AN EFFICIENT IMAGE SEGMENTATION TECHNIQUE BY FAST SCANNING AND ADAPTIVE MERGING

[1]*Jian-Jiun Ding(丁建均)*,[2]*Cheng-Jin Kuo*(*郭政錦*), [3]*Wen-Chih Hong*(*洪文琦*)

Graduate Institute of Communication Engineering, National Taiwan University

E-mail: [1]djj@cc.ee.ntu.edu.tw, [2] j96921039@ntu.edu.tw, [3] r97942118@ntu.edu.tw

ABSTRACT

Image segmentation is the front stage of many works in image processing, such as object-orient compression. Based on these requirements, a good image segmentation algorithm should have the following three advantages: (1) fast speed, (2) good shape connectivity, and (3) good shape matching. In this paper, we proposed an efficient segmentation algorithm that can achieve the three goals at the same time. With the proposed method, each pixel only has to be scanned one time. The adjustable region mean and the adaptive threshold are also used for improving the performance. Our algorithm can avoid the over-segmented problem of the watershed method. The segmentation results of our method are as well as those of the region growing method, but the running time is 120 times less. We have shown several simulation results and proven that our method does work well.

1. INTRODUCTION

Image segmentation is a very important issue among digital image processing [1][2][3]. The reason we segment images is often for further image compression or simply for image recognition. In some situation, image segmentation is concerned for a specified range of an image but not the whole image. When we are interesting in recognizing some part of the image, we use image segmentation which is like this.

Different from the description above, in this paper we develop a simple algorithm of image segmentation for the whole image. The purpose we would like to develop a new segmentation algorithm is that we want to make a better and more convenient environment for us to compress the original image after we segment it. For this purpose, the goal of the new algorithm must fit some characteristics.

The first is the speed. Since we have defined the role of image segmentation which would be the former stage of the compression stage, we do not want to waste much time when we segment it. Therefore, the speed of the new algorithm must be fast.

The second, indeed, we hope the new algorithm is **reliable**, which means the results have a good shape matching even under the fast speed.

The third, we hope the result of segmenting shape of the new algorithm will be intact but **not fragmentary** which means the new algorithm has good connectivity of its results. This is very important because bad connectivity of the segmenting result is the disadvantage of most of the fast image segmentation algorithm nowadays. The reason we do not want to have the fragmentary segmenting result is that once we send them into the advanced compression stage, we have to waste a lot of resource to record the boundaries of them though the over-segment results ensure the good shape matching of the segmenting results and the compression ratio in the compression stage at the same time.

There are advantages and disadvantages for all of the algorithms nowadays themselves. In this paper we propose a new method. We combine the characteristics of them and develop a new algorithm with a simple concept. And we will show that the new algorithm can achieve the three goals listed above which is fast with good shape matching and good connectivity of its segmenting results.

In section 2, we brief review the region-based image segmenting methods, including region growing, some data clustering methods and watershed algorithm.

In section 3, it is one of the most important parts of this paper. We propose an image segmenting method to fit the request of image compression.

In section 4, we also propose an adaptive threshold decision method to improve it and develop a method to decide “local threshold” by “local variance”.

2. REVIEW OF EXISTING METHODS

There are several basic ways to segment images nowadays. Traditionally, these methods could be classified as below：Threshold Technique, Edge-Based Segmentation, Region-based Segmentation. In the below subsections, we briefly introduce three common methods: region growing, k-means algorithm and watershed algorithm.

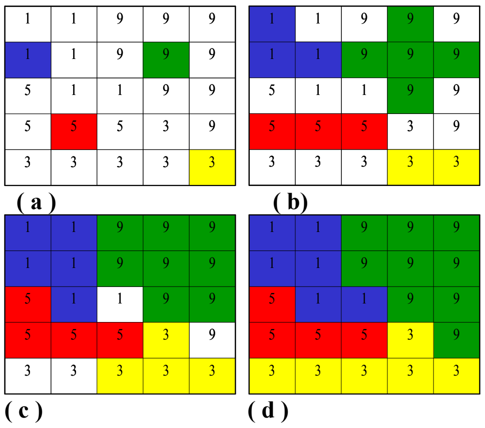

Fig. 1: Region growing

2.1. Region Growing

Region growing is one of the simplest region-based image segmentation methods and it can also be classified as one of the pixel-based image segmentations because it involves the selection of initial seed points.

