The University of Aizu

Advanced Image Processing and Algorithm

Final project

Lecturer:	Prof. R. Oka
	Prof. T. Hyashi
ID:	m5128110
Name:	Fu, Yu-Hsiang
Date:	2009/02/20

Contents

1. Brief introduction of program

In this final project report, it shows the implement of Otsu method. I chose the original Otsu method to implement the multi-threshold by using graph structure and depth first search (DFS) algorithm.

1.1. Implement schema

The schema is very simple. It consists of two parts that the first part is the single threshold and the second is the multi-threshold. The single threshold is the original procedure of two class of Otsu method. The multi-threshold is implemented by using graph structure and DFS algorithm. The simple schema is as follows:

```
if(T==1){//Single threshold or not
    SingleThreshold();//Two classes
}
else{
    MultiThresholds();//Multi classes
}
```

In the *MultiThresholds()* , it will travel all possible combination of parameters of multi-threshold which can be represent by graph structure. For example, $L=5$, $M=4$, the graph structure which is represented by matrix is as follows:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

In the DFS algorithm which is modified is as follows:

```
for(a=0;a<(L-(K-1));a++){//Each level
   dfs(a);
}
```

and

```
dfs(int v){
   if(counter<K){
      for(int a=0;a<L;a++){
         if(a>v && visited[a]==0){ //Next node must greater than current
            if(counter==(T-1)){
               doMultiThresholding(t);//Each combination
            }
        dfs(a);
}}}}
```

The *counter* is index of threshold array. The *visited[]* is an array to record which node is travelled.

1.2. Environment

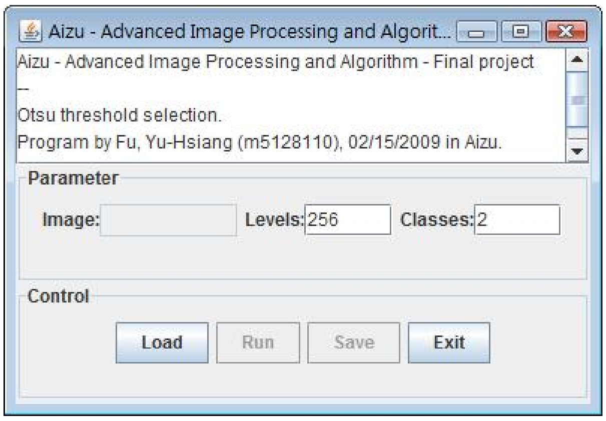
The experiment environment in this paper is HP Presario CQ45.The hardware is performed on Intel Dual Core Processor P7350 CPU, 4GB RAM and Microsoft Windows Vista Ultimate SP1 operating system. The software is performed on Java 1.6.0 Update 11 with IDE Gel RC40.

1.3. Operation steps

The program can be executed by using double-click on the “OtsuThresholdSelection.jar”. The operation steps of program are as follows:

Step1. “Load” an image.
Step2. Set the parameters.
Step3. “Run” the Otsu method.
Step4. “Save” the processed image
Step5. Repeat step2 to step4.
Step6. “Exit”.

1.4. Screenshots



Aizu - Advanced Image Processing and Algorit...

