School of Electronic and Electrical Engineering
FACULTY OF ENGINEERING

UNIVERSITY OF LEEDS

XJEL3875

Project Report

A Study of intelligent identification for the images of fabric defects

SID:	Project No. 9
Supervisor:	**Assessor:**

School of Electronic and Electrical Engineering
FACULTY OF ENGINEERING

UNIVERSITY OF LEEDS

XJEL3875 Individual Engineering project

Declaration of Academic Integrity

Plagiarism in University Assessments and the Presentation of Fraudulent or Fabricated Coursework

Plagiarism is defined as presenting someone else's work as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance.

Fraudulent or fabricated coursework is defined as work, particularly reports of laboratory or practical work that is untrue and/or made up, submitted to satisfy the requirements of a University assessment, in whole or in part.

Declaration:

- I have read the University Regulations on Plagiarism [1] and state that the work covered by this declaration is my own and does not contain any unacknowledged work from other sources.
- I confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes.
- I confirm that details of any mitigating circumstances or other matters which might have affected my performance and which I wish to bring to the attention of the examiners, have been submitted to the Student Support Office.

[1] Available on the School Student Intranet

Student Name: .... Project No. *9*

Signed: .... Date: *21.05.2020*

Abstract

As raw materials, cloths need to be detected to insure whether there is any defects and flaws on their surface or not, the intact fabrics will be processed and the blemished one will be rejected. However, so far most of textile companies still detect the original material depending on labour's vision which is easy to be influenced by the inspector's experience and physical conditions, and by the surroundings. It requires an expensive cost to train the workers who can identify all kinds of defects' characteristics, and the method is inefficient with high missing rate.

Aiming at the drawbacks and insufficiency of human's vision applied to defects detection, based on computer vision, a system is developed to detect the position as well as the area of solid fabric defects through the image of the fabric samples. Further, numerous images of various fabric defects are used to test the system's accuracy, afterwards, modify the system to improve the efficiency and accuracy of the fabric defects detection system.

Compared with existing defects testing technique, this system have a better performance of more efficient detection with higher accuracy. A fabric detection system based on python and Open Source Computer Vision Library (OpenCV) is built as the result of project, consisting of figure process, defects detection and defects extraction.

Acknowledgement

I would express my utmost gratitude to the Professor ZHANG Xuexia for her unfailing encouragement, support and guidance throughout the whole project. The far-sighted advise and enthusiasm for work I received was vital to the accomplishment of the project.

I also wish to thank the " 2019 guangdong industry innovation competition" for offering the open source of high-quality fabric images, which provided the source materials for detection system.

I would like to express my sincere thanks to my family and friends for their support and encouragement throughout the period of my course.

Finally, I would like to express my heartfelt thanks to all the teachers who have reviewed and defended my thesis in their busy schedule.

Contents List

Symbols and Abbreviations

OpenCV	Open Source Computer Vision Library
Gray	The value of grey channel
B	The value of blue channel
G	The value of green channel
R	The value of red channel
sigma	The standard deviation of the Gaussian convolution kernel
I	The input image
O	The output image
I (r, c)	The pixel greyscale value at r th row, c th column of the input figure
O (r, c)	The pixel greyscale value at r th row, c th column of the output figure
T	Threshold
scr	Source images
dst	Destination images
NMS	Non-maximum suppression
alpha	Norm value or lower range boundary get by the range normalization
beta	Not used for the norm normalization represents upper range boundary of the range normalization

Figures and Tables

Equations

Introduction

Over the years, fabric defect detection is an important step of the textile industry's production and quality management. For the moment, most of the fabric tests are manual, whose accuracy is easily impacted by some subjective factors and lacks consistency. Moreover, long-term working in bright light could cause a significant negative impact on vision. According to the survey, the manual defect detection methods at present can just detect approximate 40% - 60% of all the defects, even sophisticated fabric detection workers can only detect about 70% defects [1]. With the improvement of the fabric industry production efficiency, the manual defect detection's speed and accuracy are gradually could not meet the requirement of factory detection, the guarantee of the production efficiency and the improvement of the products' quality is vital for the modern fabric industry.

Although some fabric defect detection commercialized products are lauched according to the research of scholars around the world, the hardware of the defects detecting system is expensive, many small or medium sized enterprises have to spend millions of dollars to get a set of hardware of the fabric inspection detection system, which is not a very good deal. In addition, the system's distinguish-ability of the variety of defects and adaptability of variety of fabric are relatively limited. The suitable objects of the systems usually are monochrome fabric and only few types of defect can be detected. The value of realization of fabric defects intelligent detection through the technology of computer vision and artificial intelligence is undoubtedly significant. The requirements of the fabric defects detection system to the hardware is not high, meanwhile its accuracy of defect detection is high with cost saving.

The project research and develop an intelligent fabric defects detection system basing on the computer vision algorithm, which efficiently and reliably improves the accuracy of the fabric defects detection system and reduce the dependence on manual workers. The traditional code is cumbersome and verbose, which accounts for inefficiency and difficulty to maintain. Compared with traditional code, as a popular and reliable computer vision library, OpenCV provides various image processing algorithms and functions, and it can realize some complex algorithms through a short instruction which can greatly reduce workload and improve readability of the code.

History and Development

The defect detection technology of cloth based on machine vision sprang up since the 1980s. With the continuous research and exploration of China and foreign scholars, gratifying achievements have been achieved in the field of fabric defect detection and classification. Due to the diversity of cloth materials with different textures, there are many kinds of cloth produced and various defect types, and the image of cloth defects collected by machine vision is susceptible to the influence of light and noise, which makes the defect detection and classification research still have great challenges.

Currently, there are three kinds of detection methods for fabric defects China and abroad: statistical method, spectrum method and model method. Generally, the approaches to fabric defects detection are consists of the following categories [2] according to the different ways and theories of image preprocessing: structural method, statistics method, method through analysing frequency spectrum and the method based on model.

1.Structure method to detect the fabric defects: Texture is treated as a combination of texture primitives in the structure method then analyse the texture by acquiring texture features and deducing their substitution rules. According to this method, the texture of fabric pattern can be realized through combining simple texture structures. The structural texture analysis consists of two sequential stages: detection of basic fabric texture and modelling the overall fabric texture pattern. However, the structure method can only detect and extract the fabric defects from the fabric with regular pattern and the result is unreliable.