Region growing segmentation is an approach to examine the neighboring pixels of the initial "seed points" and determine if the pixels are added to the seed point or not. The process is iterated as same as data clustering.

Since the regions are grown on the basis of the threshold, the image information is important for us. For example, getting to know the histogram of the image would help us a lot since we can take it as a reference to choose the threshold. There is a very simple example followed below.

In Fig. 1, the criteria we made are the same pixel value. That is, we keep examining the adjacent pixels of seed points. If they have the same intensity value with the seed points, we classify them into the seed points. It is an iterated process until there are no changes between two successive iterative stages. Of course, we can make other criteria, but the main goals are all the same which is to classify the similarity of the image into regions.

It is a very reliable algorithm, for we can define our own criteria to a characteristic we would like to classify and the result will not be fragmented. However, the disadvantage of it is time consuming. Since the region growing method was proposed not for segmenting the whole image, the speed of processing the whole image using region growing method is very slow.

2-2. K-Means Algorithm

Before introducing K-means algorithm, we review the concept of data clustering. Data clustering is one of the common used methods of Region-Based image segmentation, and it is widely use in mathematics and statistic field.

Table 1: Comparison of Segmentation

	Advantage	**Disadvantage**
Seed region growing	Reliable	Time consuming Over-segmentation
Hierarchical clustering	Reliable	Time consuming
K-means	Fast	Fragmented shape
Watershed	Reliable	Over-segmentation

There are two kinds of system we used to realize the concept of data clustering which are **hierarchical clustering** and **partitional clustering** [4].

The K-mean algorithm method is the most famous partitional clustering algorithms [5]. When using the method, we have to decide the numbers of cluster before performing the segmentation. According to the number of classes, we have to minimize some criteria of each cluster. K-means is a very fast algorithm which can classify the main database by parallel dealing the process with different initial points though it causes the initial problem.

However, there is one critical disadvantage in K-means which is the main reason we would not like to choose it for our "compression-oriented" segmentation. That is the segmented results of K-means are fragmented isolated sections but not a complete one.

2.3. Watershed Algorithm

Watershed is one of edge-based image segmentation. The main goal of watershed image segmentation is to find the "watershed lines" in an image. The two main properties of watershed segmentation result are continuous boundaries and over-segmentations.

Over-segmentation is the serious problem of using the watershed algorithm. It is due to the large numbers of potential minima. As we know, the boundaries that made by the watershed algorithm are exact the watershed lines in the image. Therefore, the numbers of region basically will be equal to the numbers of minima in the image.

We hope the result of the segmentation can be some kind of good result for our "compression-oriented" image processing. Therefore, all of the methods we discuss above are not appropriate. As we can see in Table 1, region growing and hierarchical data clustering are too slow, K-mean will segment the image into countless fragmented regions, and the watershed method causes the over-segmented problem. All of the result of those methods are not we want.

3. PROPOSED METHOD

The main goal for us to segment an image is that we would like to **create a more convenient status and environment for us to do deeper image compression**

or other processing. We hope the result of the segmentation can be some kind of good result for our "compression-oriented" image processing. Therefore, all of the methods we discuss above are not appropriate. As in Table 1, region growing and hierarchical data clustering are too slow, K-mean will segment the image into countless fragmented regions, and the watershed method causes the over-segmented problem. All of the result of those methods are not we want.

We propose a simple method here. We use the merging concept to scan the whole image and see if we can merge the pixel into an exist clustering. We can assign the threshold as we want. At the same time, this algorithm would not waste time in the stage of segmenting image comparing with other image processing stages.

It is a simple concept and we list the steps of algorithm as below. We use $C[m, n]$ ($m = 1, 2, \ldots, M$, $n = 1, 2, \ldots, N$) to denote the value of the pixel $[m, n]$, use $R[m, n]$ to denote the pixel $[m, n]$ is classify into which region, use $A(j)$ to denote the mean of pixels in the j^{th} region, and use $B(j)$ to denote the number of pixels in the j^{th} region.

(Step 1): Classify the first pixel [1, 1] as Region 1, as Fig. 2(a). We set $R[1, 1] = 1$, $A(1) = C[1, 1]$, $B(1) = 1$, $m = 1$, $n = 1$, and $j = 1$.

