The Image Skeleton Extraction Algorithm Based on Highest Rectangle

Wang Weiqing

Department of Information Management, Southwest University Rongchang Campus, Chongqing
 402460, China

wwqlhy@163.com (corresponding author)

Keywords: highest rectangle; image segmentation; image thinning; skeleton extraction

Abstract. Currently, the thinning algorithms based on the template have no completeness, so there are more error-retention and error-deletion pixels in their thinning results. In this article, using the highest rectangle which can not be eliminated, we put forward a new image-skeleton-extraction algorithm. Because the highest rectangle has completeness, there are no error-retention and error-deletion pixels in its thinning results. The algorithm can effectively eliminate the fracture phenomenon that is generated in the traditional algorithm. At the same time, the skeleton-thinning width is only one pixel, and it achieved to the ideal state. Experiments proved that the thinning result of this algorithm is more effective, and its operational efficiency is better.

Introduction

With the adoption of medical imaging equipment, a growing number of medical image data is stored in the database^[1]. The medical images can provide a variety of characteristic information for diseases and pathological diagnosis, and can provide functional information for identifying diseased tissue or organs. Today, how to quickly and effectively segment those images is a medical problem with challenging^[2-3]. The commonly used algorithms include the quick thinning algorithm and OPTA (one-pass-thinning algorithm) algorithm. But we usually used the improved OPTA algorithm^[4-8]. Currently, there many thinning methods based on PCNN (third-generation artificial neural network) were proposed. But these algorithms can easily produce error branch, break points and burrs. In this article, using the highest rectangle which can not be eliminated, we put forward a new image-skeleton-extraction algorithm. Because the highest rectangle has completeness, there are no error-retention and error-deletion pixels in its thinning results. The algorithm can effectively eliminate the fracture phenomenon. At the same time, the skeleton-thinning width is only one pixel, and it achieved to the ideal state.

The image skeleton extraction algorithm based on highest rectangle

The related definitions and theorems As shown in Figure 1, each small square is a pixel, the light-gray square is edge points and the white square is non-edge points. Figure 1a is an image that will be extracted its skeleton.

Definition 1, the highest rectangle: To scan each line in an image in order, When it scans the first edge-pixel point $f(i_1, j_1)$, look the coordinates (i_1, j_1) as the upper left corner of a rectangular, continue to scan the line, until it metes the first non-edge points or the image border, and note the number of edge pixels in this line as k_1 . Then, scan the next line starting from the column j_1 , and note its edge-pixel number k_2 . Loop for scanning each line in the image as this, until the pixel in the next line is a non-edge point or the image border. Record the scanned number h and the edge-pixel number $k_t (t=1,2,...,h)$ in each line. Let $w = \min(k_1, k_2, ..., k_t, ..., k_h)$, (i_1, j_1) as the upper left corner coordinates, w as the width, h as the height of a rectangle, so this rectangle is called the highest rectangle. The highest rectangle a and b are shown in Figure 3b.

As for the searched highest rectangle, in order to distinguish the pixels in highest rectangle with the other non-edge pixels and the non-edge pixels, we use a special color (such as $RGB(1,1,1)$) as the pixel color in the highest rectangle.

Definition 2, the excluding highest rectangle and non-excluding highest rectangle: while a top highest rectangle is on the top of the highest rectangle and a right-top highest rectangle is on its right, then we call this rectangle as excluding rectangle, other wise call it as non-excluding rectangle. As shown in the Figure 1, b and c are non-excluding rectangles, and e is an excluding highest rectangle.

Definition 3, core: the center coordinates (i, j) in a non-excluding highest is called a core of the non-excluding rectangle. As shown in the Figure 1, the deep-color pixels is the core of the highest rectangle a.

As for the searched non-excluding highest rectangle, in order to distinguish its pixels with other edge pixels, non-edge pixels and the highest rectangle pixels, we use a special color (such as $RGB(0,0,0)$) as the pixel color of the non-excluding highest rectangle.

Definition 4, edge-skeleton line: Considering the two cores in two non-excluding rectangles as the end points, the line connecting the two end points is called edge-skeleton line.

Theorem 1, the highest rectangle has the completeness characteristic, that is to say that any edge image formed by edge detection, will be covered with highest rectangles.

