = \sum_{i=0}^{L-1} n_i$, the probability that each gray level appears is $p_i = \frac{n_i}{N}$. The Otsu method selects gray level k as the segmentation threshold, and divides the image into two classes: the background class named W_1 which gray level ranges from 0 to k,

the target class named W_2 which gray level ranges from $k+1$ to $L-1$. The gray level probability distributions for the two classes are given as:

$$P_{w1} = \sum_{i=0}^{k} p_i \tag{1}$$

$$P_{w2} = \sum_{i=k+1}^{L-1} p_i = 1 - P_{w1} \tag{2}$$

The mean gray level of W_1 and W_2 are:

$$\mu_{w1} = \sum_{i=0}^{k} \frac{i*p_i}{P_{w1}} \tag{3}$$

$$\mu_{w2} = \sum_{i=k+1}^{L-1} \frac{i*p_i}{P_{w2}} \tag{4}$$

The mean of total gray levels is denoted by μ:

$$\mu = P_{w1}\mu_{w1} + P_{w2}\mu_{w2} = \sum_{i=0}^{L-1} i * p_i \tag{5}$$

The between-class variance is:

$$\sigma^2(k) = P_{w1}(\mu_{w1} - \mu)^2 + P_{w2}(\mu_{w2} - \mu)^2 \tag{6}$$

Otsu method chooses the optimal threshold k by maximizing the between-class variance $\sigma^2(k)$. The larger the variance, the better the image segmentation.

There is a disadvantage that it's inefficient when the calculation is complex, as we need to traverse the gray level ranging from 0 to $L-1$ to find the best threshold. GA(Genetic algorithm) is a kind of optimization algorithm having a strong searching ability. Using GA to find the best threshold could improve the speed of image segmentation. [6]

Proposed Methodology

Reducing the searching range of threshold

It's important to choosing a reasonable range of threshold to optimize the method. Firstly, it can eliminate a large part of the gray levels to reduce the calculation job by choosing the range of threshold. Secondly, it can exclude the low or high gray levels to reduce the noise and the rate of choosing wrong threshold.

In this paper, we take the mean gray level μ as the initial threshold, take the pixels having gray levels ranging from 0 to μ as the background class W_1, and take the pixels having gray levels ranging from $\mu+1$ to $L-1$ as the target class W_2. Then we use the mean gray level T_1 of class W_1 and T_2 of class W_2 as the range of threshold.

Now, the gray level probability distributions for the two classes are:

$$P_{w1} = \sum_{i=0}^{\mu} p_i \tag{7}$$

$$P_{w2} = \sum_{i=\mu+1}^{L-1} p_i \tag{8}$$

Mean gray level T_1 and T_2 are:

$$T_1 = \sum_{i=0}^{\mu} \frac{i*p_i}{P_{w1}} \tag{9}$$

$$T_2 = \sum_{i=\mu+1}^{L-1} \frac{i*p_i}{P_{w2}} \tag{10}$$

Beyond this, we could select half of the restrained gray levels to manipulate the rest calculation. As Fig. 5 (a) and (b) show, the histogram of between-class variance is smooth, we choose the even gray levels. Once we get the best threshold in these gray levels, we still need to calculate the field values.

The consequence is shown in Fig. 1:

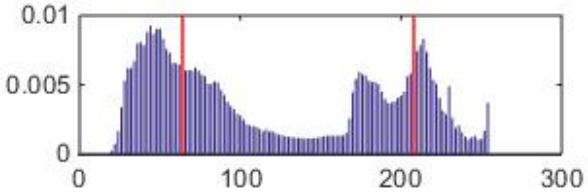


Figure 1. Histogram of gray levels restrained

In Fig. 1, the blue bars represent the even gray levels, while the red bars represent the reduced searching range of threshold.

Improving Otsu method combining with GA

Step 1: Code

In order to use GA, it is necessary to encode the value of the solution space to produce the chromosome element. Since the gray scale image contains 256 gray levels, it corresponds to one byte, so one byte is used as a chromosome. The decoding process of chromosome is exactly the inverse of the encoding process, which is the decimal number of byte.

Step 2: Initialize population

We produce 10 populations in the optimal threshold range $[T_1, T_2]$. The population is obtained by average distribution.

Step 3: Calculate the fitness

In this paper, we take the maximum of variance $\sigma^2(k)$ as the goal, so we use $\sigma^2(k)$ as individual fitness.

Step 4: Select and intercross population

We use traditional method in GA to select and intercross population. In selection, we use the probability proportional to fitness to determine the number of individual copied to next generation population.

Step 5: Mutation

Some genetic variation may result in the loss of some effective genes. In order to recover these effective genes and maintain the diversity of the population, improve the mutation rate when the individual is in a local optimum and reduce the rate of variation for high fitness individual, we use Hamming distance in mutation.