2.Statistics method to detect the fabric defects: In statistics method, first order statistics and second order statistics are used to extract fabric texture, it contains the anisotropy and consistency between the structure of fabric textures. Defect detection using statistics method includes the following specific methods: grey-level co-occurrence matrix, Mathematical morphology and Histogram statistics.

A. grey-level co-occurrence matrix (GLCM): This statistical method generalize the texture analysis method based on the fact that image pixels' distribution in the space contains the relationship between image texture messages. Because the grey-scale image is composed of pixel blocks with close grey-scale values, the characteristic values of 14 system characteristics such as entropy, moment of inertia, homogeneity and angular second-order distance of the GLCM,which can greatly represent the texture characteristics of two-dimensional fabric. GLCM could characterize the plane fabric texture features very well, but the method has two drawbacks: Poor performance when solve the image with high resolution, and high cost of the calculation using GLCM.

B. Mathematical morphology: The method extract the features of the image based on information of the geometric objects, dilation operation, erosion operation, open operation and close operation are the common operations. So far, processing image within mathematical morphological method has gradually developed to be the key point of image processing research.

C. Method of Histogram statistics: Histogram is a figure which contains the value distribution of all the image pixels. Histogram usually collect and array the data according to mean, standard deviation, variance, and the median. However, because of the simple grey-level histogram calculation process and the high dimensions of feature vector generated by Local Binary Pattern (LBP) causing the complex calculation, this method uses less because low reliability.

3.Fabric defect detection through analysing frequency spectrum: The method through the image in the frequency domain to inspect fabric defects, because the rule of frequency spectrum resembles the periodicity of basic texture elements' height in the fabric, including Fourier transform [3], wavelet transform [4], [5] and Gabor filter [6].

A. Fourier transform: This method is a common method of researching image texture in frequency domain through transform the data from time domain then analysing in detail. The frequency components in this method is fabric textures, and figure' s size is invariant in the rotating progress, The texture structure variation greatly impacts on the Fourier transform, so the approach has poor real-time performance and weak versatility.

B. Wavelet transform: This method is developed as a substitution of Fourier transform which detect the fabric through analysing the signal. The main benefits of wavelet transform are the feature of multi-resolution analysis and partial analysis. The defects generally exist in a part of surface of the fabric of the and some other methods need to analysis the entire surface of the fabric. Therefore, there are more and more widely usage of wavelet transform in the aspect of fabric defects defection.

C. Gabor filter: The filter can be customized in different proportions and orientations according to the texture structure. The size of filter and the position to process the image are determined in the image processing by the frequency and direction of the filter. This method is suitable for the detection and analysis the texture structure of the fabric.

4.Based on the model to detect the fabric defect: Common methods of fabric defect detection are Auto Regressive (AR), Gauss Mark off (GM) and other models. Firstly, variance projection was carried out on the grey value of the sub-window in the horizontal and vertical directions, then spectrum estimation was carried out on the variance projection of grey value through GM model. After spectrum estimation, the normal fabric and the fabric to be detected would have their respective correlation coefficients. By comparing the different correlation coefficients, the purpose of detecting the defects was realized. The model-based method contains complex calculation, few types of fabric defects can be detected with poor real-time performance and generality. So there is no recent research on this method .

Based on the lucubrate and comprehensive application the knowledge of computer theory, pattern recognition, automatic control theory, some of foreign researchers have launched the fabric defect detection commercialized products, such as: BARCO cloth inspection system (Belgium), Uster company's Fabriscan [7] automatic cloth inspection system (Switzerland) and EVS company's I-TEX cloth inspection system (Israel).[8] Foreign automatic detection system mostly composed of different hardware modules, equipment's operation requires professional technicians, moreover, both the price and later maintenance costs of the equipment are high. The Fabriscan from Uster, for instance, costs about $200, 000. Therefore, Chinese small and medium-sized textile enterprises generally rarely buy foreign automatic testing equipment in view of the reduction of production costs.

Chapter 1: Technical Analysis of Detection System

To realize the process of the detecting image defects, 3 main steps should be consisted: image pre-processing, image segmentation and defects extraction.

1. Image Preprocessing

The fabric images collected from the textile produce sites usually contain noises because of the illumination of real environment, the quality of camera codec quality and so on. The image noise blurs the image even covers features of the image, which brings difficulties to the analysis. Meanwhile, the margin of defects usually Therefore, image de-noising is necessary before processing the images.

1.1 grey-scale transformation

Known as there are 3 channels of colours value in colour images(red, green, blue), the volume of the data need to be processed is huge. As a normal way to simplify the information of the images and improve the speed of progressing images, grey-scale transformation normalized the all 3 channels when the three colours have the same intensity, the grey scale will be obtained, that is, from black(0,0,0) to white(255,255,255). This method does not affect the main information of the image such as outlines, margins, but the colour information is removed.

1.2 Image Filtering

Image noise reduction processing refers to the reserves the original image details characteristics, to suppress image noise reduction of noise of image filtering is the nature of operation: the high-level structure of the segmented image data as the characteristic information of the image recognition to eliminate doped in to digital image noise.

Image processing aims to improve the quality of image and provide the high definition image for the detecting and extracting the fabric defects, on the premise of preserving the original information of the image as much as possible, the noise inside the image is filtered out. The process is called smoothing or filtering, linear filter [10] are mainly adopted to blur the noise of the image through smooth the image. The linear filter means that through the linear operation of the value of every pixel in the input original image to get the value of every pixel of the output image after filtering:

$$g(x, y) = T[f(x, y)] \tag{1.1}$$

Where T is a linear operation.

The type of linear filters [9] usually contains the Mean filter, Box filter, Gaussian filter, median filter and so on.

1.2.1 Mean filter: the mean of the surrounding N x N pixel points substitutes the previous value of the pixel. All the pixel points are traversing by this method, then finish the mean filtering.

For example:

50	97	52
208	276	23
97	88	103

50	97	52
208	110	23
97	88	103

Figure 1.1

New value = (50+97+52+208+**276**+23+97+88+103) / 9 ≈110

1.2.2 Box filter: previous pixel value is substituted by the **mean** or the **sum** (optional) of N x N pixel points surrounding itself.

For example:

Calculating the mean or sum depends on the value of normalize parameter.