Prove: Suppose that there is an edge image formed by edge detection, and a pixel in it is not covered by any highest rectangle. From the definition of highest rectangle, when it scan the point in the image-scanning process, it can form a highest rectangle whose top-left corner is this pixel and its width and height are all 1 pixel. To use this rectangle to cover the edge points, then there is no edge point that is not covered by a highest rectangle. It is contradictory to the assumption there is a pixel not covered by any highest rectangle. So any edge image formed by edge detection, will be covered with highest rectangles. That is to say the highest rectangle has the completeness characteristic.

Theorem 2, any edge pixel in an edge image belongs to one but only one highest rectangle.

Prove: Assume the value of an edge pixel is $f(i, j)$, (Where $i \leq \text{image width}$, $j \leq \text{image height}$), from the definition 1, when we search the highest rectangles, the scanning order is from top to bottom and from left to right. The coordinates (i, j) belong to the coordinates in the image, so the point (i, j) will be scanned during the scanning process. Because the value $f(i, j)$ of point (i, j) is the edge value, therefore, the point (i, j) may belong to the current highest rectangle.

① When it belongs to the current highest rectangle, from the definition 3, we have to modify the value $f(i, j)$, that is to say this point is no longer an edge point, so it is not considered to belong to another highest rectangle at next scanning time. So it belongs to one but only one highest rectangle

② When it does not belong to the current highest rectangle, from the definition 3, we do not modify the value $f(i, j)$, that is to say this point is still an edge point, so it is considered to belong to another highest rectangle at next scanning time. So it belongs to one but only one highest rectangle

So, any edge pixel in an edge image belongs to one but only one highest rectangle.

Theorem 3, the edge-skeleton width of an edge image formed by any edge detection is only 1 pixel.

Prove: A skeleton line is formed by a straight line, is the connection between the two core point. From the theorem 2, any edge pixel in an edge image belongs to one but only one highest rectangle, that is to say there is only one core in any non-excluding highest rectangle. Therefore, there is only one line between two highest rectangles. Because the size of the core point is a pixel, and coordinates of two end points of the line is the only two points, so, the width of the line must be one pixel, that is to say the edge-skeleton width is only 1 pixel.

Therefore, the edge-skeleton width of an edge image formed by any edge detection is only 1 pixel.

The data structure in the algorithm From the theorem 2: any edge pixel in an edge image is covered by one but only one highest rectangle. Therefore, when we search for the highest rectangles in an edge image, after an edge point are covered by a highest rectangle, and in order to make the point does not to be scanned again and does not to be covered by another highest rectangle, the algorithm using a special data structure to store the state that express if the point is covered by a highest rectangle. While connecting the edge-skeleton line, if the scanning highest is not a non-excluding highest rectangle, we should find if there are non-excluding highest rectangles on the top, left and right of this non-excluding highest rectangle. If there is one, we must connect the cores of the two non-excluding highest rectangles. In order to find these non-excluding rectangles, we must store the core coordinates (i_h, j_h) of the non-excluding rectangle in each pixel of the non-excluding rectangle. Therefore, the algorithm designs a data structure to store the three data of a non-excluding rectangular. Its definition is as follows:

```
public struct zdzb// Define the storage structure of the highest rectangle
{ public bool zhongdian; public int zhongdianx; public int zhongdiany; }
```

The basic steps of the algorithm As shown in Figure 1a, the image is the enlargement edge image that was dealt by edge detection and binarization, each square represents a pixel, a light gray square is an edge pixel, and a white square is a non-edge pixel. It is a skeleton-extraction process of an edge image as shown in figure 1. According to the above the storage structure of a highest rectangle, we can extract the skeleton of the image as following steps:

