Automatic Separation of Overlapping Objects*

Liu Binghan, Fang Xiuduan, Wang Weizhi
Dept. of Computer Science and Technology
Fuzhou Universtiy
Fuzhou 350002
China

Zheng Zhiyong
Fuzhou General Hospital
Nanjing Military Area
Fuzhou 350025
China

Abstract **–This paper presents an algorithm based on mathematical morphology to separate overlapping objects automatically, which locates their linking points by distance transform and dilation. It develops the design and implementation of the approach. In particular, it applies the technique to the problem of separating liver cells in hepatitis pathology images. The experiments show that the technique is fast and the separated results are satisfactory.**

Keywords – Image processing Overlapping objects Distance transform Dilation

I. INTRODUCTION

In image automatic quantitative analysis, it is crucial to obtain quantitative parameters. However, due to overlapping objects in images, it is hard to analyze quantitatively those parameters related to the number of objects. To resolve the problem, a separating algorithm based on mathematical morphology is presented. It consists of the following two steps. First, to find respective cores of overlapping objects, it executes distance transform to areas including them. Second, to realize automatic separation, it applies the dilation technique to dilate outward from their cores and so finds their linking points. The experimental results demonstrate the approach is fairly ideal.

II. IMAGE AUTOMATIC SEGEMENTATION

To separate overlapping objects in an image, it is necessary to segment it so as to distinguish its objects and background. For an image with simple background, namely, with only one kind of objects, it is suitable to apply the optimized algorithm called maximal classificaticatory square error[1], which determines its single threshold automatically according to the principle to maximize classificatory square error between the two classes. For an image with complex background, a hybrid algorithm based on the idea of Ohlander's cycle segmentation[2] is proposed, which transforms complex multi-threshold segmentation into several single threshold segmentation and each time decides a single threshold according to the maximal classificatory square error algorithm. The approach of Ohlander's cycle segmentation first divides an image into several sub-images according to its global histogram. Then it repeatedly makes further segmentation according to the sub-image histograms until they can't be segmented. Since each time it executes segmentation on continuously refreshed sub-images, it reflects the image's local features in more and more detail as the number of cycle increases. At last it can acquire fine segmentation. Take a HE dyed hepatitis pathology image as an example, the original image and its resulting image after automatic segmentation are respectively shown in figure 7(a) and 7(b). In figure 7(b), blue areas denote object areas.

After an image has been segmented, there always exist noises, burrs and holes in object areas , which could be eleminated with the techniques of smoothing, open operation[4] and close operation[4] etc.

* Supported by Fujian Natural Science Foundation (A9910006)

III. AUTOMATIC SEPARATION OF OVERLAPPING OBJECTS

After image segmentation, some object areas may have only a single object, while others may have several overlapping objects. The former are relatively regular in shape and small in area whereas the latter are relatively large and irregular. Provided overlapping parts are relatively small, each object generally has its own core, the linking areas among them are depressed in shape and the linking distance is locally minimal. According to the above-mentioned modal features, to excute distance transform in object areas is an ideal approach to locate their cores. Then, dilate outward from the cores. As a result, the joins after dilation are the linking points of these objects.

A. Distance Transform

In this paper, the distance between two pixels is determined by the neighborhood including four pixels. Therefore, the distance between any two pixels, $f(i,j)$ and $f(m,n)$, is given by:

$$d(f(i,j)\,,f(m,n)\,)=|\,i-m\,|+|\,j-n\,| \tag{1}$$

Distance Transform[3] is to replace the value of a pixel in an object area with the least distance value between it and the area's edge. The transform is executed with a serial method[3] based on neighborhood. To complete the transform, it is necessary to scan the area twice. The first is to scan sequentially, that is , from left to right and from up to down. Let $f(i,j)$ be an original pixel value, without losing universality, let

$$f(i,j)=\begin{cases}1, & \text{if } f(i,j) \text{ is in an object area}\\ 0, & \text{others}\end{cases} \tag{2}$$

