***Segmentation***

Chia-Hao Tsai

E-mail: r98942062@ntu.edu.tw

Graduate Institute of Communication Engineering

National Taiwan University, Taipei, Taiwan, ROC

Yu-Hsiang Wang

E-mail: r98942059@ntu.edu.tw

Graduate Institute of Communication Engineering

National Taiwan University, Taipei, Taiwan, ROC

**Abstract**

Image segmentation is the front-stage processing of image compression. We hope that there are three advantages in image segmentation. The first is the speed. The second is good shape connectivity of its segmenting result. The third is good shape matching. Besides, we introduce many segmenting methods including threshold technique, data clustering, region growing, region merging and splitting, mean shift, and watershed. At the same time, we also compare advantages and advantages. Because of some disadvantages of them, the author creates fast scanning algorithm to improve those disadvantages and use an adaptive threshold decision to improve the efficiency of fast scanning algorithm he created [1].

1. **Introduction**

It has many issues to handle in digital image processing including image segmentation, image compression, and image recognition…etc. We will introduce image segmentation here.

Image segmentation is the front-stage processing of image compression. In general, we hope that there are three advantages in image segmentation. The first is the speed. When segmenting an image, we do not want speed much time to do it. The second is good shape connectivity of its segmenting result. When segmenting an image, we do not want the result of segmenting shape to be fragmentary. If the result of segmenting shape is fragmentary, we need take many resources to record the boundaries of the over-segment results. It is not we want to get the results. The third is good shape matching. Consequently, it will be reliable.

Image segmentation can be classified three categories traditionally including Threshold Technique, Region-Based Image Segmentation, and Edge-Based Image Segmentation. We will introduce Threshold Technique, Region-Based Image Segmentation, and Edge-Based Image Segmentation in following chapters.

1. **Threshold Technique**

Thethreshold technique is simplest in segmenting methods. To set two thresholds on the histogram of the image, we can classify between the two thresholds in the histogram as the same region and classify the others as the second region.

* 1. **Multi-level thresholding through a statistical recursive algorithm**

Multilevel thresholding for image segmentation through a statistical recursive algorithm is proposed in [9]. The algorithm is used in segmenting an image into multi-level by using mean and variance. The method can be made use of dealing with colored images or images of complex background, and then can do what bi-level doesn’t it.

Multi-level thresholding algorithm:

1. Repeat steps 2~6, *n*/2-1 times; where *n* is the number of thresholds.

2. Range ***R*** = [*a*, *b*]; initially set a = 0 and b = 255.

3. Find mean () and standard deviation () of all the pixels in ***R***.

4. Sub-ranges’ boundaries  and  are calculated as  and; where  and  are free parameters.

5. Pixels with intensity values in the interval [*a*, ] and [, *b*] are assigned threshold values equal to the respective weighted means of their values.

6. , .

7. Finally, repeat step 5 with  and with.

Using the algorithm can the compute the PSNR (peak signal to noise ratio). After applying the algorithm a few times, we can find the PSNR to be saturated. By the property, we can get the appropriate number of thresholds *n.*

1. **Region-base Image Segmentation**
   1. **Data clustering**

Data clustering is one method of Region-Based image segmentation, and it is popularly used mathematics and statistics. We can use the centroids or prototypes to present the great numbers of cluster to achieve the two goals of reducing the computational time consuming and providing a better condition to compress it.

In general, data clustering can be classified two kinds of system including hierarchical clustering and partitional clustering. In the hierarchical clustering, we can change the numbers of cluster during the process. However, in the partitional clustering, we must decide the numbers of cluster before processing.

* + 1. **Hierarchical clustering**

For the hierarchical clustering, it has an advantage of simple concept. It is roughly classified two kinds of algorithms including hierarchical agglomerative algorithm and hierarchical divisive algorithm.

Hierarchical agglomerative algorithm:

1. Let every single data point (pixel or image) in the whole image as a cluster.
2. Look for the shortest distance of two data pointin the whole image, and merge them to become a new cluster.
3. Repeat the step 1 and step 2 until the numbers of cluster attain our demand.

We can use many ways to define the distance here.

Hierarchical divisive algorithm:

1. Let the whole image as a cluster.
2. Look for the biggest diameter of the cluster groups.
3. If , split  out as a new clusterand see the rest data points of  as .
4. If, split  out as .
5. Back to step 2 and continue the algorithm until  and  is not changed anymore.

The diameter of a cluster  as The diameter is defined as .

the mean of distance between  and every single point in cluster .

Using the method of hierarchical clustering, the result is characteristic of strong correlation with the original image. Therefore, it will be reliable. Nevertheless, it has a fatal defect of computational time consuming, then it cannot be used for the large image.

* + 1. **Partitional clustering**

In the partitional clustering, we must decide the numbers of cluster before processing. The K-means algorithm is most well-known in the partitional clustering.

K-means algorithm:

1. Decide the numbers of the cluster  and choose randomly  data points ( pixels or image) in the whole image as the  centroids in  clusters.
2. Find out nearest centroid of every single data point (pixel or image) and classify the data point into that cluster the centroid located. After doing step 2, all data points are classified in some cluster.
3. Calculate the centroid of every cluster.
4. Repeat step 2 and step 3 until it is not changed.

Using the K-means algorithm, it has an advantage of less computing time. In other words, the partitional clustering is faster than the hierarchical clustering. However, the different initial centroids will bring about the different results which means the K-means algorithm has an initial problem. In order to solve the initial problem, we can choose to use one initial point or use the Particle Swarm Optimization (PSO) [2].

* 1. **Region growing**

Region growing is simplest in region-base image segmentationmethods [3]. The concept of region growing algorithm is check the neighboring pixels of the initial seed points, then determine whether those neighboring pixels are added to the seed points or not. Therefore, it is an iterative process.