(Step 2): Then, set $n = n+1$ and scan the next pixel. If $R[1, n-1] = j$ and

- **Case 1**: $|C[m, n] - A(j)| \le \text{threshold}$, (1)

then set $R[m, n] = j$ and set

$$A(j) = \{A(j)\,B(j) + C[m, n]\}/(\,B(j) + 1)$$
$$B(j) = B(j) + 1. \tag{2}$$

For the example in Fig. 2(b), if threshold = 25, then $|C[1, 2] - C[1, 1]| \le \text{threshold}$, thus, the pixel [1, 2] is also classified into Region 1. If

- **Case 2**: $|C[m, n] - A(j)| > \text{threshold}$, (3)

then set $R[m, n] = j+1$, $A(j+1) = C[m, n]$, $B(j+1) = 1$, and $j = j+1$.

For the example in Fig. 2(c), since $C(1, 4) = 80$, the mean of Region 1 is 253, $|C(1, 4) - A(1)| >$ threshold. Thus, we should assign the pixel [1, 4] as a new region, i.e., Region 2.

(Step 3): Repeat Step 2 until $n = N$.

For the example in Fig. 2(d), the first row is classified into three regions.

(Step 4): Then, set $m = m+1$, $n = 1$, and scan the first pixel in the next row. If $R[m-1, 1] = i$ and

- **Case 1**: $|C[m, n] - A(i)| \le \text{threshold}$, (4)

then set $R[m, n] = i$ and

$$A(i) = \{A(i)\,B(i) + C[m, n]\}/(\,B(i) + 1)$$
$$B(i) = B(i) + 1. \tag{5}$$

If

- **Case 2**: $|C[m, n] - A(j)| > \text{threshold}$, (6)

then set $R[m, n] = j+1$, $A(j+1) = C[m, n]$, $B(j+1) = 1$, and $j = j+1$.

(Step 5): Then, set $n = n+1$ and scan the next pixel. In this case, $m \ne 1$ and $n \ne 1$ and we should compare $[m, n]$ with the upper region and the left region. Suppose that $R[m-1, n] = i$ and $R[m, n-1] = j$.

- **Case 1**: $|C[m, n] - A(i)| \le \text{threshold}$,
  $|C[m, n] - A(j)| > \text{threshold}$. (7)

In this case, we set $R[m, n] = i$ and use (5) to adjust the values of $A(i)$ and $B(i)$.

- **Case 2**: $|C[m, n] - A(i)| > \text{threshold}$,
  $|C[m, n] - A(j)| \le \text{threshold}$. (8)

In this case, we set $R[m, n] = j$ and use (2) to adjust the values of $A(j)$ and $B(j)$.

- **Case 3**: $|C[m, n] - A(i)| > \text{threshold}$,
  $|C[m, n] - A(j)| > \text{threshold}$. (9)

In this case, we set $R[m, n] = j$, $A(j+1) = C[m, n]$, $B(j+1) = 1$, and $j = j+1$.

- **Case4**: $|C[m, n] - A(i)| \le \text{threshold}$,
  $|C[m, n] - A(j)| \le \text{threshold}$. (10)

When $i = j$, we just set $R[m, n] = i$ and use (5) to adjust the values of $A(i)$ and $B(i)$.

However, when $i \ne j$, we must **merge** Region i with Region j. In this case, we set $R[m, n] = i$, but we should reset all the pixels $[m_0, n_0]$ that satisfy $R[m_0, n_0] = j$ as

$$R[m_0, n_0] = i \tag{11}$$

at the same time. Moreover,

$$A(i) = \{A(i)B(i) + A(j)B(j) + C[m, n]\}/(B(i)+B(j)+1), \tag{12}$$

$$B(i) = B(i)+B(j)+1, \tag{13}$$

$$B(j) = 0. \tag{14}$$

For the example in Fig. 2(e), for the pixel $C[2, 4] = 85$, since the means of Region 2 (yellow color) and Region 4 (green color) are 80 and 81.5, respectively, $|C[2, 4] - 80| <$ threshold and $|C[2, 4] - 81.5| <$ threshold. Therefore, we assign [2, 4] to Region 2 and **merge** Region 2 with Region 4, as in Fig. 2(f).

(Step 6): Repeat Step 4 and Step 5 until all the pixels in the image have been scanned.

For the example in Fig. 2(g), after all the pixels are processed, we classified the image into seven regions.

(Step 7): If $B(i) < \Delta$, we delete Region i and assign the pixels in Region i to the adjacent regions. Sometimes, the isolated dots (due to details or noise) of an image may cause over-segmentation. This step can avoid the problem.

