



VC++ Programming Implementation of Profile Processing for BMP Image File Extraction

Zheng Dong, Liangyu Li & Junfeng Li

College of Information and Communication Engineering

Tianjin Polytechnic University, Tianjin 300160, China

E-mail: dongzheng829@163.com

Abstract

To analyze the image shape, it is necessary to express the profile character of image by special method under certain situation, and the aim of image profile extraction is to obtain the exterior profile character of image. Taking the BMP image file as the example, in this article, we used VC++ to implement the read, image strengthening, threshold division and profile extraction of 256bits color graph, and obtained the bitmap file which could fulfill the image analysis finally.

Keywords: BMP image file, VC++, Profile Extraction

The image profile extraction is an important image processing technology, and it is the key approach to better implement image analysis and the based to fully grasp image information.

The profile of the image is one important character of the image. The so-called profile means the set of the pixel points which pixel values change obvious surrounding the image, and it is the basic character depending by image analysis.

1. BMP image file

BMP (Bit Map Picture) format is a sort of standard format to exchange image and image data in Windows. The data of BMP image include bit map file header, bit map information header, palette and image data.

In the BMP image file, the bit map file header is the structure which length is fixed to be 14 bytes and the bit map information header is also the structure which length is fixed to be 40 bytes. The palette exists only for those bit map files which need palette. The true color palette doesn't need palette, and the bit map information header directly follows image data. The image data is the actual image data. For the bit maps which use palette, the image data are the index values of the image pixel color in the palette, and for the true color image, the image data are actual R value, G value and B value.

2. Image processing process

2.1 Image read

The image read is the first approach for the image processing, and in this approach, the height and width of the image should be read, and whether the image needs to be printed should be judged and the memory space should be distributed for reading image data.

The bit file is stored according to rows, and it is scanned from left to right and from the top down, and recorded the pixel value of every pixel point in turn. Supposed that one image has m rows and n lines, the storage sequence of BMP file is to scan the information in the $m-1$ row from the 0th list to the $n-1$ list, and scan the information in the $m-2$ row until the 0th row is scanned completely.

2.2 Image smoothing

Image smoothing is a sort of applied digital image processing technology, and its main aim is to reduce the noise of the image. The neighborhood average generally can be used to reduce the noise in the space filed. And in the frequency field, because the noise spectrum always exists in the high frequency band, so we can adopt various low pass filters to eliminate the noise.

In the experiment, we adopt the idea of smoothing template to eliminate the noise, and it mainly eliminates certain noise in the image through the average operation of one points with several neighboring points, and whether the template we select, we treat the present pixel and neighboring pixel points equally without discrimination and implement average processing uniformly, so we can effectively eliminate the noise in the image, and the main advantages of the method include simple algorithm and quick computation speed, and its disadvantage is that it will induce certain blurs in the

image.

We can adopt different templates to eliminate different noises, and the smoothing effect is related with the diameter of the neighborhood (the size of the template). The diameter is larger, the blur degree of the image is higher, or vice versa.

The usual templates include $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, $\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ and $\frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$.

The mathematical meaning of the neighborhood averaging can be denoted by the following formula.

$$g(x, y) = \frac{1}{M} \sum_{(i,j) \in S} f(i, j)$$

Where, $x, y = 0, 1 \dots N-1$, S is the set of the neighborhood taking (x, y) as the center, and M is the integer in S .

In the experiment, we utilize the VC++ to compile the code to implement the image smoothing processing, and the

result which applies the average template $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ to deal with the image is seen in Figure 2.

2.3 Binary processing and threshold division

The usual threshold processing is the binary processing of the image, which selects one threshold and converts the image into the binary image for image division, margin follow and other pre-processing.

$$g(x, y) = \begin{cases} 0 & f(x, y) < T \\ 255 & f(x, y) \geq T \end{cases}$$

The function of the threshold processing is to appoint a threshold by the user, and if the grey value of certain pixel in the image is less than the value, so set up the grey value of this pixel as 0 (black), or else set up the grey value of this pixel as 255 (white).