Region growing algorithm:

1. Choose the seed points.
2. If the neighboring pixels of the initial seed points are satisfy the criteria such as threshold, they will be grown. The threshold can be intensity, gray level texture, and color…etc.

We use the criteria of the same pixel value in Fig. 3.1, then check the neighboring pixels of the initial seed points. If their pixel values are identical with seed points, they can be added to the seed points. It is stop until there is no change in two successive iterations. We use 4-connected neighborhood to grow the neighboring pixels of the initial seed points here.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | 9 | 9 | 9 |
| 1 | 1 | 9 | 9 | 9 |
| 5 | 1 | 1 | 9 | 9 |
| 5 | 5 | 5 | 3 | 9 |
| 3 | 3 | 3 | 3 | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | 9 | 9 | 9 |
| 1 | 1 | 9 | 9 | 9 |
| 5 | 1 | 1 | 9 | 9 |
| 5 | 5 | 5 | 3 | 9 |
| 3 | 3 | 3 | 3 | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | 9 | 9 | 9 |
| 1 | 1 | 9 | 9 | 9 |
| 5 | 1 | 1 | 9 | 9 |
| 5 | 5 | 5 | 3 | 9 |
| 3 | 3 | 3 | 3 | 3 |

(a) original image (b) step 1 (c) step2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | 9 | 9 | 9 |
| 1 | 1 | 9 | 9 | 9 |
| 5 | 1 | 1 | 9 | 9 |
| 5 | 5 | 5 | 3 | 9 |
| 3 | 3 | 3 | 3 | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | 9 | 9 | 9 |
| 1 | 1 | 9 | 9 | 9 |
| 5 | 1 | 1 | 9 | 9 |
| 5 | 5 | 5 | 3 | 9 |
| 3 | 3 | 3 | 3 | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | 9 | 9 | 9 |
| 1 | 1 | 9 | 9 | 9 |
| 5 | 1 | 1 | 9 | 9 |
| 5 | 5 | 5 | 3 | 9 |
| 3 | 3 | 3 | 3 | 3 |

(d) step 3 (e) step 4 (f) step5

Fig. 3.1 An example of region growing.

* 1. **Region merging and splitting**

Region merging and splitting is a developing algorithm in segmenting the images [4]. It is used to differentiate the homogeneity of the image.

Region merging and splittingalgorithm:

1. Splitting step:

We choose the criteria to split the image based on quad tree. At the same time, we can determine the numbers of splitting levels gradually.

1. Merging step:

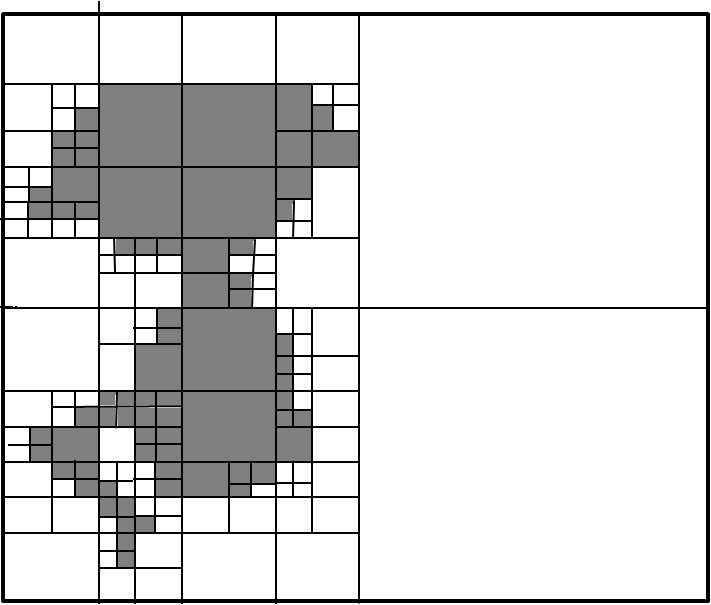
If the adjacent regions satisfy the similarity properties, we will merge them.

1. Repeat step 2 until it is not changed.

In Fig. 3.2, it is an example of region merging and splittingalgorithm. We use the splitting criteria and the merging criteria of the locating total area of one section. We split the image until get the resolution we need. Fig. 3.2 (a), (b), (c) and (d) show the splitting part and Fig. 3.2 (e) and (f) show the merging part.

splt_merge_robot3splt_merge_robot2splt_merge_robot1

(a) Original Image (b) Splitting: stage 1 (c) Splitting: stage 4

splt_merge_robot4splt_merge_robot4

(d) Splitting: stage 5 (e) Merging: stage 5 (f) Merging result

Fig. 3.2 The example of region merging and splitting.

The quad tree-based segmentation has the problem of the blocky segmentation as DCT image compression.

* 1. **Mean Shift**

Numerous nonparametric clustering methods can be classified into two large classes: hierarchical clustering and density estimation. Hierarchical clustering techniques either aggregate or divide the data based on some proximity measure. They tend to be computationally expensive and not straightforward. Differently the density estimation is regarded as the empirical probability density function (p.d.f) of the represented parameter.

The mean shift can be classified into density estimation. The mean shift adequately analyse feature space to cluster them and can provide reliable solutions for many vision tasks. Then we describe the mean shift procedure in the following:

The Mean Shift Procedure:

Given n data points **x**i, i=1,… , *n* in the d-dimensional space *Rd* and set one bandwidth parameter h > 0. The mean shift is

, (3.1)

where kernel *k*(p) is

(3.2)

when **m***h,k*(**x**) is smaller than a threshold, that means convergence then we can stop calculate mean shift. But if **m***h,k*(**x**) is bigger than threshold, we should set **m***h,k*(**x**)’s first term be the new mean and repeat computing **m***h,k*(**x**) until convergence.