For example, in Fig. 2(h), we set $\Delta = 4$ and the regions with green and purple colors in Fig. 2(g) are merged with larger regions.

(Step 8): Sort the regions according to $B(i)$, i.e., the number of pixels within them.

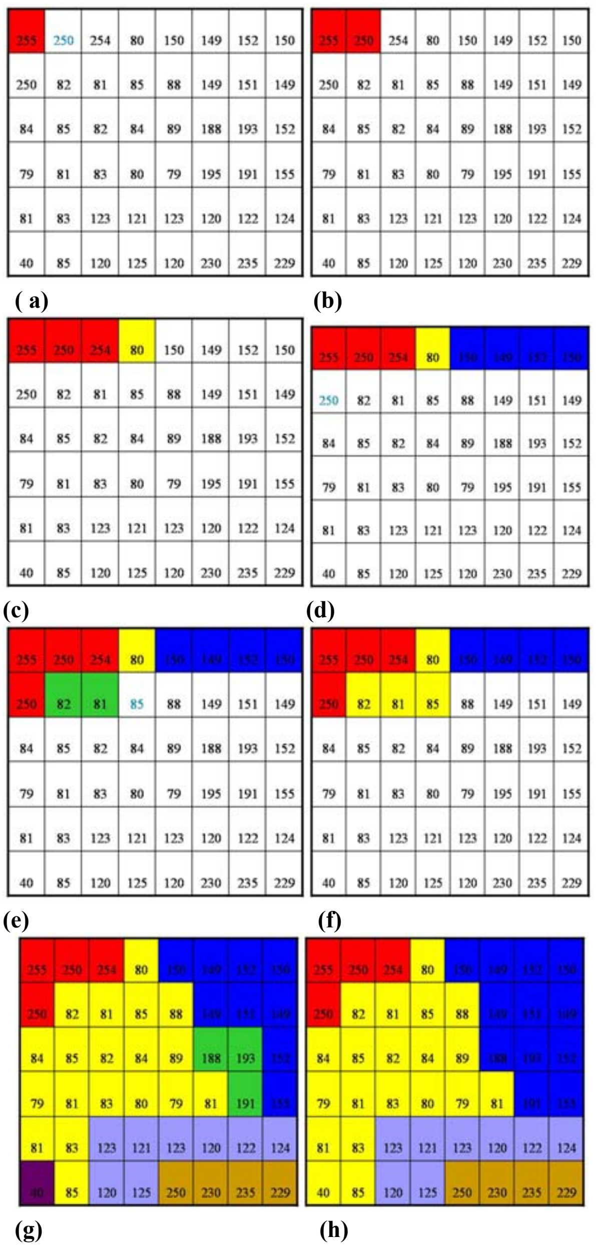
Fig. 2: Illustration of the process of our method for a simple digital image (threshold = 25).

The above process can also be applied for the color image. We only have to modify $|C[m, n] - A(i)|$ as

$$\{|C_R[m, n] - A_R(i)| + |C_G[m, n] - A_G(i)| + |C_B[m, n] - A_B(i)|\}/3, \tag{15}$$

where $C_R[m, n]$, $C_G[m, n]$, $C_B[m, n]$ are the RGB values of the pixel $[m, n]$ and $A_R(i)$, $A_G(i)$, $A_B(i)$ are the average RGB values in Region i.

Note that, with our algorithm, in addition to the pixels in the small regions should be re-processed in Step 7, almost all the pixels only have to be processed one time. This is of great help for improving the efficiency. Moreover, due to the adjustable region mean $A(i)$ and the merging operation in the Case 4 of Step 5, our method can have good performance.

Fig. 3: (a) Gray-level Lena image (b) Color Lena image

With the proposed process, an image can be segmented very efficiently. The result is as good as that of region growing method but the running time is much less.

Figs. 4-8 are the simulations for 256×256 Lena image. We will show the gray-level simulation results of using the region growing, K-means, watershed, and our proposed methods. Moreover, we will show an extra result of color image segmentation for Lena image using our method.

In Fig. 4, we can see the result of K-means. We assign the algorithms with clustering number N = 18 at first. But we just show four different sections here. As we can see, there are 4 different sections in the result and the processing time is good with about 1.36 seconds. However, we disappointed with the countless fragmented sections of Lena's hair. The result is meaningless for us for that there are many sections should be classified as the same section by human perception, i.e. hair.