In the threshold process of the image, it is very important to select the threshold, and the processing results will be different because of different thresholds. If the threshold is too big, the surplus part in the image will be extracted, and if the threshold is too small, some concerned things will be lost.

Based on above reasons, the method to select the threshold in the experiment is the method of the self-adaptive threshold selection.

The self-adaptive threshold selection is that each pixel confirms a neighborhood window surrounding itself, and then we compute the maximum and minimum of the pixels in the window, and take the average of the computation results as the threshold of the image. As seen in Figure 3, the selected window includes 8 neighborhoods, and in practice, we can also select bigger window, but if the window is bigger, we will need deal with more data and more time.

In the Figure 3, suppose C is the present pixel, P are the 8 neighborhood pixels surrounding C , and suppose \maxValue denotes the maximum value in the 8 neighborhood pixels, and \minValue denotes the minimum value in the 8 neighborhood pixels, so its threshold is

$$T = \frac{\maxValue + \minValue}{2}$$

The processing result by adopting the confirming method of self-adaptive threshold is seen in Figure 4.

2.4 Image profile extraction

The aim of profile extraction is to obtain the exterior profile character of the image and prepare for the shape analysis of the image.

The algorithm of binary image profile extraction is to empty all interior points. If one point in the original image is black and its 8 neighboring points all are black, it indicates this point is the interior point, and it should be deleted (set up its white pixel values is 255). Implement this operation to all pixel points in the image to complete the extraction of image profile.

The profile extraction result of the image in the experiment is seen in Figure 5.

3. Analysis of experiment results

In the experiment, we comprehensively utilize the BMP image file read, image smoothing, image binary processing and threshold division, and image profile extraction to basically complete the pre-processing of image.

From the example, we can see that the noise can be effectively eliminated after the image is smoothed. The profile extraction can offer conveniences for the analysis or identification of image shape.

References

Liu, Shuang. (2005). Study of Image Threshold Methods' Selection and Algorithms Implementing for Image Segmentation. *Computer Technology and Application*. No.21(2). P.73-75.

Qiu, Shikeji. (2006). *Representative Algorithm and Implementation of Visual C++ Digital Image Processing*. Beijing: People's Posting Press. P.370-418.

YanF X, Zhang H & Kube C R. (2005). A multistage adaptive thresholding method. *Pattern Recognition Letters*. No.26. P.1183-1191.

Zhang, Junli & Liu, Shuzhi. (2005). *Practice and Enhancement of Visual C++: Image Processing and Engineering Application*. Beijing: China Railway Publishing House. P.115-198.

Zhou, Changfa. (2004). *Mastery of Visual C++ Image Processing Programming*. Beijing: Electronics Industry Press. P.230-299.

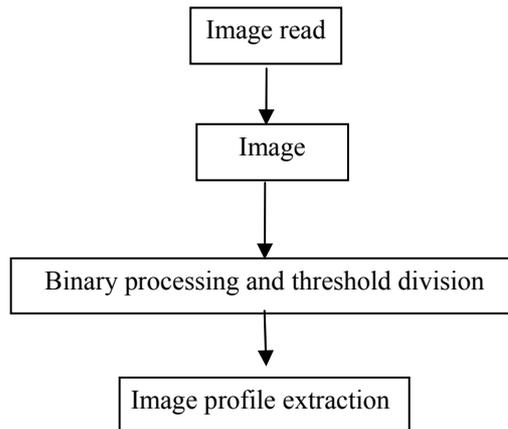


Figure 1. Approaches of Image Profile Extraction

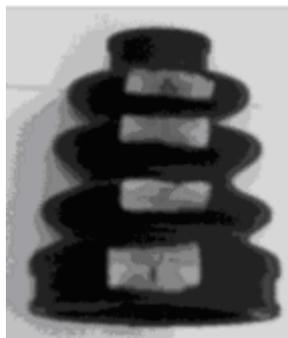


Figure 2. Image after Average Template Processing

P		
	C	

Figure 3. Confirming Method of Self-adaptive Threshold



Figure 4. Image after Self-adaptive Threshold Processing



Figure 5. Result of Image Profile Extraction