Mean shiftalgorithm:

1. Decide what features you want mean shift to consider and you should let every features be a vector. Then we could construct *d* dimensions matrix. For example,

dataPts= (3.3)

1. Randomly select a column to be an initial mean. For example,

 (3.4)

1. Construct a matrix, which is the repeat of an initial mean and use this matrix to minus “dataPts”. Then calculate the square of every components of the new matrix and individually sum every column to get a vector “SqDistToAll”. For example,

SqDistToAll==

 (3.5)

Sum every column

1. Find out the positions, which their value are smaller than (bandwidth)2 from “SqDistToAll”. Store these positions in “inInds” and label these positions in “beenVisitedFlag”.
2. Recompute the new mean among the value of “inInds”.
3. Repeat step3 ~ step5 until the mean is convergence. The convergence means the distance between previous mean and present mean is smaller than the threshold that we decide. Distance represents their mean square or the sum of their difference’s square.
4. After convergence, we can cluster those labeled positions in the same cluster. But before clustering, we have to examine whether the distance between the new found mean and those old means is too close. If it happens, we should merge those labeled positions into the old mean’s cluster.
5. Afterward eliminate those clustered data from “dataPts” and repeat step2 ~ step7 until all of “dataPts” are clustered. Then the mean shift’s clustering is finished.
   1. **Simulations of K-means algorithm, Region growing algorithm, and Mean shift algorithm in image segmentation**



Fig. 3.3 K-means clustering/time with MATLAB code (gray level image)：18 clusterings/1.36 seconds.

The simulation result of K-means algorithm is countless fragmented sections, as Fig. 3.3. Many sections should be grouped as the same section by human perception, so it is useless for us.

The simulation result of region growing algorithm is better than the simulation results of K-means algorithm. But, it will spend more time to simulate the result.

The simulation result of mean shift algorithm wastes too much time. If bandwidth is smaller, it takes longer for simulation. And it has an advantage that it can separate the face and shoulders. However, it cannot separate the other regions which is the other algorithms can separate.

**   **

**   **

**   **

**   **

**   **

**   **

Fig. 3.4 Region growing with threshold= 5 by using C++ code (gray level image); time：17.06 seconds.



Fig. 3.5 Mean shift with bandwidth= 60 by using MATLAB code (gray level image); time= 130 sec.



Fig. 3.6 Mean shift with bandwidth= 50 by using MATLAB code (gray level image); time= 726 sec.

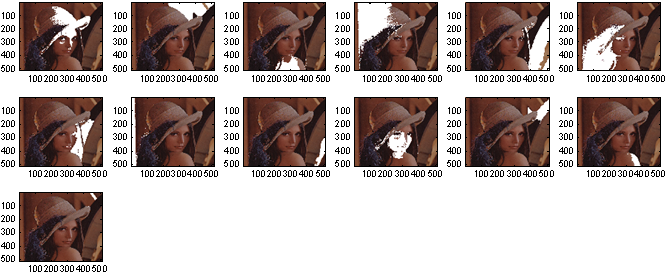


Fig. 3.7 Mean shift with reducing the position information into half and bandwidth= 50 by using MATLAB code (colored image); time= 660 sec.

1. **Edge detection**

Edge detection and corner detection discuss recently in digital image processing. Image segmentation can be regard as progress of edge detection. The watershed image segmentation is an example of edge-based image segmentation.

* 1. **Point detection and line detection**

The action of the point detection is used to detect the difference between a single pixel and the adjacent pixel.

3x3 Point detection mask:

 (4.1)

The action of the line detection resembles the point detection. It is used to detect the lines in an image.

|  |  |
| --- | --- |
| 3x3 Line detection mask for 0°: | 3x3 Line detection mask for 45°: |
| 3x3 Line detection mask for 90°: | 3x3 Line detection mask for 135°: |

Table 4.1 3x3 Line detection masks for four orientation directions.

* 1. **Edge detection**

In general, we use the derivative method to detect the edge. Nevertheless, the derivative method is very sensitive to noise. First-order derivative and second-order derivative methods are the two techniques of implementation of the derivative method. The first-order derivative is computed the gradient in an image. The second-order derivative is computed the Laplacian in an image. The second-order derivative is usually more sensitive than the first-order derivative.

* + 1. **derivative method by gradient operators**

To find the magnitude and direction of the edge, we can define the gradient vector  as below.

 (4.2)

The magnitude of the gradient vector :

 (4.3)

The direction of the gradient vector :

 (4.4)

In Table 4.2, it is the seven gradient edge detectors. In Fig. 3.1, we show that use the seven gradient edge detectors and choose proper thresholds to get the binary edge images.

|  |  |
| --- | --- |
| Roberts operator:    Gradient magnitude:  where are values from first, second masks respectively. | Prewitt edge detector:    Gradient magnitude:  Gradient direction:  where are values from first, second masks respectively. |
| Sobel edge detector:    Gradient magnitude:  Gradient direction:  where are values from first, second masks respectively. | Frei and Chen edge detector:    Gradient magnitude:  Gradient direction:  where are values from first, second masks respectively. |
| Kirsch edge detector:      Gradient magnitude:  Gradient direction:  where are values from first, second,…, eighth masks respectively. | |
| Robinson edge detector:      Gradient magnitude:  Gradient direction:  where are values from first, second,…, eighth masks respectively. | |
| Nevatia and Babu edge detector:        Gradient magnitude:  Gradient direction:  where are values from first, second,…, sixth masks respectively. | |

Table 4.2 The seven gradient edge detectors.

|  |  |
| --- | --- |
| Roberts operator with threshold=12  cv9_1.bmp | Prewitt edge detector with threshold=24  cv9_2.bmp |
| Sobel edge detector with threshold=38  cv9_3.bmp | Frei and Chen gradient operator with threshold=30  cv9_4.bmp |
| Kirsch compass operator with threshold =135  cv9_5.bmp | Robinson compass operator with threshold=43  cv9_6.bmp |
| Nevatia-Babu 5X5 operator with threshold=12500  cv9_7.bmp |  |

Fig. 4.1 To use the seven gradient edge detectors and choose proper thresholds to get the binary edge images.