Parameter: $\alpha = \begin{cases} \dfrac{1}{\text{width} \cdot height}, & normalize = 1 \\ 1, & normalize = 0 \end{cases}$ The convolution kernel $\mathrm{K} = \dfrac{1}{\alpha}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

Normalize equals 1, then $\mathrm{K} = \dfrac{1}{3 \times 3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

50	97	52
208	276	23
97	88	103

$\Rightarrow$

50	97	52
208	110	23
97	88	103

Figure 1.2

New value = (50+97+52+208+**276**+23+97+88+103) / 9 ≈110

Normalize equals 0, then $K = \frac{1}{1}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

50	97	52
208	276	23
97	88	103

$\Rightarrow$

50	97	52
208	994	23
97	88	103

Figure 1.3

New value = (50+97+52+208+**276**+23+97+88+103) =994

When normalize parameter equals 0, the substituting value could exceeds the range of the pixel value and it would be truncated as the maximum, so that a pure white image will be shown in this position.

1.2.3 Gaussian filter: The weighted average of the surrounding N x N pixel points substitutes the previous value of the pixel. All the pixel points are traversing by this method, then finish the Gaussian filtering. Weight parameter are larger when the pixel is nearer to the centre point.

For example:

$$K=\begin{bmatrix}0.05 & 0.1 & 0.05\\ 0.1 & 0.4 & 0.1\\ 0.05 & 0.1 & 0.05\end{bmatrix}$$

50	97	52
208	276	23
97	88	103

50	97	52
208	167	23
97	88	103

Figure 1.4

New value = (50×0.05+97×0.1+52×0.05+208×0.1+ **276**×0.4 +23×0.1+ 97×0.05+ 88×0.1+ 103×0.05) ≈167

1.2.4 Median filter would arrange the surrounding pixels according to the magnitude of the value, the point is surrounded by N x N pixel points and the median value of them are used to substitute the previous pixel value. All the pixel points are traversing by this method, then finish the Median filter.

For example:

50	98	52
208	276	23
97	88	103

50	98	52
208	97	23
97	88	103

Figure 1.5

New value = [23 50 52 88 **97** 98 103 208 276] = 97

Most of the image noise belongs to Gaussian noise, Based on many experiments, it is concluded that the Gaussian filter is effective to remove the Gaussian noise in production environment according to [11]. Gaussian filtering remove the Gaussian noise, whose principle is using a convolution kernel to calculate with value of each pixel in image and replace the pixel value by the result of convolution kernel and pixel calculation. The probability density function of Gaussian noise follows a normal distribution.

Mathematically speaking, the Gaussian filter of an image is to convolute it with the Gaussian function [12]. The one-dimensional zero-mean Gaussian function:

$$g(x) = e^{(\frac{-x^2}{2\,sigma^2})} \tag{1.2}$$

Sigma is the standard deviation of the Gaussian convolution kernel. Sigma determines the smooth degree of the filtering, bigger the sigma are, smoother the image are.

One-dimensional Gaussian function formula:

$$\mathrm{G}(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} \tag{1.3}$$

Two-dimensional Gaussian function formula:

$$\mathrm{G}(x,\ \mathrm{y}) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1.4}$$

There are some non-negative values in all domains of the Gaussian, which requires an infinite convolution kernel. However, only values within 3 times of standard deviations in the range of mean are needed, all the value in the other parts can be discarded.

1.3 Image Enhancement

Image enhancement mainly overcomes the phenomenon of low contrast caused by the small grey-scale range of the image. The purpose of image enhancement is that amplifying the grey-scale of the input image to the designated degree, so as to make the outline of defects in the image look clearer,

Linear transformation is a common method to enlarge the contrast ratio of the image. The principle of the linear transformation as follow:

Label the input image as *I*, the width as *W* and the height as *H*, and output image is denoted as *O*, the process of the linear transformation is defined by the following formula:

$$\mathrm{O}(r,c) = a \times I(r,c) + b, 0 \le \mathrm{r} < \mathrm{H}, 0 \le \mathrm{c} < \mathrm{W} \tag{1.5}$$

If a=1 and b=0, O as the same as I. On the one hand, value a affects the contract of the figure. When a>1 larger a enhances the contrast of the figure. When 0<a<1, smaller a reduces the contrast of the figure. On the other hand, value b affects the brightness of the figure. When b >0, larger b enhances the brightness. When b < 0, smaller b reduces the brightness.

However,there are different values of a and b adapting to various figure because every image's total brightness are different. So that a linear transformation method automatically selects the values of a and b is applied, which is called Histogram normalization.Histogram normalization assume that the input image is represented as *I*, the high is represented as *H*, the width is represented as *W* and *I (r, c)* represents the grey-scale value of the pixel point at r th row , c th column of the input figure *I*. Mark the Minimum grey-scale value arise in *I* as I_{min}, *and m*ark the Maximum grey-scale value arise in *I* as I_{max}, *i.e.* $I(r, c) \in [Imin, Imax]$, the following mapping relationship between I (r, c) and O (r, c) have to obey to make the range of output image grey-scale value become $O\,[O_{min}, O_{max}]$:

$$O(r,c) = \frac{O_{max} - O_{min}}{I_{max} - I_{min}}(I(r,c) - I_{min}) + O_{min}, 0 \le r < H, 0 \le c < W \tag{1.6}$$

This process is histogram normalization, which is a linear transformation method that automatically selects the values of a and b, where:

$$a = \frac{O_{max} - O_{min}}{I_{max} - I_{min}} \tag{1.7}$$

$$b = O_{min} - \frac{O_{max} - O_{min}}{I_{max} - I_{min}} \times I_{min} \tag{1.8}$$

2. Image Segmentation

Image segmentation [13] is the extraction process certain blocks and important features in an image, which can segment and extract edge information of different objects of the image as well as different lighting changes and colour information according to the original information of the image or its grey-scale value. Image segmentation technique is to divide the pixel matrix of image data into several non-overlapping subsets.The traits of elements in every subset are the same or similar, but the image features of different subsets are obviously different, which is the crucial process of the subsequent analysis of the image.

The available image segmentation methods contains: segmenting according to the region, segmenting according to the edge, segmenting according to the combination of the region and the edge. In recent years, some new method about segmenting image are developed as the invention of many new theory and improvement of the traditional technology: based on mathematical morphology, the fuzzy method, the neural network approach, based on the graph theory, according to the immune method, the granular computing method, etc.

