Contour Extraction and Processing Approaches for Glioma Cell

Zhou Zhen  
Biomedical Engineering  
Capital Medical University  
Beijing, China  
zhouzhenbme@gmail.com

Tong Long Zheng  
Biomedical Engineering  
Capital Medical University  
Beijing, China  
tlzeng@cpums.edu.cn

Mabin Rong  
Biomedical Engineering  
Capital Medical University  
Beijing, China  
mabinrong@gmail.com

Abstract—Some problems about glioma cell contour extraction and contour processing were discussed. In the process, regional segmentation and chain-code tracing were used for color image directly, while gray level pretreatment was avoided. Thus the color information of the slice was fully stored and utilized. By using seed technique in regional segmentation, cell contours were separated from background as much as possible. By using bit parallel processing of logic operation in chain-code tracing, parallel processing of multiple sample points were implemented, which resulted in the increase of processing speed.

Keywords—Contour tracing; seed growing; chain-code; parallel processing

I. INTRODUCTION

In the process of classifying the slice of glioma pathologic, cell perimeter, area, degree of rotundity collection and color characteristic parameter extraction are very important [1,2]. We used to study the slice of glioma pathologic after making them micrograph bivalue. By bivalue, the processing speed can be increased, but a lot of slice information is lost, which makes the subsequent recognition and classification difficult.

This study made a full use of the color difference between cells and background of the micrograph, then employed the seed-growing algorithm to extract the cells, and finally, according to the markers, achieved the cell contour extraction and preservation by chain-code tracing.

II. BASIC CONCEPTION

A. Seeds growing

The algorithm means that scanning circumjacent pixels of the seeds and regarding the circumjacent pixels, which is below a threshold, as new seeds, repeating this process till no pixels can be regarded as seeds.

B. Vector of bitmap data

X_{nm} is a matrix. The elements of the 1\textsuperscript{st} row and the J\textsuperscript{th} col of X_{nm} can be denoted as vectors.

\[
X[I]=(X[I,0], X[I,1], X[I,2], \ldots, X[I,M-1]); \quad X[J]=(X[0,J], X[1,J], X[2,J], \ldots, X[N-1,J]).
\]

X_{0}[I], X_{c}[I], X_{u}[J], X_{d}[J] are separately defined as matrix X[I,J] shifting right, left, up, down one pixel. Namely:

\[
X_{0}[I]=(0,X[I,0],X[I,1],X[I,2],\ldots,X[I,M-2]); \quad X_{c}[I]=(X[I,1],X[I,2],X[I,3],\ldots,X[I,M-1],0);
\]

\[
X_{u}[J]=(X[1,J],X[2,J],X[3,J],\ldots,X[N-1,J],0); \quad X_{d}[J]=(X[0,J],X[1,J],X[2,J],\ldots,X[N-2,J]);
\]

The pixels of edge are filled with 0.

The cell contour is defined as Q. For example, Q (X, Y) is the circumjacent pixels’ value of pixel (X, Y); P (X, Y) is the value of pixel (X, Y). If there are background pixels in circumjacent of (X, Y), the pixel (X, Y) is defined as contour pixel.

\[
Q_8(X,Y)=P(X,Y) \text{ AND}\{P(X-1,Y-1) \text{ OR } P(X+1,Y-1) \text{ OR } P(X-1,Y) \text{ OR } P(X+1,Y) \text{ OR } P(X-1,Y+1) \text{ OR } P(X+1,Y+1) \}\]

In order to save time, only four circumjacent pixels are calculated. This predigesting doesn’t affect the result of collecting contours.

\[
Q_4(X,Y)=P(X,Y) \text{ AND}\{P(X,Y-1) \text{ OR } P(X+1,Y-1) \text{ OR } P(X-1,Y+1) \text{ OR } P(X+1,Y+1) \}
\]
The direction from one pixel to its adjacent pixel is one of the eight directions shown in Figure 1.

Chain-code algorithm is employed to record the contour. This approach not only can denote direction to the next pixel in edge list, but also easily track each pixel of the cell contour. The definition of chain-code value is according to the directions. The value of chain-code increase 1, its direction will anticlockwise circumrotate 45°. The location of each cell contour pixel is tracked by decoding chain-code. Decoding chain-code is based on the value of chain-code. Decoding is a reverse process of the coding process. Acquirement of the next cell contour pixel depends on the previous cell contour pixel. It can avoid scanning all of the cell pixels, the subsequent process speed is increased and many memories can be saved. Table 1 denotes the relation between the value of chain-code and coordinate of pixels. So the coordinates of cell contour can restore based on decoding chain-code.

<table>
<thead>
<tr>
<th>Chain-code</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift of X</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Shift of Y</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

III. CELL SEGMENTATION AND CONTOUR EXTRACTION

In the slice of glioma pathologic, the color of cell and background is different. Different kinds of glioma pathologic show different colors; even the same kind of glioma pathologic may show different colors, because dyeing deepness is different. Reference color of the cell was picked from the slice. Based on reference color, the criterion was defined by the difference between the pixel’s color and reference color. Threshold was adjusted till cells and coordinate was segmented [3]. In virtue of the characters of cell and texture of background, glioma was easily recognized and analyzed.

A complex process of recognition was divided into two processes, extracting cells of tumor and analyzing background texture [4].

A. Cell Extracting

Because of color segmentation, noise inevitably exists in the images. Median filter can restrain noise and keep the edge clear-cut. Median filter was employed to eliminate noise.

By Comparison, “regional growing” can give satisfying result. First, we chose cells with evident character, and marked them. Second, any one of the pixels in region of cells can be regarded as the seed, then grow from these seeds.