When performing the distance transform to some pixel (i,j), since there are only two pixels having finished the transform amongst its four-pixel neighborhood, the transform has to be operated based on local neighborhood. Let $s(i,j)$ be the result of the first scan., $s(i,j)$ is given by

$$s(i,j)=\begin{cases}0, & f(i,j)=0\\ min\{\,(s(i,j\text{-}1)+1),\,(s(i\text{-}1,j)+1)\,\}\,, & f(i,j)=1\end{cases} \tag{3}$$

In figure 1 and figure 2, the original images and their corresponding results after the first scan are shown. Because the first scan only takes local neighborhood into account, the distance values in the right-down area aren't the least to the edge. So scan the area again in inverse order, namely, from right to left and from bottom to top. After the second scan, the final result of the distance transform is obtained. Let $d(i,j)$ be the operation result of the second scan, $d(i,j)$ is defined as follows

$$d(i,j)=\begin{cases}0, & s(i,j)=0\\ min\{\,(d(i+1,j)+1),\; (d(i,j+1)+1),\,s(i,j)\,\}\,, & s(i,j)\neq 0\end{cases} \tag{4}$$

In figure 3, the final distance images are demonstrated.

In a distance image, the distance value of a pixel directly reflects its distance to the edge of the area. Thus, finding the greatest distance value can locate the area's cores. The equation is given by

$$d(i,j)=\begin{cases}d(i,j), & d(i,j)\geqslant max\{d(i\text{-}1,j\text{-}1),d(i,j\text{-}1),\\ & \quad d(i+1,j\text{-}1),\,d(i,j\text{-}1),\\ & \quad d(i,j+1),d(i\text{-}1,j+1),\\ & \quad d(i,j+1),d(i+1,j+1)\}\\ 1\,, & \text{others}\end{cases} \tag{5}$$

In figure 4, the results of core location are shown. It is obvious that the core pixels compose the core areas, which are respectively denoted as k, $k=2,3,4\ldots$, as the figure 5 shows.

B. Dilation

Dilation[4] is to expand boundary points of linking parts of specified objects outward for one circle.

Let $f(i,j)$ be original pixel and $g(i,j)$ be the result of dilation, then in the order of sequential raster, $g(i,j)$ is given by

$$g(i,j)=\begin{cases} 1, & f(i,j)=1 \\ 1, & f(i,j)=0 \,\&\&\, f(i-1,j-1)+f(i-1,j)+ f(i-1,j+1)+f(i,j-1)+ f(i,j+1)+f(i+1,j-1)+ f(i+1,j)+f(i+1,j+1) \geqslant 1 \\ 0, & \text{others} \end{cases} \quad (6)$$

To separate overlapping objects automatically, it is necessary to dilate core areas for several times until original object areas are all restored and at the same time they can't be altered. During dilation, the joins are marked with the number "-1". The dilation is executed as the following equation *(1)*.

$$G_k(i,j)=\begin{cases} k, & d(i,j)=k \\ k, & d(i,,j)=1 \text{ AND } (d(i-1,j-1)=k \text{ OR } d(i-1,j)=k \text{ OR } d(i-1,j+1)=k \text{ OR } d(i,j-1)=k \text{ OR } d(i,j+1)=k) \text{ OR } d(i+1,j-1)=k \text{ OR } d(i+1,j)=k \text{ OR } d(i+1,j+1)=k) \\ -1, & d(i,j)\neq k \text{ AND } d(i,j)>1 \text{ AND } (d(i-1,j-1)=k \text{ OR } d(i-1,j)=k \text{ OR } d(i-1,j+1)=k \text{ OR } d(i,j-1)=k \text{ OR } d(i,j+1)=k \text{ OR } d(i+1,j-1)=k \text{ OR } d(i+1,j)=k \text{ OR } d(i+1,j+1)=k) \\ d(i,j), & \text{others} \quad k=2,3,4..... \end{cases} \quad (7)$$

C. The algorithm of automatic separation

In the algorithm of automatic separation, it is required to obtain the parameter of the number of objects, labeled *Num*. The value of *Num* is related to two modal parameters of a single object, that is, the parameter of area, labeled *Pa*, and the parameter of roundness degree, labeled *Pr*, which are acquired by training and learning specified object images. The algorithm of automatic separation of overlapping objects is depicted as follows:

① Scan object areas one by one in the resulting image of automatic segmentation and get the following three parameters of the corresponding modal features, namely, area parameter, labeled *AREA*, circumference parameter, labeled *L*, and roundness degree, labeled *R*, where *R* is given by the equation $R=(4\pi\times AREA)/L^2$. If the scan has been completed, then stop.