2.1 Threshold segmentation

Threshold segmentation in image segmentation technology [14] is a hot point for a long time, which is simple in calculation and efficient in performance. The general principle of the threshold segmentation is get the continuous boundary region through calculating the input grey-scale values, so that the image is divided into independent regions one another. The threshold segmentation is generally based on the Otus method, which means segment the input image into foreground object and background object by threshold, which is suitable for the image with obvious difference between the grey level range of the target objects and background. The key step of Otus method is that solving the optimal grey threshold, which determines the effect of threshold segmentation.Threshold segmentation formula is shown:

$$g(x) = \begin{cases} 0, & f(x,y) \le T \\ 1, & f(x,y) > T \end{cases} \tag{1.9}$$

Table 1.1 The value of Threshold T

T	Meaning
Constant C	Global threshold segmentation
Variable	Variable threshold segmentation
Relevant for f(x,y)	Dynamic threshold segmentation

The types of threshold segmentation are: binary, anti-binary, cutting segmentation, anti-threshold and zero threshold:

2.1.1 Binary threshold processing

Image binary segment means that all the pixels of the image are set as 0 or 255, in other words, the entire image only contains 2 grey-scale values: black (0) and white (255).

Binary threshold processing would process the original image as the binary image consisting of 2 values, the method to process all the pixels is that: for the value of pixels which higher than threshold, set them to the maximum , then, for the value of pixels which lower than or equal to threshold, set them to 0. In the 8-bit image, the maximum is $2^8 - 1 = 255$, and the rule can be represented as:

$$\mathrm{dst}(x,y)=\begin{cases}\max val, & src(x,y)>thres\\ 0, & src(x,y)\le thres\end{cases} \tag{1.10}$$

2.1.2 Inverted binary threshold processing

Inverted image binary segment means that all the pixels of the image are set as 0 or 255, in other words, the entire image only contains of 2 grey-scale values: black (0) and white (255).
Inverted binary threshold processing would process the original image as the binary image consisting of 2 values, the method to process all the pixels is that: for the value of pixels which higher than threshold, set them to 0 , then, for the value of pixels which equal to or lower than the threshold, set them as maximum. The rule can be represented as:

$$\mathrm{dst}(x,y)=\begin{cases}0, & src(x,y)>thres\\ \max val, & src(x,y)\le thres\end{cases} \tag{1.11}$$

2.1.3 Truncate processing

Truncate processing would set the value of pixels which larger than the threshold as threshold, value of pixels which smaller or equal the threshold would remain unchanged. The rule can be represented as:

$$\mathrm{dst}(x,y)=\begin{cases}\mathrm{thres}, & src(x,y)>thres\\ src(x,y), & src(x,y)\le thres\end{cases} \tag{1.12}$$

2.1.4 Inverted threshold to zero processing

Inverted threshold to zero processing would reset the value of pixels which higher than threshold to 0, the value of pixels which smaller or equal the threshold would remain unchanged. The rule can be represented as:

$$\mathrm{dst}(x,y)=\begin{cases}0, & src(x,y)>thres\\ src(x,y), & src(x,y)\le thres\end{cases} \tag{1.13}$$

2.4.5 Threshold to zero processing

Threshold to zero processing means that the values of the pixels great than the threshold would remain unchanged, meanwhile it would reset the value of pixels which smaller or equal the threshold to 0. The rule can be represented as:

$$\mathrm{dst}(x, y) = \begin{cases} src(x, y), & src(x, y) > thres \\ 0, & src(x, y) \le thres \end{cases} \tag{1.14}$$

According to the official website of OpenCV [15], the visual representations of threshold in different types are shown in Figure 1.6:

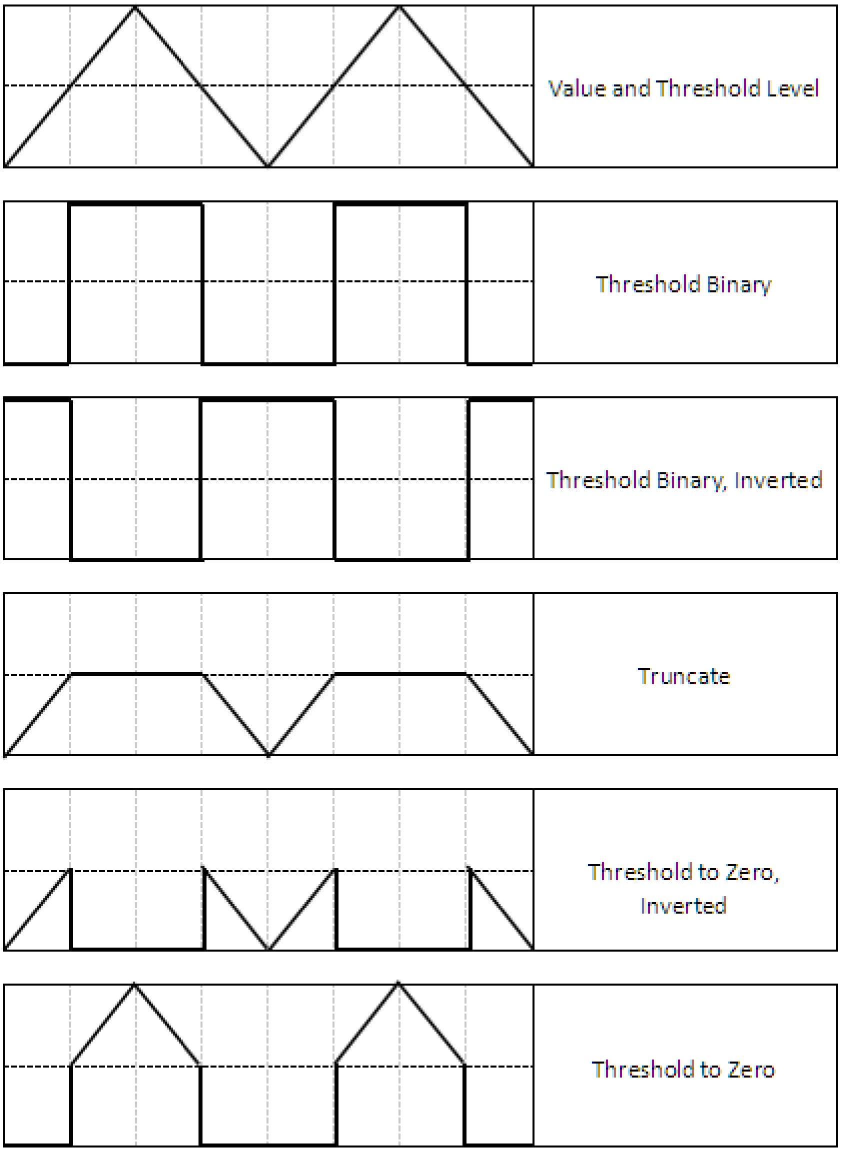


Figure 1.6 [15].