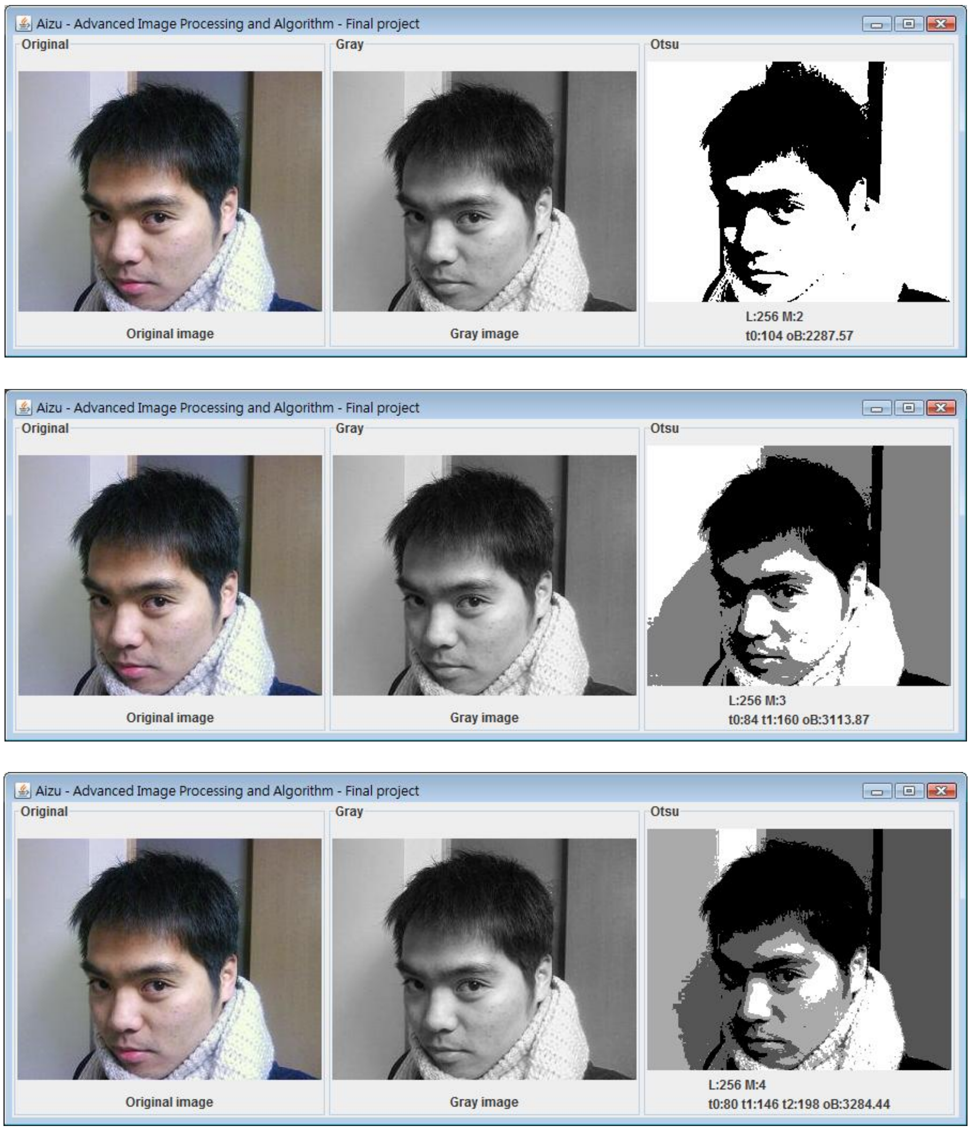
Aizu - Advanced Image Processing and Algorithm - Final project
--
Otsu threshold selection.
Program by Fu, Yu-Hsiang (m5128110), 02/15/2009 in Aizu.

Parameter

Image: Levels: 256 Classes: 2

Control

Figure 1. The main window



Aizu - Advanced Image Processing and Algorithm - Final project

Original image | Gray image | L:256 M:2 t0:104 oB:2287.57

L:256 M:3 t0:84 t1:160 oB:3113.87

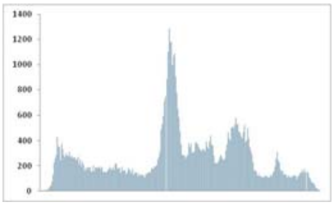
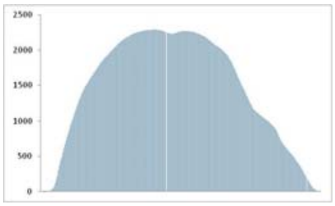
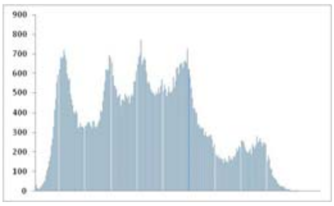
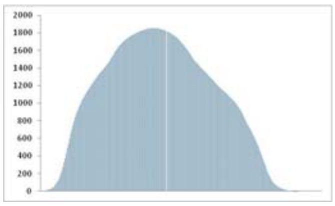
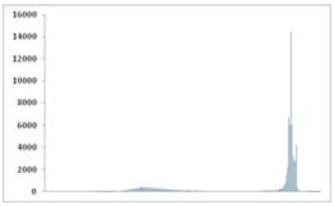
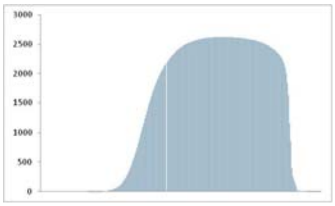
L:256 M:4 t0:80 t1:146 t2:198 oB:3284.44

Figure 2. The results of each execution

2. Results of the implement

2.1. Test images

Table 1. The test images

File	Image	Histogram of gray level	Histogram of σ_B^2
alaric.jpg			
lena.jpg			
coins.jpg			

2.2. Results of test images

The table 2 shows the result of each execution with threshold(s), between-class variance variable σ_B^2 and time.