* + 1. **derivative method by Laplacian operators**

In Table 4.3, it is the three Laplacian operators. In Fig. 3.2, we show that use the three Laplacian edge operators and choose proper thresholds to get the binary edge images.

|  |
| --- |
| Laplacian operator: |
| Minimum-variance Laplacian operator: |
| Laplacian of Gaussian (LoG) operator for 5x5 mask (Mexican hat function): |

Table 4.3 The three Laplacian operators.

|  |  |
| --- | --- |
| Laplacian of mask= withthreshold=20  cv10_11.bmp | Laplacian of mask= withthreshold=20  cv10_12.bmp |
| Minimum-variance Laplacian with threshold=15  cv10_2.bmp | Laplacian of Gaussian with threshold=5000 (kernel size=11)  cv10_3.bmp |
| Difference of Gaussian with threshold=2000 (inhibitory  excitatory  kernel size=11)  cv10_4.bmp |  |

Fig. 4.2 Use the four Laplacian edge detectors and choose proper thresholds to get the binary edge images.

The Laplacian of Gaussian (LoG) operator is also called Mexican hat function. It can achieve two goals. The first is using Gaussian function can decrease the noise influence to smoothen the images. The second is using the Laplacian operator will produce zero-crossing that can use to detect the edges. However, the drawback of derivative method is sensitive to noise, and use the LoG operator can solve the problem. Furthermore, because the difference between every single pixel of a continuous ramp edge is not so obvious in an image, another disadvantage of derivative method is not sensitive to ramp edges.

* 1. **Edge-Based Image Segmentation**

**Watershed image segmentation:**

Watershed image segmentation can be regarded as an image in three dimensions (two spatial coordinates versus intensity). We will use three types of point which “minimum”, “catchment basin”, and “watershed line” to express a topographic interpretation. There are two properties of continuous boundaries and over-segmentation in watershed image segmentation. Because watershed image segmentation has the disadvantage of over-segmentation, we use the maker to improve it.

