|  |
| --- |
| ME 5405 Machine Vision |
| Assignment- computing project |
| Submitted by |
| Dai Weiheng学号 |
| Luo Zhiyao学号 |
| Yang Wenjie A0209483Y |
| 2019/2020 |

[1 Introduction 3](#_Toc21877381)

[2 Methodology 4](#_Toc21877382)

[2.1 Image display 4](#_Toc21877383)

[2.2 Thresholding](#_Toc21877384) 4

[2.3 Pattern Fragmentation 7](#_Toc21877385)

[2.4 Character rotation 11](#_Toc21877386)

[2.5 Outline recognition 11](#_Toc21877387)

[2.6 Character Thinning 11](#_Toc21877388)

[2.7 Character Rescheduling 11](#_Toc21877389)

[3 Summary 12](#_Toc21877390)

[3.1 Overall flowchart and screen dumps 12](#_Toc21877391)

[3.2 Comparison of algorithm implemented in the two images 12](#_Toc21877392)

Ⅰ. Introduction

The project requires us to process two images which are separately composed of a 64 by 64 coded array (i.e. image1) and a BMP colored image of the label on a microchip (i.e. image2). Images1 contains an alphanumeric character ranging from 0 to 9 and A to V for each pixel in the image, which represent 32 levels of gray value respectively. According to the instruction, the images are expected to be processed to display different effects with MATLAB, including image display, binarization, character segmentation, image rotation, outline depiction, picture skeletonization and character reordering.

The algorithms implemented to fulfill the tasks and the comparison of peculiarity of using different methodologies will be expounded step by step in the next chapter.

ⅡMethodology

# 2.1Image display

### Image Reading

Matlab provides a variety of image reading command supporting different formats. In terms of image2, we can simply read the file using the following command:

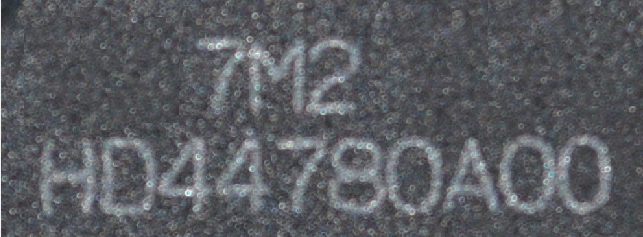
Im2\_rgb = imread('charact2.bmp');

where rgb is a variable assigned by the matrix containing color and intensity information of Image 2. More specifically, the variable rgb is a 3679903 uint8 array representing 367990 pixels in red, green and blue color channels respectively.

To show Image 2 on screen, we utilize command:

imshow(Im2\_rgb);

The screen dump we obtain is shown in figure.



For Image1, it is not possible to derive the image array though ready-made function in Matlab. Firstly, the txt file is read into Matlab in the format of string matrix. By inspection the matrix is 14096 so we reshape it into a 2-dimensional 6464 string matrix. After that, every element is traversed in the matrix and judged by a conditional judgement statementto find out the string whose ASCII code is not less than the ASCII code of ‘A’. Then we reassign corresponding number value to those elements which satisfy the criteria of the conditional judgment so that the whole array will be transformed into a string array full of number strings. Finally we use function “double()” to convert the string matrix into a double-floating matrix. In this manner, Image 1 will then be able to displayed on screen using “imshow()”, as can be seen in figure. The Matlab code of showing Image1 is as follows:

代码

A much more succinct method of mapping A-Z to 10-35 is to apply table array. A ASCII match table shown as table is built in order to construct the corresponding one-to-one match.

|  |  |  |  |
| --- | --- | --- | --- |
| number | 0~9 | ASCII code between 10~A | A~Z |
| string | 0~9 | 0 | 10~35 |

Matlab code of the conversion is shown as follows:

Table代码

The snapshot of displaying Image 2 is shown in figure.