Next we see the results of the region growing method in Fig. 5 The result of region growing is very good especially compared with the result of K-means in the part of hair. However, we have to realize that it costs us about 136 seconds which is 2 more minutes to get the perfect result. As we mentioned in previous section, the result of region growing is very reliable which is match the result here. The only thing we have to consider about of using the region growing method is the speed of it. We would not like to spend so much time on segmentation stage. Therefore, we think the region growing method is not suitable.

Fig. 6 shows the results of watershed method. The first result (a) is the watershed segmentation without using any improved method. As we can see, the advantage of its fast speed (processing time: 1.22 seconds) is no more important under the serious over-segmented problem which we emphasized in the previous context.

Even we improve the result by using the gradient method with watershed segmenting method in (b) (c), we can see the problem cannot be solved, either.

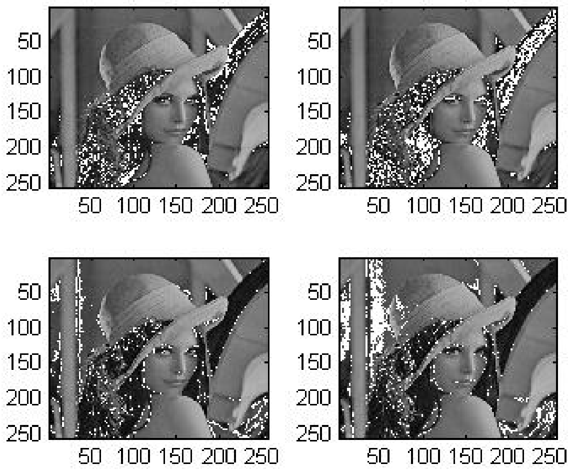
Fig. 4: The segmentation results of the **K-means** method (running time = **1.36 seconds**.)

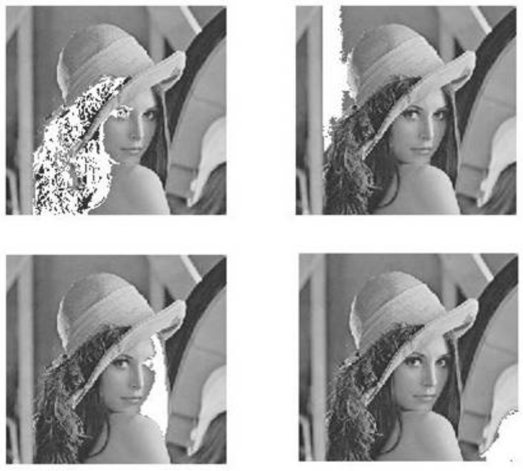
Fig. 5: The segmentation results of the **Region growing** method with threshold = 0.1 (running time = **136.48 seconds**.)

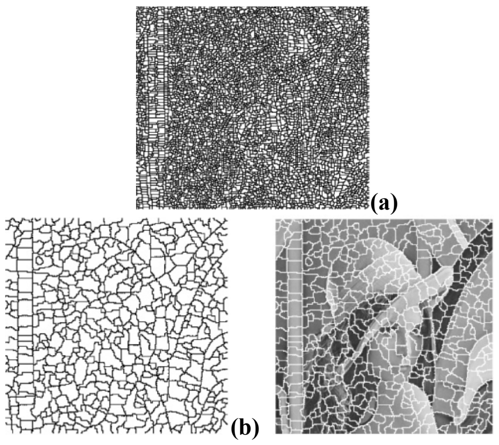

Fig. 6: The segmentation results of the **Watershed** algorithm (running time = **1.23 seconds**.) (a) Pure watershed method, (b)(c) watershed method with improvement of gradient method

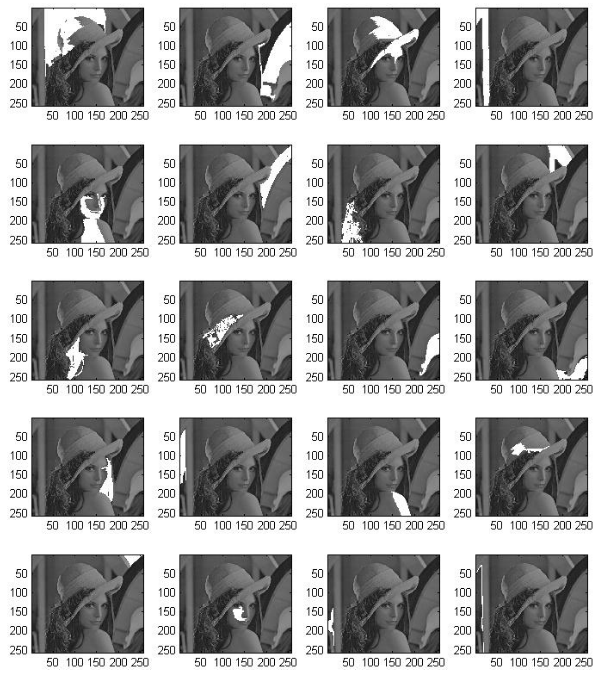
Fig. 7: The segmentation results of the **Proposed** method (running time = **1.13 seconds**.)