② Calculate *Num* according to the formula $Num = AREA / Pa$. If ($R < Pr$) and ($Num > 0$), then continue, else go to step ①;

③ Execute distance transform to the corresponding object area. Locate the core areas and denotes them with the number k, k=2,3,.... If (k-1 <= *Num*), then $Num = k-1$, else select only *Num* core areas and denotes them 2,3,... *Num*+1 and set the values of object pixels not in core areas equal to 1;

④ Dilate repeatedly the respective Num core areas according to the equation (7). If the current dilation hasn't altered any pixel value of the area, then set a mark denoting that the dilation operation on it has been completed.

⑤ If the total *Num* core areas have been dilated, then go to step ①, else go to step ④.

After the algorithm of automatic separation is executed, the points with pixel value –1 are the object linking points, as the figure 6 shows, where "*" denotes "-1".

```
000000000000000000
000000001110000000
000000011111000000
000000111111100000
000000111111100000
000000011111000000
000000001110000000
000000011111000000
000000111111100000
000000111111100000
000000011111000000
000000001110000000
000000000000000000
```

(a) two objects in series

```
000000000000000000
000000000000000000
000000000000000000
000110001100011000
001111011110111100
011111111111111110
011111111111111110
001111011110111100
000110001100011000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
```

(b) three objects in series

```
000000000000000000
000001110000000000
000011111000000000
000111111000000000
000111111001110000
000011111011111000
000000111111111000
000111111111111000
000111111011110000
000111111001100000
000011111000000000
000001110000000000
000000000000000000
```

(c) three objects in parallel

Fig.1 original iamges

```
000000000000000000
000000001110000000
000000012221000000
000000123332100000
000000123443200000
000000012344000000
000000001230000000
000000012341000000
000000123452100000
000000123563200000
000000012344000000
000000001230000000
000000000000000000
```

```
000000000000000000
000000000000000000
000000000000000000
000110001100011000
001221012210122100
012332123321233210
012343234432344320
001234012340123400
000120001200012000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
```

```
000000000000000000
000001110000000000
000012221000000000
000123332000000000
000123443001110000
000012344012221000
000000123123332000
000112342344443000
000122345012340000
000123456001200000
000012345000000000
000001230000000000
000000000000000000
```

Fig.2 distance values after sequential scan

```
000000000000000000
000000001110000000
000000012221000000
000000123332100000
000000123432100000
000000012321000000
000000001210000000
000000012321000000
000000123432100000
000000123332100000
000000012221000000
000000001110000000
000000000000000000
```

```
000000000000000000
000000000000000000
000000000000000000
000110001100011000
001221012210122100
012332123321233210
012332123321233210
001221012210122100
000110001100011000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
```

```
000000000000000000
000001110000000000
000012221000000000
000123321000000000
000122321001110000
000011221012221000
000000122123321000
000011232123321000
000122321012210000
000123321001100000
000012221000000000
000001110000000000
000000000000000000
```

Fig.3 distance values after the second scan

```
000000000000000000
000000001110000000
000000011111000000
000000111111100000
000000111411100000
000000011111000000
000000001110000000
000000011111000000
000000111411100000
000000111111100000
000000011111000000
000000001110000000
000000000000000000
```

```
000000000000000000
000000000000000000
000000000000000000
000110001100011000
001111011110111100
011331113311133110
011331113311133110
001111011110111100
000110001100011000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
```

```
000000000000000000
000001110000000000
000011111000000000
000113311000000000
000111311001110000
000011111011111000
000000111113311000
000011131113311000
000111311011110000
000113311001100000
000011111000000000
000001110000000000
000000000000000000
```