### Image Monochrome

Matlab supports converting color image to image monochrome using function “rgb2gray()”. It converts RGB values to grayscale values by forming a weighted sum of the R, G, and B components:

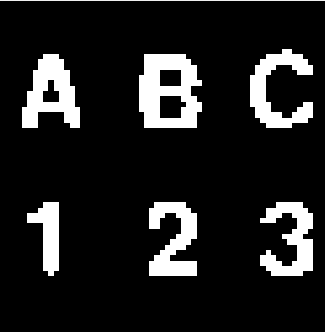
Since floating-point operation is with low efficiency, the conversion of RGB image should involve floating-point operations as little as possible. Therefore, we can transform equation into integer division:

Then we can derive grayscale pictures of image1 and image2 as shown in fig

一二的灰度图

2.2Binarization

**Preprocessing of Image 1**

Image 1 has relatively simple composition whose background pixel brightness is zero, considering its few noise points and interference factors, the threshold can be set as zero for image binarization.

**Preprocessing of Image 2**

**1Otsu’s threshold**

Maximum interspecific variance method, as known as Otsu’s method, is an adaptive threshold method proposed by Japanese scholar Otsu in 1979，which also used as the theory of function ’graythresh’ in matlab. It divides the image into two parts, background and target, according to the gray level characteristics of the image. This method aims at computing a thresholding point T such that the errors in wrongly classifying is minimized.

The algorithm is simple in calculation and not easily affected by the brightness and contrast of images, so it has been widely used in digital image processing. Since variance is a measure of gray distribution uniformity, the larger the interspecific variance between background and foreground is, the larger the difference between the two parts of the image is, and the segmentation with the largest inter-class variance means the smallest misclassification probability.

For the obtained gray image, the segmentation threshold of the object and background is set as T. The number of pixels whose gray value is less than the threshold T, the proportion of the number of pixels belonging to the object in the whole image, and the average gray value of target are respectively denoted as N0, w0 and ave0. The corresponding number, proportion and the average gray values of background are denoted as N1, w1 and ave1. The total average gray scale of the M×N image is ave, and the interspecific variance is g. Substitute equation (3) into equation (4), and get the equivalent formula (6) of inter-class interspecific:

Finally, the threshold T that maximizes the interspecific variance g is obtained by means of traversal method to realize the binarization.

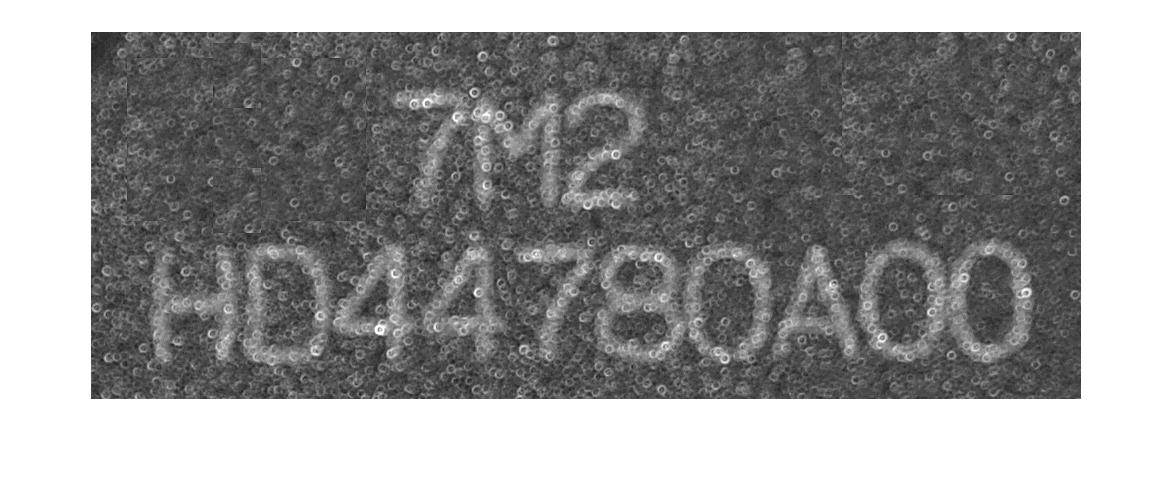
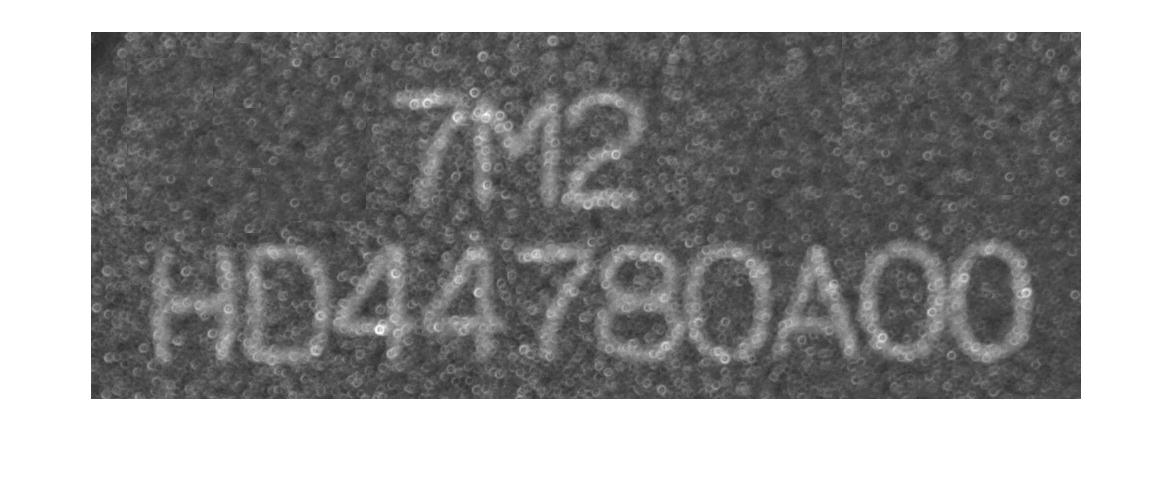
代码及图片