In addition to setting a single threshold, there is a double fixed threshold method. Set the two thresholds ($T_1 < T_2$)for the grey image in advance, then, the grey-scale value of pixels which smaller than T_1 and the grey-scale value of pixels which larger than T_2 would be set to 0, the grey-scale value of the pixel which in the range of T_1 and T_2 would be set to 255.The rule can be represented as:

$$\mathrm{dst}(x,y)=\begin{cases}0, & \mathrm{src}(x,y)<T_1\\ 255, & T_1\le \mathrm{src}(x,y)<T_2\\ 0, & \mathrm{src}(x,y)\ge T_2\end{cases} \tag{1.15}$$

2.2 Mathematical Morphology

Mathematical Morphology is the very important research area in the image process progress.It mainly extract the weight information from the images. The weight information usually have significant meanings to the expression and description of images' shape, which is also the most underlying shape feature when comprehend the image information.

Mathematical Morphology mainly contains: erosion operation, dilation operation, open operation, close operation, morphological gradient operation, tophat operation, blackhat operation and etc. The erosion operation and the dilation operation are the foundations of mathematical morphology operations, other different types of operations are realized by the combinations of that two operations.

2.2.1 Erosion operation

As one of the most fundamental mathematical morphology operations, erosion can eliminate the bound points of the image, so that the image can be inward contract along the boundary, it can also eliminate the parts which smaller then specific structuring elements. In the process of the erosion operation, a structuring element is used to scan the pixels of the image to be processed one by one, then determine the erosion results according to the relationship between the structuring element and the image.

Only when the structuring elements (kernel) totally locate the interior of foreground object, the pixel of the image corresponding to the coordinate of the kernel centre point would be processed to 1, the other value of the image would be processed to 0.

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

(a)

1
1
1

(b)

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

(c)

0	0	0	0	0
0	0	0	0	0
0	1	1	1	0
0	0	0	0	0
0	0	0	0	0

(d)

Figure 1.7 Erosion process

a denotes the image to be eroded.

b denotes the structuring elements (kernel)

The shadow part of the c is the whole 3 possible locations that the kernel totally locate the interior of the foreground object. The centre of the kernels are: image[2,1], image[2,2], image[2,3], respectively.

d is the erosion result, i.e. when the kernel totally locate the interior of the foreground object, the pixel of the image corresponding to the coordinate of the kernel centre point would be processed to 1, when the kernel does not totally locate the interior of the foreground object, the pixel of the image corresponding to the coordinate of the kernel centre point would be processed to 0.

The function erodes the original image, and a specific structuring element is used in the erosion process to decides the shape of a pixel neighbourhood over which the minimum is taken [16]:

$$dst(x,y) = \min_{(x',y'):element(x',y')\neq 0} src(x+x', y+y') \tag{1.16}$$

2.2.2 Dilation operation

Dilation operation is the other basic operation. In opposite to the erosion operation, dilation operation can expand the margin of the images. Dilation operation merge the background points which touch the foreground with the current object so that it realize the outwards expansion of boundary points. If the the distance of objects of image is too close, the objects can connect together. Dilation operation is greatly helpful to full the existing blank of the image causing by image segmentation. As the same as the erosion operation, in the process of the dilation operation, a structuring element is used to scan the pixels of the image to be processed one by one, then determine the dilation results according to the relationship between the structuring element and the image.

If there is any point of the structuring elements (kernel) locates in the foreground image, then the pixels of the image corresponding to the coordinate of kernel centre point would be processed to the foreground colour. If there is no point of the structuring elements (kernel) locates in the foreground image, then the pixels of the image corresponding to the coordinate of kernel centre point would be processed to the background colour.

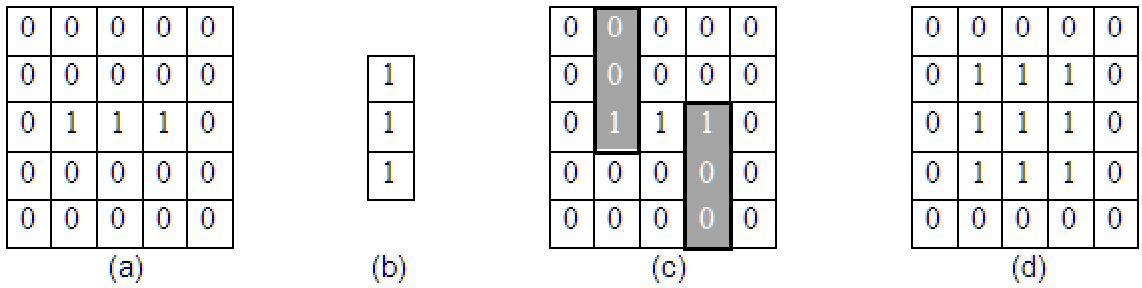


Figure 1.8 Dilation process

a denotes the image to be dilated.

b denotes the structuring elements (kernel)

The shadow part of the c is the 2 possible locations that exist the coincidence pixel between kernel and foreground object when the centre pixel point of the kernel locate the image[1,1], image[3,3]. When the centre of the kernels located at image[1,1], image[1,2], image[1,3], image[2,1], image[2,2], image[2,3], image[3,1], image[3,2], image[3,3], respectively, the pixel points of kernel exist the pixel coinciding the foreground object.

d is the dilate result, i.e. If there is any point of the structuring elements (kernel) coincide the foreground object, the pixel of the image corresponding to the coordinate of the kernel centre point would be processed to 1, If there is no point of the structuring elements (kernel) coincide the foreground object, the pixel of the image corresponding to the coordinate of the kernel centre point would be processed to 0.

The function dilates the original image, and a specific structuring element is used in the dilation process to decides the shape of a pixel neighbourhood over which the maximum is taken [16]:

$$dst(x,y) = \max_{(x',y'):element(x',y')\neq 0} src(x+x',y+y') \tag{1.17}$$

2.2.3 Open operation

The open operation erode the image firstly, then dilate the eroded results.The aim of open operation is applicable for de-noising, counting, etc.