The last simulation result is our method. It shows in Fig. 7. As we can see, the shapes of clusters in the result of our method are similar to the result of region growing. That is because we use the region-based merging algorithm in our method. By this kind of algorithm, we can easily avoid the phenomenon shows in K-means method and watershed method. Mention the hair part of the result. In our method, we classify the section of Lena's hair into two main parts. In the region growing method, we can classify it almost into one single part. Though the result is not so perfect comparing with region growing, as we can see, the processing time is very short.

Furthermore, we can still improve the result by improve the selection of our threshold. This part of improvement we will introduce in the next section. Besides, we attach a result of processing color Lena image by using our method in Fig. 8.

The region growing method has good segmentation results, but the speed is slow (The required time is about 120 times longer than that of our method). The K-means and the Watershed methods are fast, but the performances of segmentation need to be improved. By contrast, our method can achieve the goals of **high efficiency** and **better performance** at the same time.

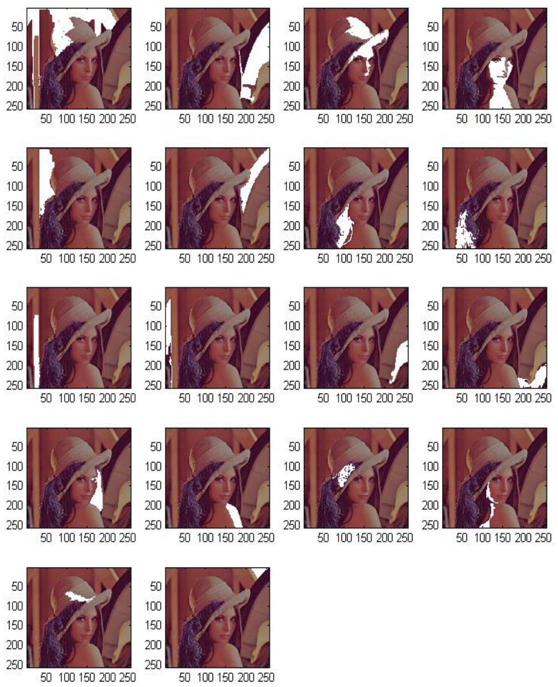
Fig. 8: The segmentation results of our **Proposed** method for color Lena image (running time = 1.63 seconds)

We list a comparison of our algorithm to another three algorithms in Table 2. From the results described above, our method has obvious predominance in **speed**, especially when compared with the region growing method.

The second term which is the result shape, our method still takes a big advantage on the K-means and the watershed method. **Our result shape will be intact, but not fragmentary or over-segmented**. And that is the biggest advantage of the region growing method.

The third is the system reliability. Because of the over-segmentation phenomenon, **our method will be better than watershed in the system reliability**. For our goal, our method is also better than K-means method because the result of k-means is fragmentary and compress the segmenting result will waste too much resource for us to record the boundaries. Our method takes advantage on the k-means and the watershed methods in the system reliability term. However, we have to say our reliability is worse than the region growing method. As we can see the simulation result in the previous context in Section 3, the result of region growing method is indeed better than ours, if it can successfully runs on any image.

For this, we would like to improve our algorithm. We would like to improve the reliable of our system and we try to improve it from the decision of the threshold.

We use an interesting idea of adaptive threshold decision by local variance and frequency. We will describe it in detail in next section.