It can be seen that Otus’s thresholding doesn’t work well on Image 2.

The reason why Otus’s thresholding cannot handle the spotted noise distributed in image 2 is thatnoise spots are with excessive brightness. A naïve way to attenuate brighten spots is to rise the thresholding value T, as a result of which, however, the outline of relevant characters will fade out as T inclines. An example is shown in figure. Therefore, preprocessing approaches are implemented to resolve this problem.

**2Sharpening**

  In order to improve the high frequency components and enhance the image contour, the original image is preprocessed by unsharp mask (equivalent to using low-pass filter).High-frequency-enhanced areais obtained by calculating the difference between original image and preprocessed image, discounted by a correction factor.And thenthe overall sharpened image is derived by summing the original image with the high-frequency-enhanced area. In Matlab, we can use imsharpen(image, ‘Radius’, r, ‘Amount’, a)to realize image sharpening, where 'Radius' denotesstandard deviation of the Gaussian lowpass filter, and 'Amount' denotes strength of the sharpening effect, respectively.

The effect of image sharpening is shown in figure

**3Gaussian Filtering**

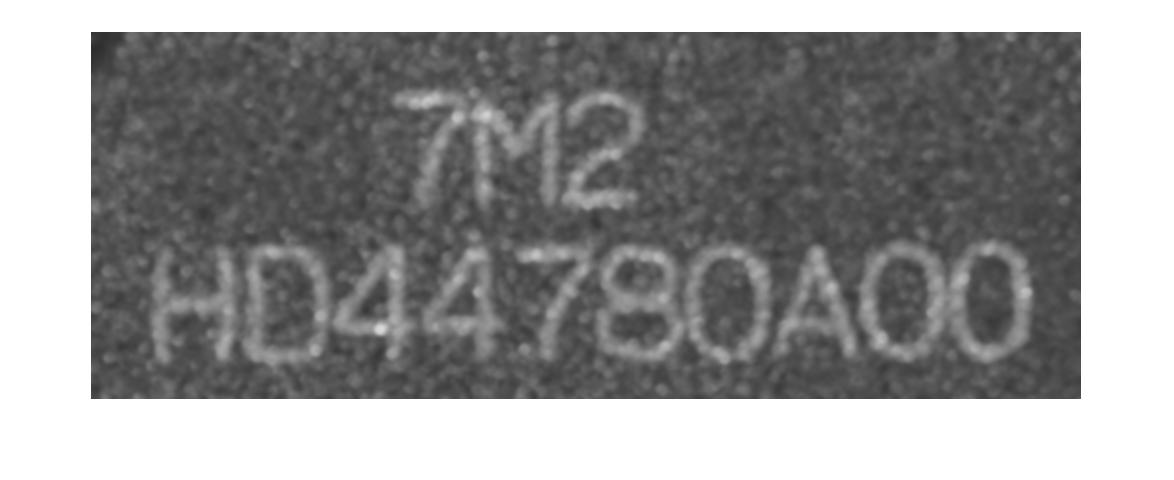
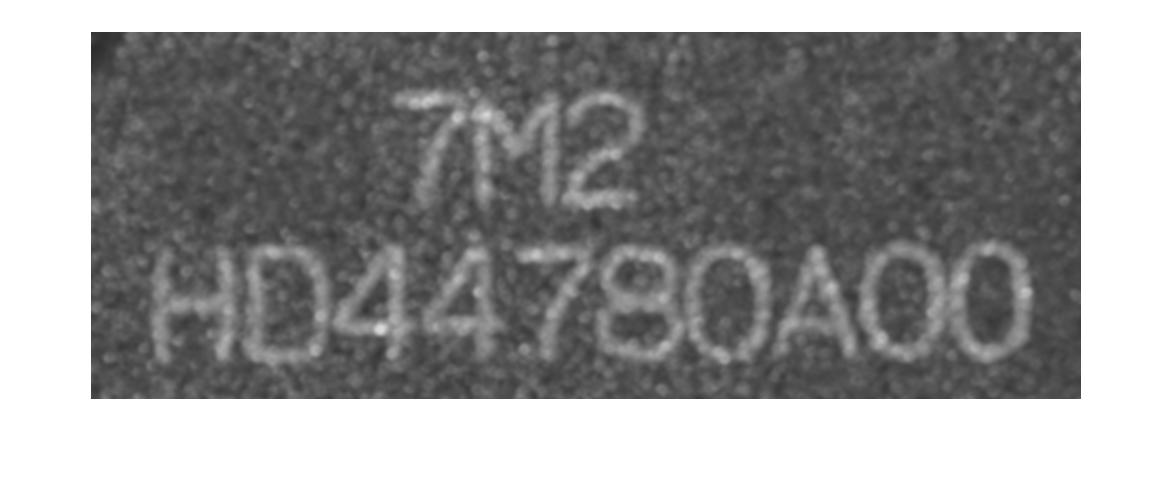
高斯滤波是一种线性平滑滤波，适用于消除高斯噪声，广泛应用于图像处理的减噪过程。通俗的讲，高斯滤波就是对整幅图像进行加权平均的过程，每一个像素点的值，都由其本身和邻域内的其他像素值经过加权平均后得到。

高斯滤波的具体操作是：用一个模板(或称卷积、掩模)扫描图像中的每一个像素，用模板确定的邻域内像素的加权平均灰度值去替代模板中心像素点的值。

高斯滤波后图像被平滑的程度取决于标准差。它的输出是领域像素的加权平均，同时离中心越近的像素权重越高。因此，相对于均值滤波（mean filter）它的平滑效果更柔和，而且边缘保留的也更好。

**4Bilateral filtering with Gaussian kernels**

网址<https://blog.csdn.net/Jfuck/article/details/8932978>

Gausianbalitary

Binarizedimage after preprocessing

**5Open Operation (Image Geometry)**

**代码 图**

6Fillin the hole

According to the observation, there is a small hole in the letter h, which may affect the subsequent contour extraction. Function imfill in MATLAB is used to fill the target area in advance:

[length0, width0]=(size(segment\_image));

part1=segment\_image(:,1:round(width0/10));

part1=imfill(part1, 'holes');

part2=segment\_image(:,round(width0/10)+1:width0);

img\_fill = cat(2,part1,part2);

subplot(3,4,8);

imshow(img\_fill);

2.3Pattern Fragmentation

There are two main steps to achieve character segment. Firstly, different characters are separated according to the connected region, and secondly, eliminate the phenomenon of character adhesion.