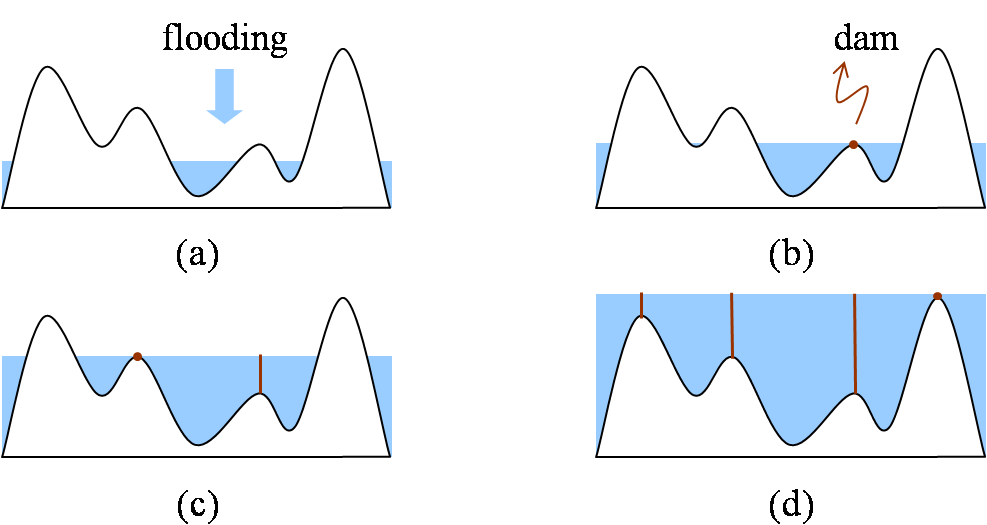
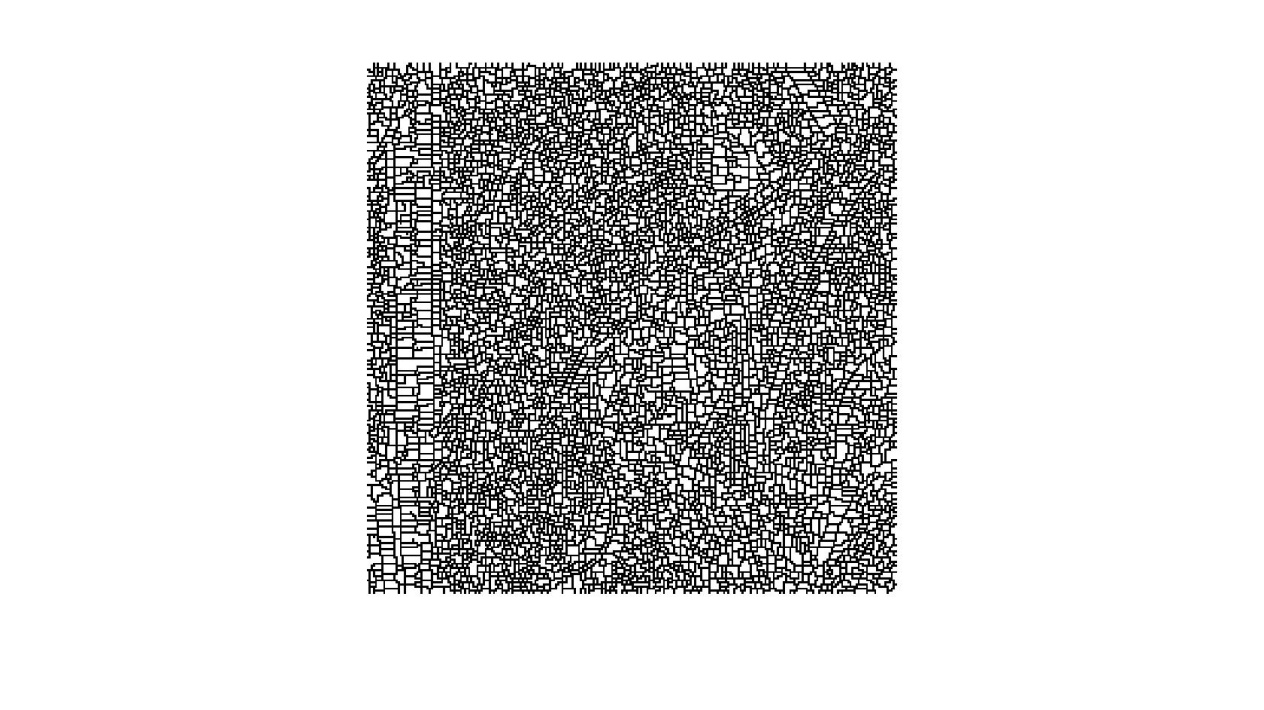
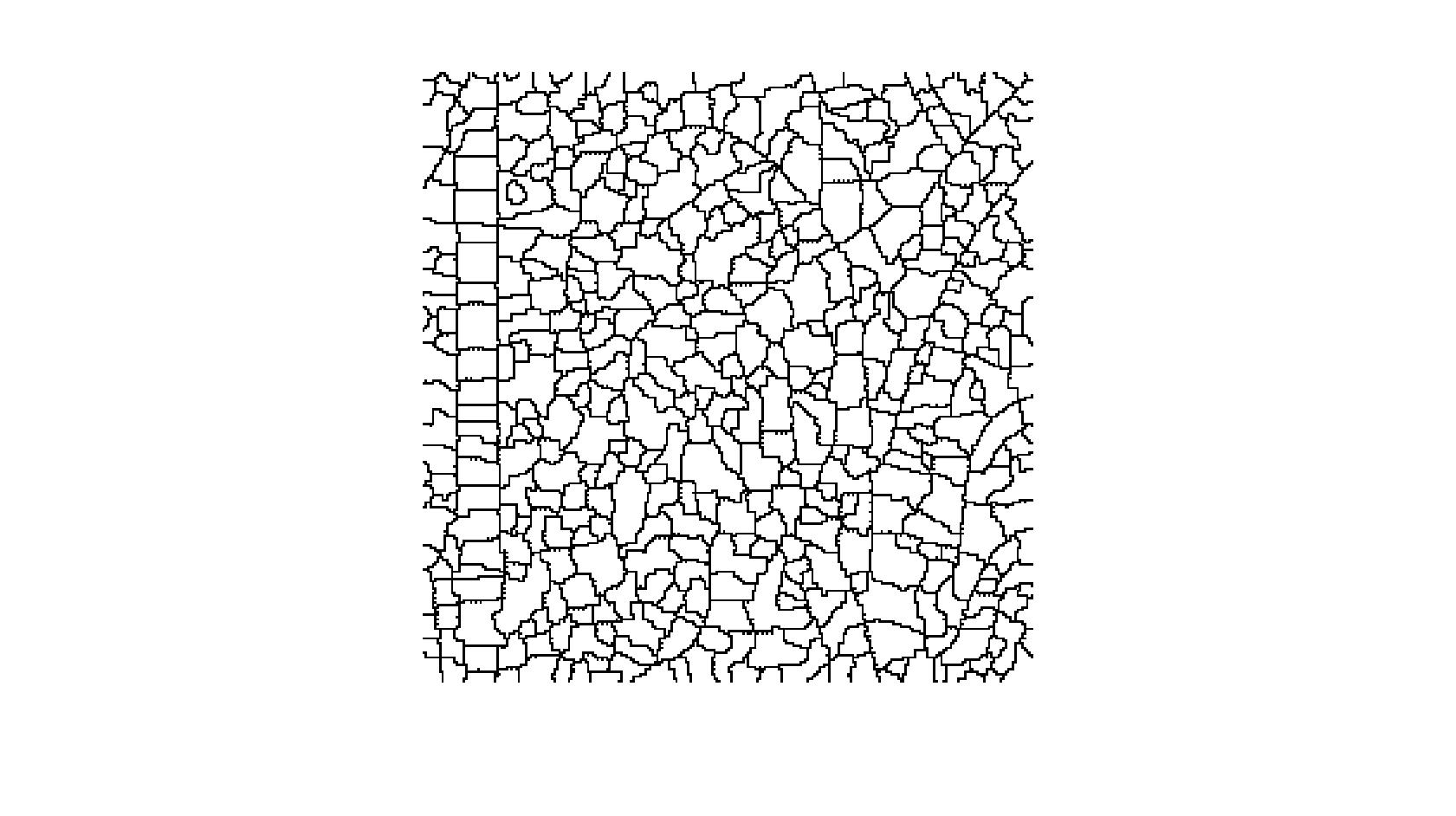
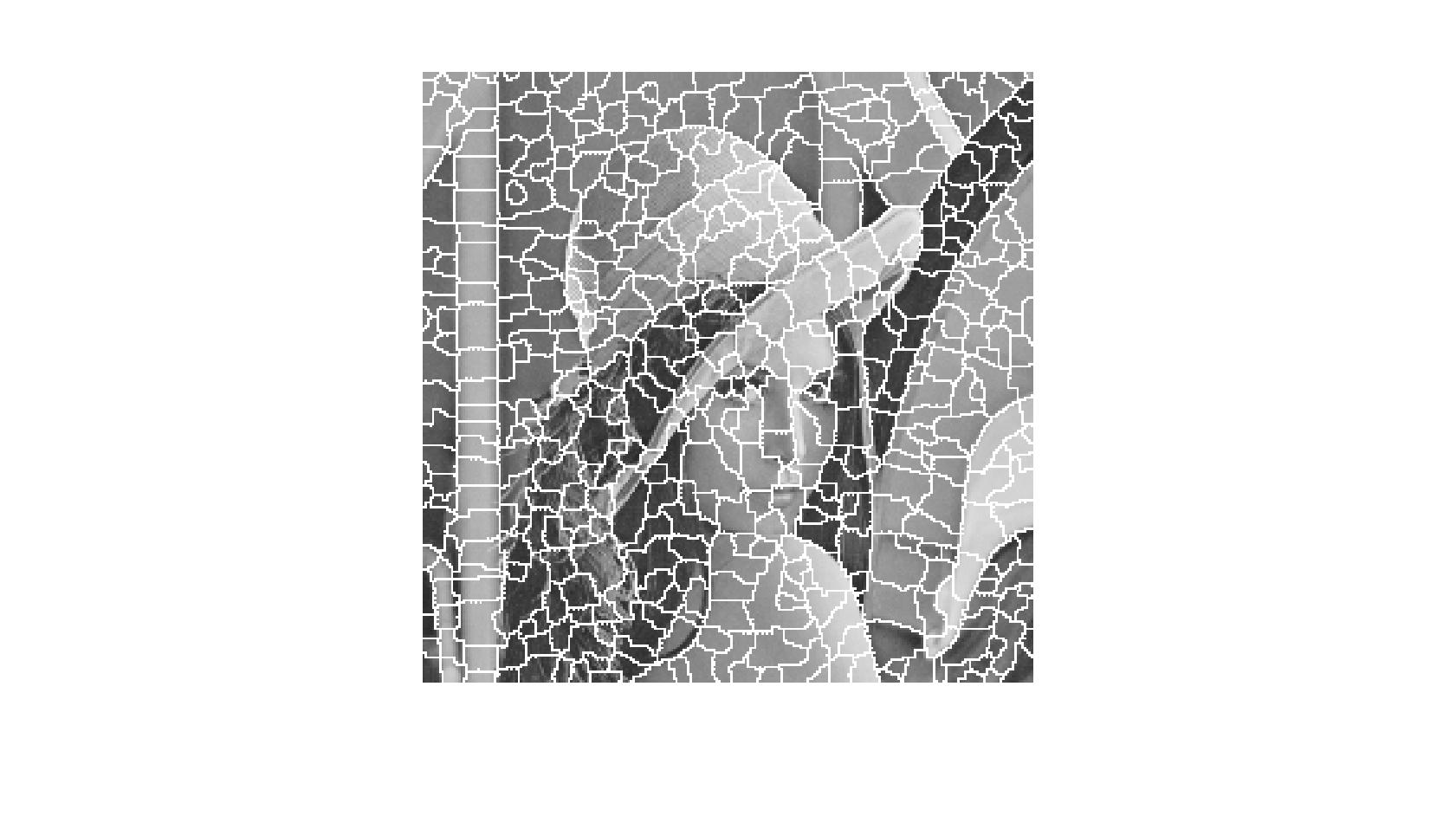