The function of the open operation is shown as:

$$dst = open(src, element) = dilate(erode(src, element)) \tag{1.18}$$

2.2.4 Close operation

The close operation dilate the image firstly, then erode the dilated results. It is helpful for closing the small holes of foreground object, removing the ink dot of the objects and connecting the different foreground objects.

The function of the close operation is shown as:

$$dst = close(src, element) = erode(dilate(src, element) \tag{1.19}$$

2.2.5 Morphological gradient operation

Firstly, dilate and erosion the original image respectively. The morphological gradient operation can be got from the dilation image minus the erode image, the operation can obtain the margin of the source image.

The function of morphological gradient operation is shown as:

$$dst = morph\_grad(src, element) = dilate(src, element) - erode(src, element) \tag{1.20}$$

2.2.6 Tophat operation

Firstly, implementing the open operation to the original image. The tophat operation can be got from the source image minus opened image. The tophat operation can obtain the noise information of the source image, or get the margin information which lighter than the margin of the original image.

The function of the tophat operation is shown as:

$$dst = tophat(src, element) = src - open(src, element) \tag{1.21}$$

2.2.7 Blackhat operation

Firstly, implementing the close operation to the original image. The blackhat operation can be got from the the closed image minus the original image. The blackhat operation can obtain the interior small holes of the original image, the black plot of the foreground object or get the margin information which more dark than the margin of the original image.

The function of the blackhat operation is show as:

$$dst = blackhat(src, element) = close(src, element) - src \tag{1.22}$$

2.2.8 Kernel function

When the processing the morphological operations, a specified kernel (structuring element) should be chosen and used. The kernel function constructs a proper structuring element which can be further process such as erosion, dilation or other morphology operations.

3. Defects Extraction

The extraction of defect features [17] can be made by using conventional statistical features, variance of grey value, strength of grey value, texture information roughness and aspect ratio, etc. In image information, objects are measured by a certain scale, and the initial scale of the image is transformed to get the locational information of the image at various scales. The edges detection of these information is carried out, and these contour edges are used as feature vectors.

Canny edges detection is a kind of a method of edge detection using multilevel edges detection algorithm, the goal of Canny is that creating the best edge detection algorithm which can identify as many real edges of an image as possible, then identified the position of edges as close as possible, meanwhile all the edges of the image could only be identified once, and noise information in the figure will not be identified and detected as edges. In conclude is that the Canny detects and accurately position the defects while make no mistake.

According to John F.Canny [18], the Canny detection is mainly consisted by these four steps: Firstly, de-noising. Noise influence the precise of edge detection, therefore, firstly, noise should be filtered, then calculate the magnitude and direction of the gradient, the next step is that through Non-maximum suppression (NMS) to remove the spurious response to edge detection, finally, determine the edge by the dual-threshold algorithm.

3.1 Applying the Gaussian blur to remove the noise in the image

On account of the detection of image edge is much easier to be interfered by noise, therefore, the original image usually need to be filtered to avoid detecting the wrong edge information. The goal of the filtering is smoothing some weak texture non-edge areas, so that a much more precise edge can be detected and got.

In the process of filtering, the weight average of pixel value around the pixel point is calculated to obtain the final filter result. For the Gaussian filter, closer to the centre the point is, larger the weight value is. The kernel of the Gaussian filter is variable, and the size of the kernel intensely impact on the edge detection. Larger the kernel is, the less sensitive the information is to the noise, however, more the locating mistakes of edge detection is. Usually a kernel with 5 X 5 size can meet the most situations.
For instance, a 5 X 5 Gaussian filter (σ =1) is shown as following:

$$K = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \times A \tag{1.23}$$

3.2 Calculating gradient

Magnitude of the gradient should be calculated, which is the speed of the change of image. For the margin part of the image, the grey value change and the magnitude of the gradient is larger; in the opposite, for the relatively smoothed part of the image, the grey value change and the magnitude of the gradient is smaller. In the normal case, the magnitude of the gradient means to calculate the margin information of the image.

Usually derivation is used to get the magnitude of the gradient, however the approximate value of the gradient can be got through calculate the difference of pixel values. The Sobel operator are mentioned which is used by Canny edge detection.

The Sobel operator is a kind of discrete and differential operator, which consists of the Gaussian smooth and differential derivative operation. It find the edge taking the acknowledge of partial difference, and the result it got is the approximately value of gradient. The Sobel operator is also called Sobel filter.

Calculating the convolution of Sobel operator and the original image can calculate the approximately values of the horizontal and vertical partial derivative.

SRC is the original image, if the SRC have 9 pixels:

The horizontal partial derivative G_x :

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \bullet src = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} P1 & P2 & P3 \\ P4 & P5 & P6 \\ P7 & P8 & P9 \end{bmatrix}$$

To calculate the P5 horizontal partial derivative P5x:

$$P5x = (P3 - P1) + 2 \bullet (P6 - P4) + (P9 - P7) \tag{1.24}$$

i.e. The pixel values on the right of P5 minus the pixel values on the left of P5, in addition, mediate pixels P4 and P6 are closer than other 4 pixels to P5, so that the weight of the difference of pixel value of P6 and P4 is 2 while the weight of the difference of pixel values of P3 and P1, P9 and P7 is 1.

The vertical partial derivative G_y :

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \bullet src = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \bullet \begin{bmatrix} P1 & P2 & P3 \\ P4 & P5 & P6 \\ P7 & P8 & P9 \end{bmatrix}$$

To calculate the P5 horizontal partial derivative P5x:

$$P5y = (P7 - P1) + 2 \bullet (P8 - P2) + (P9 - P3) \tag{1.25}$$

i.e. The pixel values on the right of P5 minus the pixel values on the left of P5, in addition, mediate pixels P8 and P2 are closer than other 4 pixels to P5, so that the weight of the difference of pixel value of P8 and P2 is 2 while the weight of the difference of pixel values of P7and P1,P9 and P3 is 1.

Then the direction of the gradient is calculated, which is orthogonal to the direction of the margin. Edge detection operator returns the Gx (horizontal magnitude) and Gy (vertical magnitude). The magnitude (G) of the gradient and direction (Θ, represented by angel) of the gradient can be calculated as following:

$$G = \sqrt{G_x^2 + G_y^2} \tag{1.26}$$

$$\theta = \operatorname{atan2}(Gy, Gx) \tag{1.27}$$

Where atan2(•) represents arctan function which contain 2 parameters.

The direction of gradient is always orthogonal to that of margin, and usually proximately value the direction as horizontal (left and right) , vertical (top and bottom) , diagonal (top left, bottom left, top right and bottom right), total 8 directions.