Table 2: Comparison of algorithms

	Region growing	**K-means**	**Watershed**
speed	Our method is much faster	Our method is a little faster	Our method is a little faster
Shape connectivity	Performances are similar	Our method: intact K-means: fragmentary	Our method: intact Watershed: over-segmentation
Shape match	Performances are similar	Performances are similar	Our method > Watershed

4. ADAPTIVE THRESHOLD

In our algorithm, the threshold showed in section 3 is not changed in the whole algorithm running procedure. That is, the whole image is using the same global threshold. That makes some restrictions because the variance and frequency of some parts of the original figure is different from another. We would like to improve our method from this point. We would like to make a new procedure that could adaptively decide the threshold with the local frequency and variance in the original figure.

We have a simple idea. First, we would like to select the threshold based on the local variance and frequency of a figure. We hope the adaptive decision of threshold could improve the efficiency of the result using our method.

Here is the step of the algorithm：

1. Separate the original figure to 4*4, 16 sections
2. Compute the variances and frequencies of the 16 sections, respectively
3. Depending on the local variances and frequencies, we select the suitable threshold.

To sum up, we conclude four situations like Fig. 9 for our improvement.

We will assign the larger value of threshold to the area which fits first case than the one fitting the second case than the third one, and we will assign the smallest value of threshold to the area fitting the fourth case.

For the first case, we would like to assign a larger value of threshold to the figure area which is with high frequency and high variance. The reason for a higher threshold we select is that there are often many edges and different objects in this kind of area. The larger value of threshold may cause a rough segmenting result, but we believe the clear edge and the high variety between different objects will make the segmenting work. The larger threshold will remove some over-segmentation cause by the high variance and high frequency.

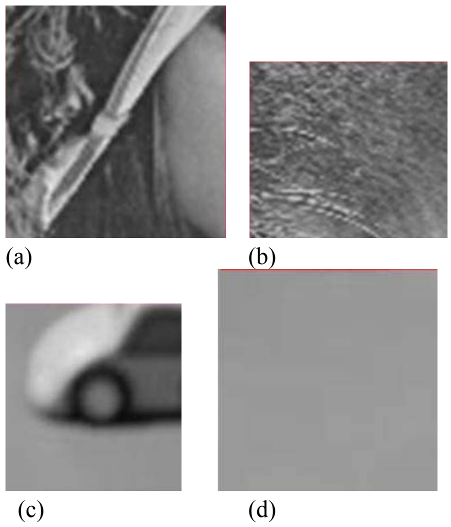

Fig. 9: (a) high frequency and high variance, (b) high frequency and low variance, (c) low frequency and high variance and (d) low frequency and low variance

It is the same reason why we select the second high threshold value for the second case. In the third case and the last case, we will assign them the smaller value of threshold. It might be thought that the smallest threshold in case four will cause an over-segment result. However, the stable and monotonous characteristic in case four will not make the over-segmentation work. For all the example showed below, we use a formula to decide the threshold：

$$threshold = 16 + F + V \tag{16}$$

The formula of F：

$$F = A \cdot \text{ (local average frequency) } + B \tag{17}$$

The formula of V：

$$V = C \cdot \text{ (local variance) } + D \tag{18}$$

In this paper we always try to control the threshold value between 16 and 32 for the best testing threshold value with the original method (without using adaptive threshold) will be 24. For that, the range of F will be 0 to 8, and so does the range of V. The maximum of F and V are all 8, which make the maximum of threshold be 32.

```
If local average frequency>9,
        F=6;
Else if local average frequency<3,
        F=0;
End
If local variance>3000,
        V=6;
Else if local variance<1000,
        V=0;
End
```