Hamming distance is defined as follow [7]:

$$H_{i,j} = \sum_{k=1}^{n} \left| a_{ik} - a_{jk} \right| \tag{11}$$

In this equation, n represents length of the binary number of an individual (n equals 8 in this paper), a_{ik} is the value of the k'th bit of individual i. According to the Hamming distance, the mutation rate is defined as:

$$P_m = \begin{cases} P_{m1} - \frac{P_{m1}-P_{m2}}{f_m-f_a} \times \frac{2(f_m-f)|H_{i,j}-H_a|}{H_m-H}, & f \geq f_a \\ P_{m1} \times \frac{2(H_{i,j}-H_a)}{H_m-H_a}, & f < f_a \end{cases} \quad (12)$$

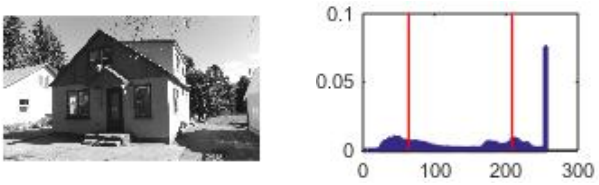
f is the individual fitness value to be mutated. f_m is the largest individual fitness. f_a is the average fitness value of each generation. $H_{i,j}$ is the absolute Hamming distance between two individuals. H_a is the average absolute Hamming distance between individuals. H_m is the largest absolute Hamming distance of the population. When the individual fitness of variation is greater than the average fitness, the variation rate of individual is smaller. When the individual fitness is less than the average fitness, the greater the variation rate is. Hamming distance implements the adaptive mutation of the individual and improves the efficiency of the algorithm.

Step 6: Ending condition

The algorithm terminates when the evolutionary algorithm computes all the thresholds in the optimal threshold range $[T_1, T_2]$, or if the optimal individuals within the population have not changed in 5 generations of evolution. Otherwise, go to step 3.

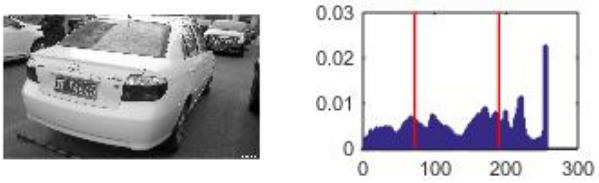
Simulation and result

For reducing the searching range of threshold we could find that it works in mostly condition. Here are the two gray scale images and their searching range of threshold of test samples.



(a) (b)

Figure 2. Sample One (a) Gray scale image (b) Histogram



(a) (b)

Figure 3. Sample Two (a) Gray scale image (b) Histogram

In these two samples, we could find that the original searching scope ranges from 0 to 255. After reducing the range, sample one has interval $[64, 208]$ and sample two has interval $[72, 190]$. The intervals are shown in Fig. 2 (b) and Fig. 3 (b) between the red lines.

Through these two aspects of optimization, we get some samples to test the improved method.

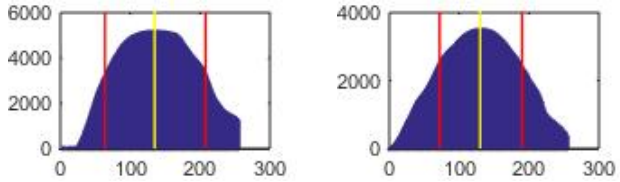
Firstly, we could find that the proposed method is effective in finding the threshold. The binary images are shown below:

(a) (b)

Figure 4. Binary image (a) Sample 1 (b) Sample 2

The best threshold found by this method is the same as original Otsu method does. Thus the binary images are the same.

Second, from the two histograms of between-class variance below, we could see that the reduced intervals involving the largest between-class variances.



(a) (b)

Figure 5. Histogram of between-class variance (a) Sample 1 (b) Sample 2

In Fig. 5 (a) and (b), the x axis of yellow line represents the largest between-class variance.

In order to compare with the original Otsu method, we list some data below:

Table 1. Data comparison between proposed method and the original Otsu method

	Sample 1		Sample 2	
Image resolution	1920*1080		1920*1080	
Method	Author's	Original	Author's	Original
Variance calculation (times)	73	256	50	256
Time consuming in Matlab (s)	0.174	0.237	0.152	0.206
Searching range	$[64, 208]$	$[0, 255]$	$[72, 190]$	$[0,256]$
Best/Largest threshold	135	135	130	130
Time cut off	27%	-	26%	-

From Table 1, we can find that the proposed method running faster than the original method. It is about 26% faster than the original in most condition. In sample 1, as the optimal individual is uncertain after 5 generations, we calculated 73 times in $[64, 208]$.

Summary

Threshold selection is a very basic method in image segmentation. In some place such as traffic monitoring, there are lots of continuous real-time computing. The improved Otsu method can reduce

the calculation time eventually. However, one-dimensional Otsu method is not suitable for the situation that the one-dimensional gray-scale histogram does not contain obvious peaks and troughs. At this time, two-dimensional Otsu method is recommended. As the two-dimensional Otsu method is too slow, our future work is aiming at speeding up this method.

References

[1] Kurita, T., N. Otsu, and N. Abdelmalek. "Maximum likelihood thresholding based on population mixture models." Pattern Recognition 25.10(1992):1231-1240.

[2] Xue, J. H., and D. M. Titterington. "t-Tests, F-tests and Otsu's methods for image thresholding. " IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society 20.8(2011):2392-2396.

[3] Sezgin, Mehmet, and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging, 2004.

[4] Goldberg, David E. Genetic algorithms in search, optimization, and machine learning /. Addison-Wesley Pub. Co. 1989.

[5] Kumar, Sushil, et al. "Bi-level thresholding using PSO, Artificial Bee Colony and MRLDE embedded with Otsu method." Memetic Computing 5.4(2013):323-334.

[6] Hammouche, Kamal, M. Diaf, and P. Siarry. "A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation." Computer Vision & Image Understanding 109.2(2008):163-175.

[7] Hamming, R. W. Error detecting and error correcting codes. Bell System Tech. J. 29, (1950). 147–160.