3.3 Non-maximum suppression

After obtained the magnitude of the gradient and direction of the gradient, all the pixel points of image are traversed, and all the non-edge points are removed. Judge the pixel point whether is the maximum of the gradient around the pixel points which have the same direction of the gradient or not. If it is the partial maximum in the positive/ negative direction then retain the value of this point, or else, restrain this point (return the value of this point to zero).

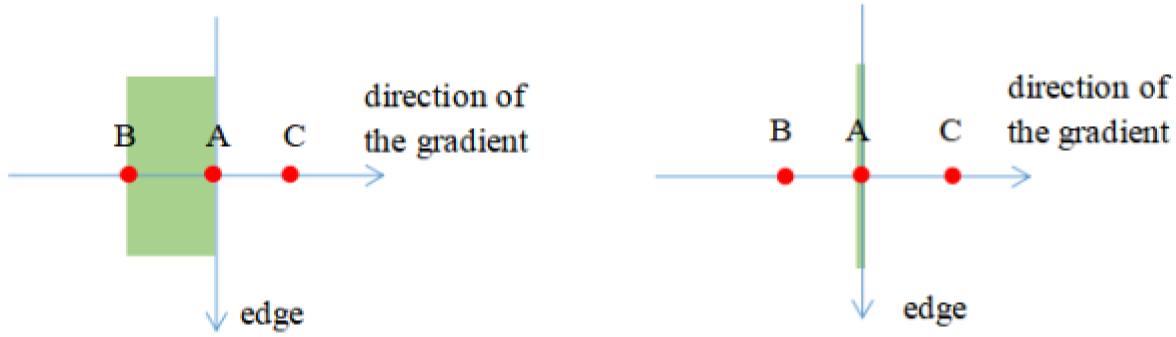


Figure 1.9 Non-maximum Suppression example

As shown in Figure 1.9, A, B, C these points have the same direction which is vertical the edge, and according to the comparison and judgement, point A have the partial maximum, so retain point A and restrain point B, C (return the value of point B, C to zero).

Through the above process,basically only retain one point of the several edge points in the same direction, so that it thin the edge.

3.4 Dual-threshold algorithm to determine edge

Complete the above steps, remaining pixel of edges provide a more precise contours of edges in an image, the strong edge of the image is contained in the current edge image. However, some virtual edges is also contained in the edge image. The virtual edges can be produced by the real image or produced by the noise. For the latter, it must be removed. This is realized by selecting a high threshold and a low threshold.

Set two thresholds, one is higher threshold maxVal, the other is lower threshold minVal. According to the relationship between the magnitude of the gradient of current edge pixels and these two thresholds, to judge the property of the edges. The specific steps as following:

a. If the magnitude of the gradient of current edge pixels larger than or equal to the maxVal, then mark the current edge pixels as strong edge.

b. If the magnitude of the gradient of current edge pixels larger than the minVal and smaller than the maxVal, then mark the current edge pixels as virtual edge. (need to be retained)

c. If the magnitude of the gradient of current edge pixels small than the minVal, then restrain the current edge pixel (return the value of this point to zero).

The virtual edge obtaining from the above steps need to be further processed. Normally, according to whether the virtual edge and the strong edge are linked to determine the virtual edge belonging to which situation. If the virtual edge and the strong edge are continuous, then process the virtual edge as real edge, on the contrary, if the virtual edge and the strong edge are discrete, then process the virtual edge as weak edge and restrain the edge.

In the left of the Figure 1.10 displays the three edge messages, in the right of the Figure 1.10 is the sketch map classifying the edge information.

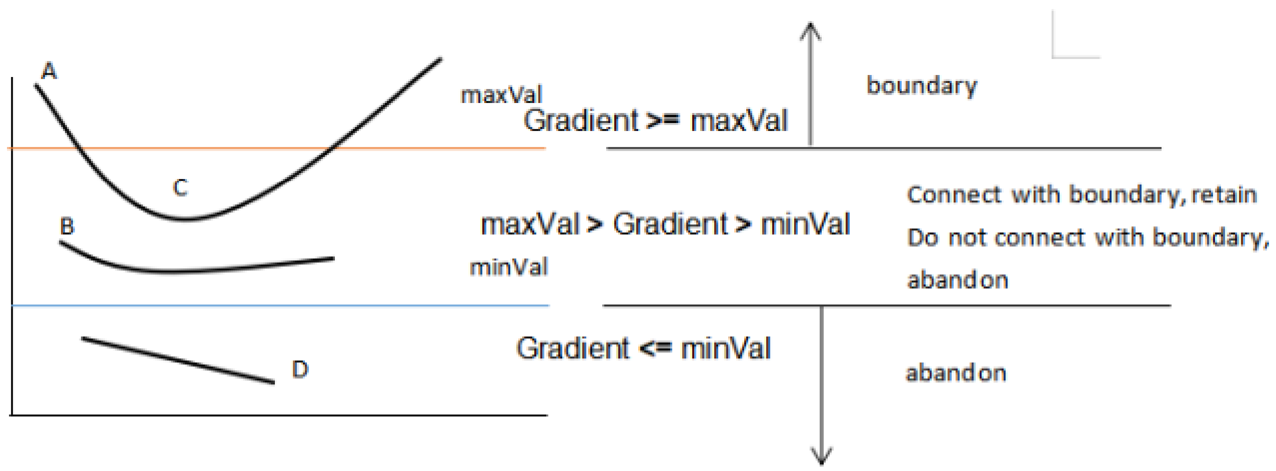


Figure 1.10 Edge Classify

The magnitude of the gradient of point A greater than maxVal, so point A is the strong edge. Both the magnitude of the gradient of point B and that of point C are between maxVal and minVal, so point B and point C is the virtual edge. The magnitude of the gradient of point D less than minVal, so point D should be restrain (abandon).

Figure 1.11 displays the process result of the virtual edge point B and point C in the Figure 1.10. The magnitude of the gradient of point B is between maxVal and minVal, which is virtual edge, but the point do not connect with strong edge, so point B should be restrain (abandon). The magnitude of the gradient of point C is between maxVal and minVal, which is virtual edge, and it connect wit strong edge point A, so so point C should be retain.