Let us check a simulation result as below: The local variance of Lena.bmp：

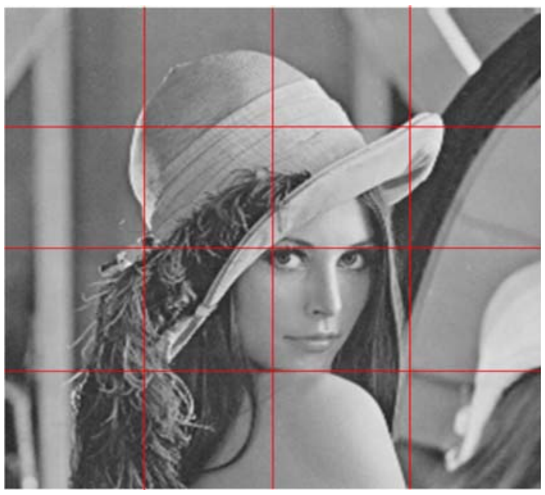

Fig. 10: Lena.bmp

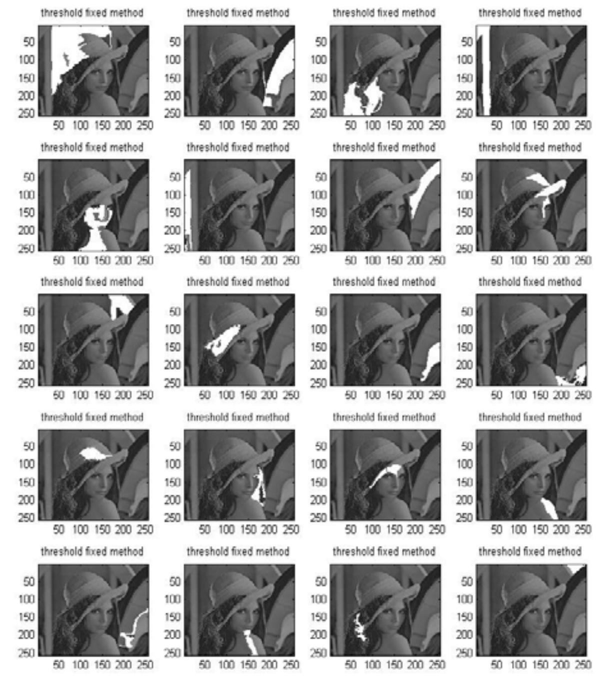

Fig. 11: The result of segmentation with adaptive threshold in (16) for Lena image

$$\begin{matrix} 716 & 447 & 1293 & 2470 \\ 899 & 1579 & 1960 & 2238 \\ 1497 & 1822 & 1974 & 1273 \\ 2314 & 1129 & 1545 & 1646 \end{matrix} \tag{19}$$

The local average frequencies of Lena.bmp are

$$\begin{matrix} 6.9451 & 8.8807 & 7.2965 & 7.0914 \\ 8.0413 & 10.0076 & 8.4951 & 7.6421 \\ 9.6709 & 10.3219 & 7.9815 & 6.1310 \\ 9.0464 & 10.4821 & 7.1513 & 6.4118 \end{matrix} \tag{20}$$

For the Lena.bmp as in Fig. 10, we hope the new algorithm will make the hair section all combine together because the hair section is belong to the case 1. We list the result in Fig. 11.

Compare Fig. 11 with Fig. 7, we find that the results of the segmentation with adaptive threshold can separate the hair region from the feather of the hat. That is exactly the goal we want to achieve.

5. CONCLUSIONS

We discuss the three essential characteristics when we define the image segmenting algorithm as the front-stage processing of image compression which is the **fast speed,** the **good shape connectivity** of segmenting result, and the g**ood shape matching**. We also discuss three candidate popular algorithms for this kind of role.

However, none of the three algorithms have these three characteristics at the same time.

Therefore, we propose a useful segmenting algorithm. It is fast with the speed. It has good shape connectivity with its segmenting results. It also has a not bad shape matching.

Moreover, in order to improve the shape matching of the algorithm, we develop a adaptive threshold selection method base on the local variance and frequency.

By checking out the simulation results in the last of the few chapters, we can conclude that our proposed method does take some advantages on the three traditional image segmenting methods we discuss in the front of Section 3. For the compression-oriented image segmenting algorithm, we can say our method is successful so far.

ACKNOWLEDGEMENT

The authors would like to thank the members of the NTU thesis defense committee for their constructive and valuable comments and suggestions. They also thank the support of the project 98-2918-I-002-012 by National Science Council of Taiwan.

REFENRENCES

[1] R. C. Gonzalez, R. E. Woods, *Digital Image Processing 2nd Edition*, Prentice Hall, New Jersey 2002.

[2] M. Petrou and P. Bosdogianni, *Image Processing the Fundamentals*, Wiley, UK, 2004.

[3] W. K. Pratt, *Digital Image Processing 4nd Edition*,John Wiley & Sons, Inc., Los Altos, California, 2007.

[4] J. Tilton, “Analysis of Hierarchically Related Image Segmentations”, presented at the *IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data*, Greenbelt, MD, 2003.

[5] S.C. Satapathy, J. V. R. Murthy, R. Prasada, B.N.V.S.S, R. Prasad, P.V.G.D. “A Comparative Analysis of Unsupervised K-Means, PSO and Self-Organizing PSO for Image Clustering”, *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications*, 2007. vol. 2, pp.: 229-237, Dec. 2007,