- ① To scan the image from left to right, and from top to bottom, and find out a highest rectangle, if it find a highest rectangle then go to the step ②, otherwise end the algorithm;
- ② After found out a highest rectangle, determine whether the highest rectangle is a non-excluding highest rectangle, if it is a non-excluding highest rectangle then go to the step ③, otherwise go to the step ⑦;
- ③ To fill the information of zhongdian, zhongdianx and zhongdiany in each pixel in the non-excluding rectangle;
- ④ Determine whether there is a non-excluding highest rectangle on the left of this non-excluding highest rectangle, if there is one, then connect the line between the two cores of two non-excluding highest rectangle;
- ⑤ Determine whether there is a non-excluding highest rectangle on the right of this non-excluding highest rectangle, if there is one, then connect the line between the two cores of two non-excluding highest rectangle;
- ⑥ Determine whether there is a non-excluding highest rectangle on the top of this non-excluding highest rectangle, if there is one, then determine whether there are two non-excluding highest rectangles on the left and the right of this non-excluding highest rectangle, if there is none then connect the line between the two cores of this non-excluding highest rectangle and its top non-excluding highest rectangle;
- ⑦ To fill the information of zhongdian in each pixel in the non-excluding rectangle;
- ⑧ Go to step ① to scan the next pixel, until finish to scan the edge image.

The algorithm of searching for the highest rectangle The step ① of the algorithm will search for all of the highest rectangle in the edge image, the searching process is as definition 1. The characteristic of the searching algorithm is that each searching line is relative to top-left-corner coordinates of the rectangle. Suppose the first searched edge pixel is (x_1, y_1) , and the edge-pixel number which is along the vertical y is h , let $y_2 = y_1 + h$. Assume the starting point of step i is column x_1 , the searched continuous pixel number in the line $i + h$ is k_i . So the rectangular height is: $w = \min(k_1, k_2, \dots, k_i, k_h)$. Let $x_2 = x_1 + w$, then the coordinates of the top-left and right-bottom corner of the highest rectangle is respectively (x_1, y_1) and (x_2, y_2) . Thus, the searching algorithm of a highest rectangle is as follows:

```

for i ← 1 to Height
  for j ← 1 to Width // To scan the edge image from left to right and from top to bottom
    c1 ← GetPixel(j, i); // Get the data of each pixel
    if (c1.R = 0)
      then lefty ← i; leftx ← j; gao ← 0; kuan ← Width; c1 ← GetPixel(j, i);
      while (lefty < Height-1 && c1.R = 0)
        do gao++; kuan1 ← 0
        while (leftx < Width && c1.R = 0)
          do c1 ← GetPixel(leftx, lefty); kuan1++; leftx++
          if (kuan1-1 < kuan) then kuan ← kuan1-1
          lefty++; leftx ← j; c1 ← GetPixel(leftx, lefty)
        leftx ← j; lefty ← j; rightx ← i + gao - 1; righty ← j + kuan - 1

```

The judgment rules of non-excluding highest rectangle and its algorithm When searched a highest rectangle, we must first determine it is an excluding or non-excluding highest rectangle. Its judgment rules is as definition 4, and the algorithm is as follows:

```

if(leftx-1>←0)// Determine whether the above line pixels is in the edge image
then for j1 ← leftx to rightx // Scan the above line of the highest rectangle
    c1 ←GetPixel(j1, leftx - 1) // Get the value of each pixel
    if (zd[j1,leftx-1].zhongdian = 1)// whether it is a non-excluding highest rectangle
        then tichushang←true; break
if (rightx +1 <= Width) // whether the right column line pixels is in the edge image
then for (i1 ← leftx to rightx // Scan the right column of the highest rectangle
    c1 ←.GetPixel(rightx+1,i1 ) // Get the value of each pixel
    if (zd[rightx + 1, i1].zhongdian =1) then tichuyou ← true; break
  
```

As shown in Figure 3b, b and c are non-excluding rectangles, and e is an excluding rectangle.

An example of the image skeleton extraction algorithm based on highest rectangle As shown in Figure 1a, the image is the enlargement edge image that was dealt by edge detection and binarization, each square represents a pixel, a light gray square is an edge pixel, and a white square is a non-edge pixel. After the step ① of the algorithm, it gets the highest rectangles a-q which covered all the edge points of the edge image, it is shown as Figure 4b. After the step ② of the algorithm, it gets the non-excluding highest rectangles and their cores, the result is shown as Figure 4c. After the step ③ of the algorithm, it used DDA algorithm to connect the cores of the two neighboring non-excluding highest rectangles, and fill them with the skeleton-filling color, so it gets the finally edge-skeleton line, it is shown as Figure 1d.

a The original enlargement image

b The searching results of highest rectangles

c The non-excluding highest rectangle and its core

d The skeleton line

Fig.1 An example of the image skeleton extraction algorithm based on highest rectangle

Experiment and comparative analysis of the algorithms

The experimental environment is: CPU is AMD Athlon (tm) 64 X2 Dual Core Processor 3600+, its frequency is 2GHz; the memory is 1G; Operation system is Microsoft Windows XP Professional Service Pack 3; the programming language is Visual C#.net 2005.