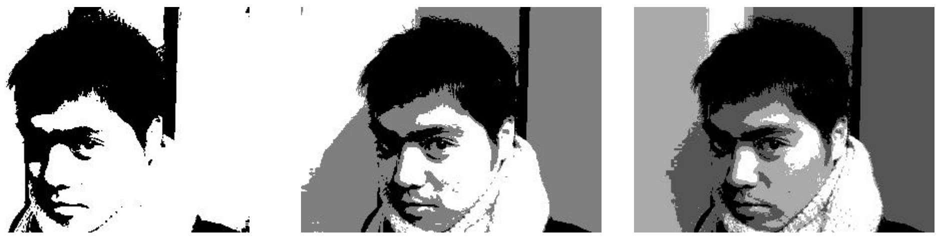
Table 2. The results of test images

Image	$M=2$			$M=3$			$M=4$		
	T	σ_B^2	$Time$	T	σ_B^2	$Time$	T	σ_B^2	$Time$
Alaric	t_0:104	2287.56	0.015	t_0:84	3113.87	0.684	t_0:80	3284.44	58.673
				t_1:160			t_1:146		
							t_2:198		
coins	t_0:166	2621.45	0.007	t_0:122	2714.55	0.893	t_0:106	2746.92	62.196
				t_1:188			t_1:148		
							t_2:201		
Lena	t_0:103	1853.06	0.008	t_0:79	2307.08	0.766	t_0:58	2503.40	68.813
				t_1:145			t_1:107		
							t_2:159		

Time unit: ms (milisecond)

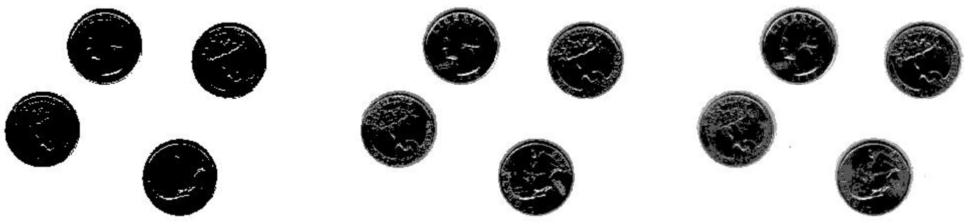
2.3. Results of processed images

The figures which are from Fig.3 to Fig.5 show the result of each execution with the variable M is form 2 to 4.



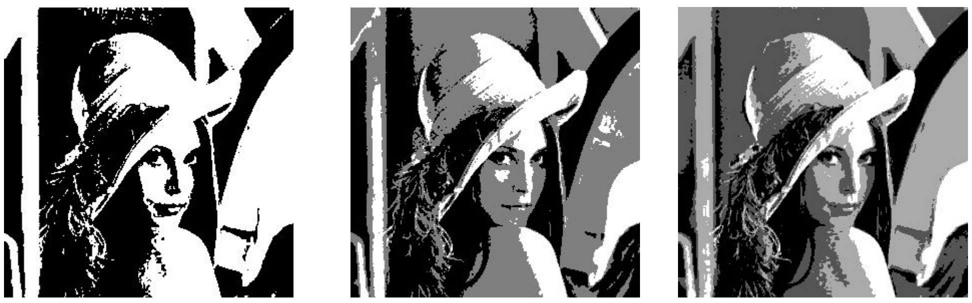
$M = 2$ $M = 3$ $M = 4$

Figure 3. The results of image, Alaric



$M = 2$ $M = 3$ $M = 4$

Figure 4. The results of image, coins



$M = 2$ $M = 3$ $M = 4$

Figure 5. The results of image, Lena