Fig. 4.3 Watershed algorithm [7].

Watershed algorithm with using marker:

1. Use a smoothing filter to preprocess the original image, then the action can minimize the large numbers of small spatial details.
2. Use two markers (internal markers and the external markers) to define the criteria of markers.
3.  (b)



(c)

Fig. 4.4 The simulation result of Watershed algorithm with MATLAB code; time: 1.23 seconds (a) pure watershed method, (b)(c) watershed method with improvement of gradient method.

The simulation result of watershed algorithm has an advantage that it is fast speed. At the same time, it has a critical over-segmented problem.

* 1. **The comparison of threshold technique and methods of region-based image segmentation and edge-based image segmentation**

|  |  |
| --- | --- |
| **The segmenting methods** | **advantages** |
| **Threshold technique** | 1. Simplest method in segmenting images. |
| **\*Hierarchical clustering** | 1. The concept is simple. 2. The result is characteristic of strong correlation with the original image. (reliable) |
| **\*Partitional clustering**  **(K-means algorithm)** | 1. Fast speed. 2. The concept is simple, because numbers of cluster is fixed. |
| **\*Region growing** | 1. Can correctly separate the regions of same properties we define. 2. Clear edges, which means the good segmentation results. 3. The concept is simple. 4. Good shape matching of its results. 5. Can determine seed points and criteria 6. Can choose the multiple criteria simultaneously. |
| **\*Region merging and splitting** | 1. We split the image until get the resolution we need. 2. The splitting criteria and the merging criteria can use different criteria. |
| **\*Mean shift** | 1. Can separate the face and shoulders. |
| **△Watershed** | 1. Fast speed. 2. The large numbers of segmented region result is reliable. |

Table 4.4 The advantages of threshold technique and methods of region-based image segmentation and edge-based image segmentation.

|  |  |
| --- | --- |
| **The segmenting methods** | **disadvantages** |
| **Threshold technique** | 1. Not involve the spatial information of the images, so it will bring about noise, blurred edges, or outlier in the images. |
| **\*Hierarchical clustering** | 1. Has a problem of computational time consuming, then it cannot be used for the large image. |
| **\*Partitional clustering**  **(K-means algorithm)** | 1. A problem of choice of numbers of cluster . 2. The different initial centroids will bring about the different results. 3. Cannot show the characteristic of database. |
| **\*Region growing** | 1. Has a problem of computational time consuming 2. Cannot differentiate the fine variation of the images. |
| **\*Region merging and splitting** | 1. Computation is extensive. 2. Has the problem of the blocky segmentation. |
| **\*Mean shift** | 1. Has a problem of computational time consuming  2. Cannot separate the other sections except the face and shoulders. |
| **△Watershed** | 1. Over-segmentation. |

Table 4.5 The disadvantages of threshold technique and methods of region-based image segmentation and edge-based image segmentation.