Fig. 4 the resulting images after cores location

```
000000000000000000
000000001110000000
000000011111000000
000000111111100000
000000111211100000
000000011111000000
000000001110000000
000000011111000000
000000111311100000
000000111111100000
000000011111000000
000000001110000000
000000000000000000
```

```
000000000000000000
000000000000000000
000000000000000000
000110001100011000
001111011110111100
011221113311144110
011221113311144110
001111011110111100
000110001100011000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
```

```
000000000000000000
000001110000000000
000011111000000000
000112211000000000
000111211001110000
000011111011111000
000000111113311000
000011141113311000
000111411011110000
000114411001100000
000011111000000000
000001110000000000
000000000000000000
```

Fig.5 the resulting images after marking core areas

```
000000000000000000
000000002220000000
000000022222000000
000000222222200000
000000222222200000
000000022222000000
00000000***0000000
000000033333000000
000000333333300000
000000333333300000
000000033333000000
000000003330000000
000000000000000000
```

```
000000000000000000
000000000000000000
000000000000000000
000220003300044000
002222033330444400
022222*3333*444440
022222*3333*444440
002222033330444400
000220003300044000
000000000000000000
000000000000000000
000000000000000000
000000000000000000
```

```
000000000000000000
000002220000000000
000022222000000000
000222222000000000
000222222003330000
000022222033333000
000000****33333000
000044444*33333000
000444444033330000
000444444003300000
000044444000000000
000004440000000000
000000000000000000
```

Fig.6 the resulting images after separating objects

IV. EXPERIMENTAL RESULT

We have taken the HE dyed hepatitis pathology images under 200 time view as examples and collected a large number of images as training samples. The upper limitation of the cell area *Pa* is specified as 200 pixels and the roundness degree *Pr* is 0.7 by analysis and calculation. Figure 7 demonstrates the resulting image after executing the automatic separation algorithm.

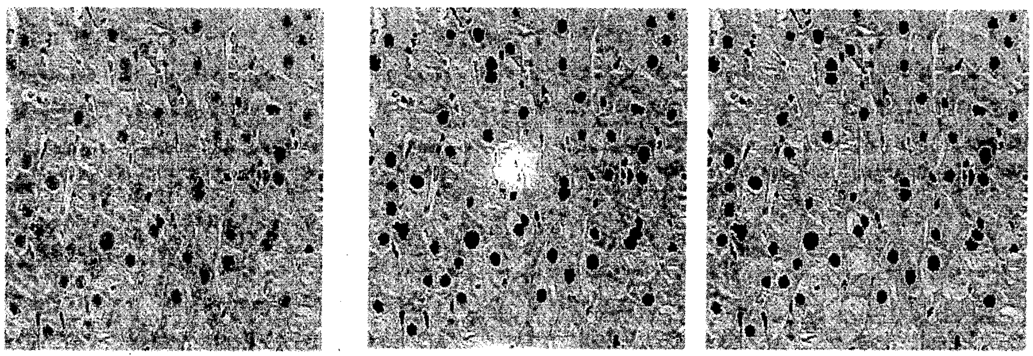

Fig.7 (a) original image (b) the resulting image of automatic segmentation (c) the resulting image of cell separation

As the experiments show, the algorithm is fast. It finds the respective core of each object by scanning twice. What's more, dilation is executed simultaneously from each core. So it can restore the original object areas as quickly as possible and locate the linking points correctly. However, the algorithm isn't fit for the problem that objects overlap a large area because under the circumstance the cores of objects overlap each other. Only if the cores of objects don't overlap, the automatic separation algorithm can get rather fine result.

V. REFERENCES

[1] Wang Jifen, Zhang Xinrong, *Computer Image Recognition*, China Railway Press, Bejing, 1988, 85~89.

[2] Wang Runsheng, *Image Understanding*, Publishing House of National Defence Science and Technology University, Changsha, 1994, 93~94.

[3] Fang Ruming, Cai Jianrong, Xuli, *The Technology of Computer Image Processing and its Application to Agriculture Engineering*, Tsinghua University Press, 1999, 77~82.

[4] Tang Changqing, *Mathmatical Morphorlogy and its Application*, Science Press, 1900.