The experiment selected medical image database on China Medical Imaging, and selected 10 images in it. After their edge detection, experiment them with the image skeleton extraction algorithm based on highest rectangle and the skeleton-extraction in reference [8]. The result is as figure 3.

According to the results in Figure 3, we can see that: The extraction skeleton from the algorithm in reference [8] is not robust, its break points is too many, the skeleton line is not too smooth, Accidentally deleted more skeleton points, its deformation is too much, and its time complexity is higher. At the same time, we can found that: The extraction skeleton from the algorithm based on highest rectangle is more robust, its break points is little, the skeleton line is more smooth, Accidentally deleted skeleton points is little, its deformation is small, and its time complexity is lower.

Fig. 3 The skeleton-extraction-contrast experiment

Conclusion

Medical image-skeleton-extraction algorithm has important practical value in medical image process, and other research areas, extraction of skeleton has been highly concerned about. In the skeleton extraction process, Traditional template-based skeleton extraction algorithms need more comparison operation, their time complexity are higher. For the incompleteness characteristic of the elimination and reservation templates, so there are some error-retention and error-deletion pixels in their skeleton-extraction results, so there are more breakpoints and superposition points in the image. We deeply studied the skeleton extraction algorithms, and changed the traditional thought of skeleton-extraction algorithms based on templates, designed a new skeleton-extraction algorithm based on the highest rectangle. It only needs to search non-excluding highest rectangle, then use DDA algorithm to connect the cores, so it need less comparison operation, and highest rectangle has completeness, therefore, there is no error-retention and error-deletion pixel in the image. At the same time it can be proved that the width of the extracted skeleton is only one pixel, and there is less break point. It improved the extraction efficiency and its computation-time efficiency. Experiments show that: The computation time of the algorithm based on the highest rectangle is more efficient. There are small break points in the skeleton-extraction result. It laid the foundation for medical image processing and other areas.

References

- [1] Wu Shuang, Zhang Yifei, Xiu Fei, Wang Daling, Bao Yubin, and Yu Ge. An Interest Point Based Feature Extraction Method for Medical Image Classification[J]. Journal of Computer Research and Development. 2007(suppl):429-334.
- [2] ZHU W ei-feng, LI dun-li, W EI Ping. Segm entation of m edical im age based on the grey relational analysis[J]. OPTICAI TECHNIQUE. 2007(suppl):429-334.
- [3] W ang Peng, Zhang Youguang, Zhang Shuo. A Synthetically Thinning Alg0rithm f0r Fingerprint Image[J]. JOURNAI OF COMPUTER—AIDED DESIGN & COMPUTER GRAPHICS. 2009,21(2):179-189.
- [4] Feng xingkui, I i I inyan, Yan Z uquan. A new thinning algorithm f0r fingerprint image[J]. Journal of Image and Graphics: A , 1999, 4(10): 835—838 (in Chinese)
- [5] Wang Yelin, Ning xinbao, Yin Yilong. study on the fingerprint image thinning algorithm[J]. Journal of Nanjing University: Natural Sciences, 2003,39(4): 468-475 (in Chinese)
- [6] Wang Jialong, Guo Cheng' an. An improved Image template thinning algorithm[J]. Journal of Image and Graphics: A, 2004, 9(3): 297-301(jn Chinese)
- [7] YANG Fan, ZHAO Shun-dong. Effective mixed fingerprint image quick thinning algorithm[J]. Application Research of Computers. 2008,25(1):3034-3035.
- [8] MEI Yuan, SUN Huai-jian, XIA De-shen. An Improved Template-based Rapid Thinning Algorithm[J]. Journal of Image and Graphics. 2006,11(9):1306-1311.