\*: means one method of region-based image segmentation.

△: means one method of edge-based image segmentation.

1. **A problem discussion for non-closed boundary segmentation**

**5.1 A problem discussion for non-closed boundary segmentation**

We can use texture feature or boundary shape to represent a region. Fourier descriptor is most widespread one of methods of representing boundary shape. The effect of the Fourier descriptor is that coverts the boundary segment into frequency domain. Afterward we can truncate the high-frequency component to achieve the goal of compression. But, there are some problems for the Fourier descriptor.

The variable *R* is indicated the ratio of the number of compressed term P to the number of original term *K*, it means compression rate. The high-frequency component expresses for fine detail, and the low-frequency represents global shape in Fourier transform theorem. When *P* is smaller, the lost detail is more on the boundary.

Specially, when the compression rate *R* is below 20%, the corner of the boundary shape will be smoothed for the Fourier descriptor and the reconstruction result is not very good. That is a big problem. The reason of the big problem is the corner or boundary usually accompanies the high-frequency component. When we get rid of the high-frequency component to achieve the goal of compression, the corner or boundary will be sacrificed in the original boundary simultaneously.

**5.2 Asymmetric Fourier descriptor of non-closed boundary segment**

The method is proposed and called “Asymmetric Fourier descriptor of non-closed boundary segment” that can improve the problem we mentioned above [6]. It has four steps for the method. We will introduce the four steps as below.

1. Predict and mark the corner point in boundary.
2. Segment the boundary shape into several non-closed boundaries. Nevertheless, if we get rid of the high-frequency in a non-closed boundary, the reconstructed boundary will be a closed boundary. That is a serious problem. We solve the problem as below.
3. Record the coordinates of two end points of the boundary segment.
4. Base on the distance between two end points, the boundary points are shifted linearly.
5. Add an odd-symmetric boundary segment to make the boundary segment to be closed and continuous perfectly between two end points.
6. Calculate the Fourier descriptor of the new boundary segment.

Fig. 5.1 Several steps in solving the non-closed boundary segments problem

1. Truncate the high-frequency component to achieve the goal of compression.
2. Do the boundary segment encoding and decoding in Fig 4.2 and Fig 4.3.

Eventually, the method of “asymmetric Fourier descriptor of non-closed boundary segment” can solve the problem we mentioned above even when compression rate *R* is below 20% and upgrade the efficiency in the compression system.



Fig. 5.2 Boundary segment encoding.



Fig. 5.3 Boundary segment decoding.

1. **Fast scanning**
   1. **Fast scanning**

This method uses the concept of merge to scan the whole image (from top-left to down-right) and determine which cluster the pixel is proper to join. The determined method we used here is focus on pixel value and defined a threshold as a merged criterion. Afterward we will add two more concerned factors, the local variance and local average frequency to strengthen our method’s tenacity.

Fast scanning algorithm:

1. In the beginning, we decide the threshold whether the pixel can be merged into the cluster or not and set the top-left pixel (1, 1) as the first cluster called cluster *C1*. The pixel we are scanning called *Cj* and this pixel’s left side pixel called *Ci* (*Ci* is decided by which the left pixel’s cluster is).
2. From the first row, we scan the next pixel (1,1+1) and compare it with left pixel (1,1)’s cluster. The mathematical formula is

If *Cj* – *centroid*(*Ci*) ≦ threshold, we merge *Cj* into *Ci* and recalculate the centroid of *Ci*.

If *Cj* – *centroid*(*Ci*) ≧ threshold, we set *Cj* as a new cluster *Ci+1*.

1. Repeat step 2 until all of the pixel in first row have been scan.
2. Scan the next row and compare pixel (x+1, 1) with its upper side cluster *Cu*. Make a decision whether we can merge pixel (x+1, 1) into *Cu*.

If *Cj* – *centroid*(*Cu*) ≦ threshold, we merge *Cj* into *Cu* and recalculate the centroid of *Cu*.

If *Cj* – *centroid*(*Cu*) ≧ threshold, we set *Cj* as a new cluster *Cn*, where n is the cluster number so far.

1. Scan this row’s next pixel (x+1, 1+1) and compare it with the region *Cu*, *Ci* which is upper to it and left side of it, respectively. Make a decision whether we can merge pixel (x+1, 1+1) into *Cu* or *Cj*.

If *Cj* – *centroid*(*Cu*) ≦ threshold and *Cj* – *centroid*(*Ci*) ≦ threshold,

1. Combine the region *Cu* and *Ci* to be region *Cn*, where n is the cluster number so far.
2. Merge *Cj* into *Cn*.
3. Recalculate the centroid of *Cn*.

Else if *Cj* – *centroid*(*Cu*) ≦ threshold and *Cj* – *centroid*(*Ci*) > threshold,

Merge *Cj* into *Cu* and recalculate the centroid of *Cu*.

Else if *Cj* – *centroid*(*Cu*) > threshold and *Cj* – *centroid*(*Ci*) ≦ threshold,

Merge *Cj* into *Ci* and recalculate the centroid of *Ci*.

Else

Set *Cj* as a new cluster *Cn*, where n is the cluster number so far.

1. Repeat step 4 ~ step 5 until the whole image has been scanned.
2. To find out the small region from the clusters in the step 1 ~ step 6. For the 256x256 input images, the small region defined as the size below 32. Set those small region as *Ri*, where *i* = 1 ~ *k*.
3. Scan *Ri*, start from the top-left pixel of *Ri* called *Ri*(p), p is the number of *Ri*. Compare it with the region *Cu*, *Ci* which is upper to it and left side of it, respectively.