The labeling method is adopted to detect the connected region and conduct character segmentation. Starting from the first pixel, each element in the image is traversed to determine whether it is zero. If the pixel is zero, it will be regarded as background and not be labeled, and the probe will continue. If the pixel is a non-zero element, the current label will be assigned to this pixel and then be increased by one. The subsequent pixels will be detected next.

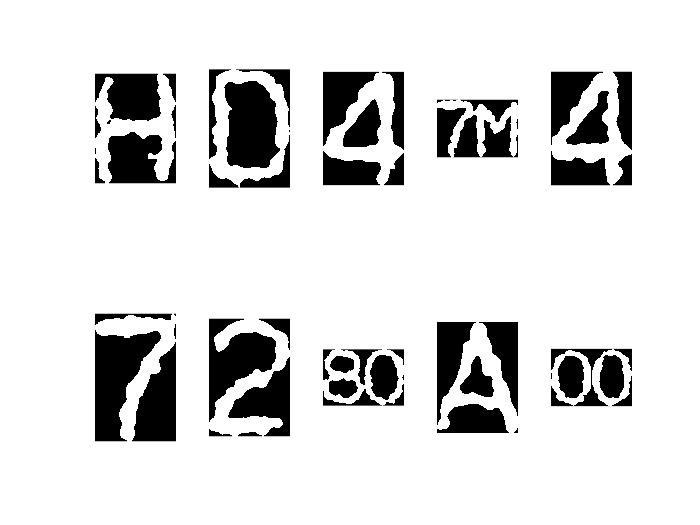
代码

Once the traversal is complete, the detection is traversed again from the first pixel. The connected region can be divided into two types, 4-neighbor-connected and 8-neighbor-connected. According to the selected connecting mode, the non-zero detected pixel and its upper and left pixels are taken as the minimum value among these 3 pixels. If the label of detected pixel is zero, it will be skipped to the next pixel until the traversal is completed. The area with the same label is stored separately to achieve the effect of character segmentation.

代码

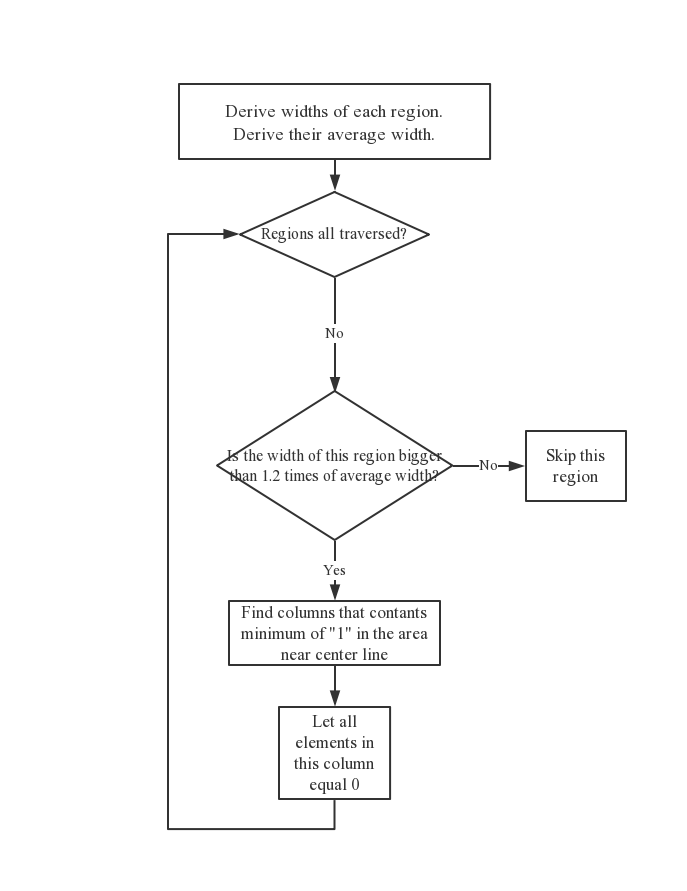
以下是流程图及分割后的屏幕截图

|  |
| --- |
| region=regionprops(binary\_image,'Image','BoundingBox'); |

[stats](https://www.mathworks.com/help/releases/R2019a/images/ref/regionprops.html?searchHighlight=regionprops&s_tid=doc_srchtitle#buoixjn-1-stats) = regionprops(binary\_image,'Image','BoundingBox')

This returns a struct array containing a struct for each 8-connected component (object) in the binary image. Therefore, we can obtain characters fragments as shown in figure.