First, the eight circumjacent pixels round the seeds will be scanned. If the circumjacent pixel doesn’t belong to other regions and is below a threshold, this pixel should be regarded as cell pixel and was added into seeds region and thus formed new cells region A1. Then eight circumjacent pixels of each seed in A1 will be scanned. The pixels, which below a threshold, will be added into A1, and then form new seeds region A2. We repeat this process till no seeds can be collected, namely the growing process ended.

TABLE I. THE RELATION OF CHAIN-CODE VALUE AND COORDINATES

<table>
<thead>
<tr>
<th>Chain-code</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift of X</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Shift of Y</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

Figure 2 is a seed growing illustration. In order to illustrate this process, we chose one vector of R, G, B vectors. The seed was 9; the color threshold of the seed was 2. If there was color difference between the candidate pixel and the seed, the pixel was regarded as seed. After first growing, F(j-1,k), F(j,k-1), F(j,k+1) were collected form seeds region (b). The average seeds value F1=8.25. After the second growing, (b) caught circumjacent pixels F(j+2,k), F(j,k+1), and formed new seeds region (c). The average seeds value F2=7.38. After third growing, (c) caught F(j+1,k+1) and formed new seeds region (d). The average seeds value F3=6.29. Then, no pixels could be collected at the fourth growing, and the growing process ended.

In the practical process, R, G, B was calculated together. If all the R, G, B vectors of a pixel were below a color threshold, the pixel could be collected in seed region. By employing the recursion algorithm in VC++6.0, the result was satisfactory.

B. Tracking cell contours

We appended a mark bit in every byte when image data were saved in array. If the mark was “1”, it indicated that it was the pixel of cell. If the mark was “0”, it indicated it was the pixel of background. The algorithms of extracting contour bases on mark bit.

In order to increase the processing speed, parallel processing [5] was employed, but not line-by-line scanning. It is known from above definition that the contour can be recorded with the chain-code. In order to farther increase the speed of extracting, “4 connected contours” was adopted.
Q_{4}(X,Y) = P(X,Y) \land \{ P(X,Y-1) \lor P(X-1,Y) \}
\lor \{ P(X+1,Y) \lor P(X,Y+1) \} = P(X,Y) \land \{ P(X,Y) \lor P(Y) \lor P(X) \lor P(Y) \}

\begin{array}{c|cccc|cccc}
0000 & 1111 & 1000 & 0000 \\
0110 & 0001 & 1000 & 0111 \\
0110 & 1001 & 1000 & 1001 \\
0110 & 1001 & 1011 & 1001 \\
0000 & 1111 & 0000 & 1001 \\
\hline
0111 & 1110 & 1111 & 0000 \\
0100 & 0010 & 1111 & 0110 \\
0100 & 0010 & 1111 & 0110 \\
0111 & 1110 & 1111 & 0000 \\
\end{array}

Figure 3. Sample of extracting contours

“1” in Figure 3 (a) represented the region of cell; (b) could be got by “NOT” (a); (b) shifted up, down, left, right one bit; four images of (c) could be obtained. (d) was by method of ‘OR’ the four images of (c). (e) being gotten by method of (d) AND (a). (e) is the border of cell nuclear. Because all of the pixels are processed at the same time, the parallel processing is higher in efficiency than line-by-line processing. But this method needed more space of computer memories. So parallel processing increased speed and sacrificed memories.

Description of the Chain-code tracking arithmetic

In order to store information of cells, chain-code tracking algorithm was employed. The steps of chain-code tracking algorithm are as follows:

(a) Search the start contour pixel of a cell by scanning mark of every pixel. Recode the coordinate of start contour pixel S_X, S_Y turning to c;

(b) Continue searching the start contour pixel of a new cell and record S_X, S_Y, then turn to c, if no new cell contour pixel can be found, turn to d;

(c) Scan the eight-circumjacent pixels of the candidate pixel. If no contour pixel can be found, turn to b; if contour pixel is found, record the chain-code value “C_value”. Take this contour pixel as current pixel. Turn to c.

(d) Finish tracking contour.

C. Mode of recording cell contour in program

Cell contours were recorded with VC++6.0, the records of cell contours consisted of the starting coordinates and the chain-code. Because the number of the cell is not certain, a linked list was chosen for recording. CobList in MFC was used to denote the linked list. The object of CobList “m_corelist” was defined. Because CobList requires that every element was an instance of the class derived from CobList, the class “Ccore” derived from Class “Cobject” was designed. The class “Ccore” represented a cell. In Class “Ccore”, variable Cpoint was used to record start pixel of cell contour and dynamic array m_codevalue that was used to record the value of chain-code.

**program:**

Design the class of Ccore deriving from CObject:

```
Class CCore : Public CObject
{……
public :CCore(Cpoint m_point);
protected:
    Cpoint m_point; //start pixel
Public:
    Carray <int,int>  m_codevalue;
    // Store the value of chain-code.
}
```

Create a class deriving from CobList, this class was used to manage chain list which was composed of Ccore*.

D. Lubricity processing

After contours of cell were tracked and recorded, lubricity processing should be done. Since the values of chain-code were recorded according to orientation of contour, the chain-code was first decoded, and then lubricity processing could be done. Decoding the chain-code was guided by the relation of chain-code’ value and cell contour’s coordinate [6]. A lubricity method was chosen to make the contour lubricous.

IV. CONCLUSION

In the experiment, as to different slice, the effects of extracting the cells were different. For the slice with few overlapping of cells, the effects were perfect. And the effect of the dyes of slices also affected the result of segmentation. In our future research, we will analyze texture information of slice’s background in the subsequence studies. We aim to develop a system that can perfectly finish segmentation and cell analysis.
Figure 4. (a) Original image; (b) Segmentation result

REFERENCE