If |*Ri*(p) –*centroid*(*Cu*)| < |*Ri*(p) –*centroid*(*Ci*)|, we merge *Ri*(p) into *Cu* and recalculate the centroid of *Cu*.

If |*Ri*(p) –*centroid*(*Cu*)| > |*Ri*(p) –*centroid*(*Ci*)|, we merge *Ri*(p) into *Ci* and recalculate the centroid of *Ci*.

1. Repeat step (a) until all the pixels of *Ri* have been scanned.
2. Repeat step (a) ~ step (b) until all the small regions have been merged into bigger cluster.



Fig. 6.1 The simulation result of Fast scanning algorithm by using MATLAB code.

* 1. **The Improvement of Fast scanning with adaptive threshold decision**

In our primitive method, the threshold of the whole image is all the same. However, the different parts of image have the different color distribution or the different variance and frequency. So we propose the concept of adaptive threshold decision dependent on local variance and local frequency.

The Improvement of Fast scanning algorithm with adaptive threshold decision:

1. Separate the origin image to 4\*4, 16 sections.
2. Compute the local variance and local frequency of the 16 sections, respectively.
3. According to the local variance and local frequency, compute the suitable threshold. The method is below.



Fig. 6.2 Lena image separated into 16 sections

To summarize, we classify four situations for the improvement.

1. High frequency, high variance



Fig. 6.3 Figure of high frequency and high variance

1. High frequency, low variance



Fig. 6.4 Figure of high frequency and low variance

1. Low frequency, high variance



Fig. 6.5 Figure of low frequency and high variance

1. Low frequency, low variance



Fig. 6.6 Figure of low frequency and low variance

From the four situations, we assign the largest value of threshold to the figure with high frequency and high variance. Although the larger value of threshold will cause a rougher segmentation, the clear edge and the variety between different objects will make the segmentation successfully. So the larger value of threshold will avoid some over-segmentation cause by high frequency and high variance.

Then we reduce the threshold in order for case 2 ~ case 4. The smallest value of threshold may cause over-segment result generally. But the low frequency and low variance region’s character is monotonous, so case 4 can endure the smallest threshold and not make the over-segment work.

For example, defined a formula for threshold:

Threshold = 16 + *F* + *V*  (6.1)

The formula of F:

*F* = *A* × (local average frequency) + *B*  (6.2)

The formula of F:

*V* = *C* × (local variance) + *D* (6.3)

In this thesis, we wants control the threshold value between 16 and 32, so the range of *F* will be 0 to 8 and so does the range of *V*. The control procedure is below.

If local average frequency > 9

*F* = 8; (6.4)

else if local average frequency < 9

*F*=0; (6.5)

end

If local variance > 3000

*V* = 8; (6.6)

else if local variance < 1000

*V* = 0; (6.7)

end

So the value of *A*, *B*, *C*, *D* will be 4/3, -4, 0.004, -4, respectively.

We can also change the value of *A*, *B*, *C*, *D* to change the range of the final threshold. The following equation can achieve:

[*A*, *B*] = solve(‘*A*=(*F*min – *B*)/3’,’*B*=*F*max – 9\**A*’); (6.8)

[*C*, *D*] = solve(‘*C*=(*V*min – *D*)/1000’,’*V*max – 3000\**C*’); (6.9)



Fig. 6.4 The simulation result of Fast scanning algorithm withadaptive threshold decision by using MATLAB code.

1. **Conclusion**

We want to make a better environment to compress after we segment it, so hope that there are three advantages in image segmentation. The first is the speed. The second is good shape connectivity of its segmenting result. The third is good shape matching. Moreover, data clustering, region growing, and region merging and splitting do not have these three characteristics at the same time, so the author creates fast scanning algorithm to improve those disadvantages [1]. In the end, the author uses adaptive threshold decision by local variance and frequency to improve his algorithm [1].

**Reference**

[1] C.J. Kuo, Fast Image Segmentation and Boundary Description Techniques, M.S. thesis, National Taiwan Univ., Taipei, Taiwan, R.O.C, 2009.

[2] S. Satapathy, JVR. Murthy, B. Rao, P. Reddy, “A Comparative Analysis of Unsupervised K-Means, PSO and Self-Organizing PSO for Image Clustering”, International Conference on Computational Intelligence and Multimedia Applications 2007.

[3] S. W. Zucker, "Region Growing: Childhood and Adolescence," Computer Vision, Graphics, and lmage Processing, vol. 5, pp. 382-389, Sep. 1976.

[4] Xiang, R. and Wang, R., 2004. Range image segmentation based on split-merge clustering. In: 17th ICPR, pp. 614–617.

[5] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Vol. I, Addison Wesley, Reading, MA, 1992.

[6] J. J. Ding, J. D. Huang, C. J. Kuo, W. F. Wang, “Asymmetric Fourier descriptor of non-closed boundary segment,” *CVGIP*, 2008.

[7] J. Huang, “Image Compression by Segmentation and Boundary Description,” M.S. thesis, National Taiwan University, Taipei, Taiwan, 2008.

[8] L. Lucchese and S.K. Mitra, “Color Image Segmentation: A State-of-the-Art Survey”.

[9] S. Arora, J. Acharya, A. Verma, Prasanta K. Panigrahi, “Multilevel thresholding for image segmentation through a fast statistical recursive algorithm,” Pattern Recognition Letters 29, pp. 119–125, 2008.

[10] J. J. Ding, S. C. Pei, J. D. Huang, G. C. Guo, Y. C. Lin, N. C. Shen, and Y. S. Zhang, “Short response Hilbert transform for edge detection,” *CVGIP*, 2007.

[11] S. C. Pei; J. J. Ding, "Improved Harris’ algorithm for corner and edge detection", *Proc.* *IEEE ICIP*, vol.3, On page(s): III - 57-III – 60, Sept. 2007

[12] D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach toward Feature Space Analysis,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, pp. 603-619, 2002.