By inspection some characters are connected with each other. To separate them, we develop an algorithm to detect and isolate characters. Flow chart of above algorithm can be seen in figure.

代码The code is shown as following:

2.4Character rotation

<https://blog.csdn.net/lkj345/article/details/50555870>

For independent letters, rotation matrix is used to complete the rotation around the centroid of the image via point processing.

[x1, y1, 1]=[x0, y0,1]

Before performing rotation, we need to set the center of the image as the center of rotation.

[x1, y1, 1]=[x0, y0,1]

Secondly, the length and width of the original image will be changed, so it is necessary to enlarge the image in advance.

代码图前处理

Due to that coordination in matrix are integers, and new coordinates of pixels in the rotated matrix are decimal form, the reverse mapping combined with interpolation method is used to eliminate the error. the rotated image of the whole node map, back to the original image according to the original image in the grey value of pixels around this point interpolation method is used to judge the right grey value, the rotation is applied to new pixels. Find the original coordinates of each pixel of rotated image, according to the gray value of the pixels around the coordinate in the original image, the suitable gray value is determined by interpolation method. After completing the rotation, reset the origin back to the top left corner.

The inverse form of the rotation:

[x0, y0, 1]=[x1, y1, 1]

代码图旋转及插值

Bilinear interpolation is linear interpolation in two directions using the gray scales of four adjacent pixels, for example:The computation of bilinear interpolation method is more complicated than that of the nearest point method. It has the property of low pass filtering, which damages the high frequency components, and the image looks smoother.

2.5Outline recognition

The object of image contour extraction can be binary image or gray level image.

Binary image contour extraction is based on the preprocessing of the images, only need to hollow out the internal pixel points. If the gray values of the 4-connected neighbors are all one, it can be estimated that this pixel belongs to the internal points, otherwise it is an outline point. Converting all the internal pixel to background(gray value=0) can extract the outline of an binary image. In this case, the estimation of 4-connectedor 8-connected neighbors can result in similar outline, thus the former method is chosen, the code and image are shown in fig.代码及图片

<https://blog.csdn.net/wqvbjhc/article/details/6065484>

<https://blog.csdn.net/GoodShot/article/details/10051309>

2.6Character Thinning

<https://blog.csdn.net/superdont/article/details/4621820>

2.7Character Rescheduling

After the character segmentation, it is found that the size of each character region are not equal, so the maximum character is taken as the standard, and the matrix of other characters are expanded, so that all character matrices could be connected by cat function and rearranged and displayed proportionally in the same line.

代码图及图片

ⅢSummary

3.1Overall flowchart and screen dumps

The overall flowchart is shown as following Figure3.1:

The screen dumps of specific steps are appended after the corresponding content. The Figture3.2 shows the overall result.

3.2Comparison of algorithm implemented in the two images

The first image is to convert the coded array in the text into corresponding gray values, so as to convert the text information into a digital matrix, and present it in the form of images for subsequent processing. Due to the small amount of information with no noise points in the background and that there are 32 gray levels, so the threshold selection applied to binarization is relatively simple. The preprocessing mainly lies in the corresponding transformation of [text](javascript:;) [content](javascript:;) and gray value. The second bmp image is a color image with many interference factors and unclear contour of object. Firstly, the image is transformed into gray value image, and then a series of preprocessing such as denoising, smoothing and sharpening are carried out on the gray value image. Thus, the appropriate threshold can be selected to turn the gray scale image into binary image. In this image, the optimization is very important, which is the basis of a series of subsequent processing. In terms of character segmentation, image 1 and image 2 can be processed in a similar way, but image 2 requires additional cutting processing to segment and recognize the merged characters. The diversity between the input type and the picture content leads to that the most significant difference between the processing of 2 images is the pre-processing period. The operations of rotation, outline extraction, thinning, scale and [realignment](javascript:;) of the segmented binary image are basically similar as well.

conclusion