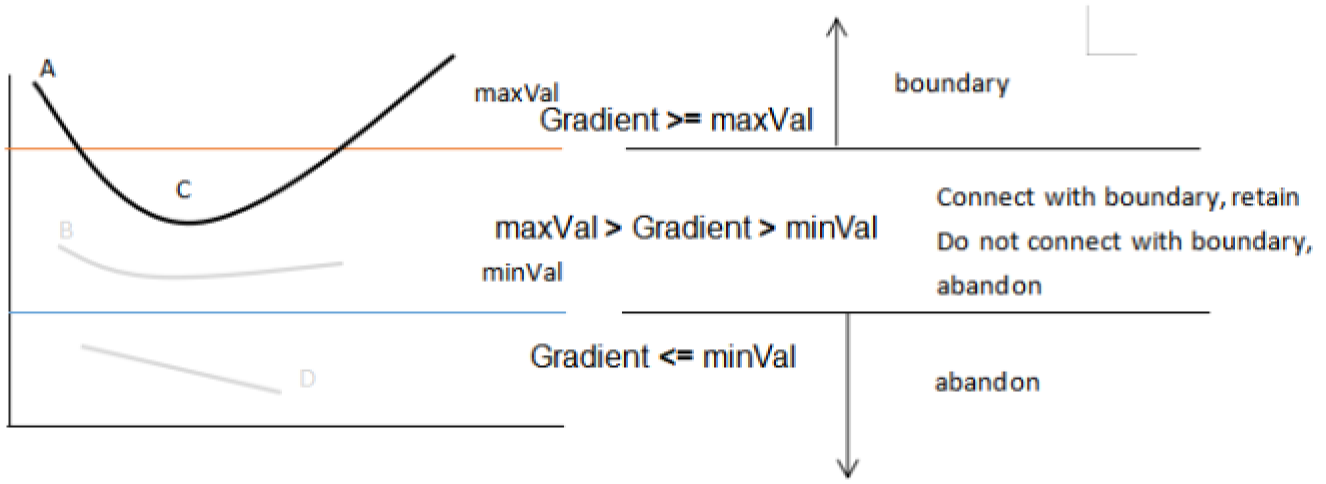


Figure 1.11 Process Result of Edge Classify

The higher threshold maxVal and lower threshold minVal are not fixed, need to be defined pointing to different images.

4. Chapter Summary

This chapter starts from image preprocessing to defects extraction, and briefly introduces the image processing threshold segmentation, defect feature extraction and the relevant machine vision technology used for defect identification, which lays a reliable technical theoretical basis of the frame design and detailed realization of the subsequent system.

Chapter 2: Framework Design of Detection System and Implementation

The fabric defect detection system based on computer vision is mainly composed of three functional modules: machine vision module, image detection, feature extraction. The software functions of the overall system adopts the idea of modular design, and the coupling degree between modules is small. Each module uses multi-threading technology to realize, the operation of each module is relatively independent.

1. Design Requirements

The purpose of this research is to design a cloth defects detection system which is easy to use and contains machine vision technology. For cloth fabric users to provide efficient, intelligent, real-time defect detection system, for the subsequent production and processing of the first quality.

The functions of the detection system need to achieve the following design objectives:

(1) pre-processing the cloth image: image smoothing, linear filtering, morphological processing;

(2) threshold processing was carried out on the image after noise reduction to segment the suspicious cloth defect image;According to the different cloth texture to set the appropriate pixel grey value to segment the image;

(3) extract the credible defect features from the segmented grey value image, and calculate the length and area values of the features;

(4) image preprocessing, image segmentation and defect feature extraction of the whole detection system can work in parallel with each functional module, and the detection of cloth defects can be completed in real time.

2. Overall Architecture

The fabric defects detection system in this project runs on the Windows platform, and the open source computer vision library (OpenCV) is used. OpenCV's license is BSD, which is free to be used in academic research and commercial products of enterprises. Support high-level language C++, Java, Python direct call, support PC Windows, Linux, Mac OS and mobile phone iOS system and Android system. OpenCV is designed to improve the efficiency of graphics computing, focus on real-time process.

Written in optimized C/C++, OpenCV is able to multi-core process. In OpenCV, a common computer graphics algorithm is implemented, which is a very practical graphics development framework.

The detection system's software architecture is shown as Figure 2.1:

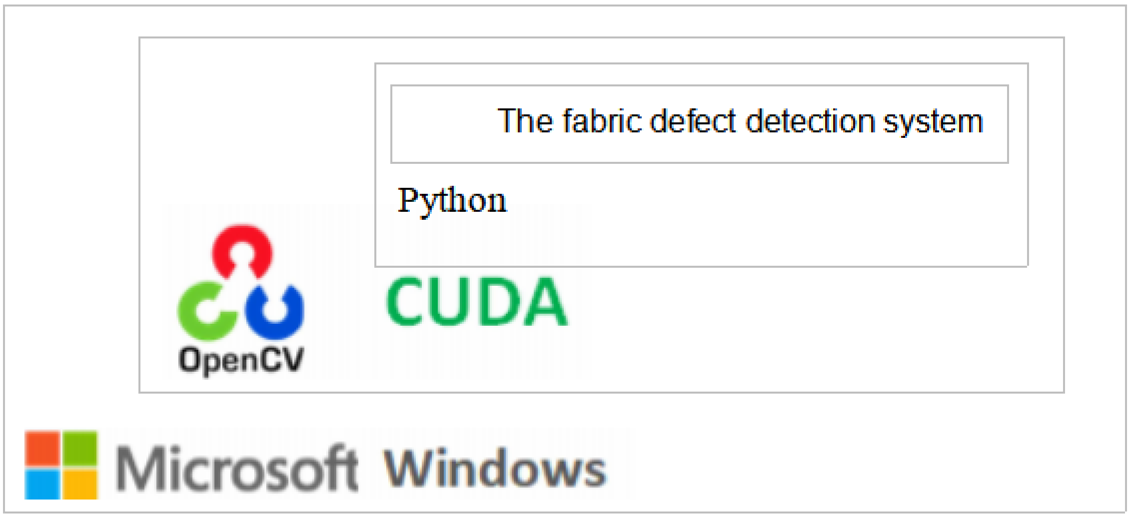


Figure 2.1 Software Architecture of the Detection System

The detection module design based on the design requirements and software architecture. The system adopts the design idea of modularization, and task scheduling and collaboration between modules are carried out by message teams. The main modules of the system include image preprocessing module, image segmentation module and defect feature extraction module.

The division of modules is shown in Figure 2.2:

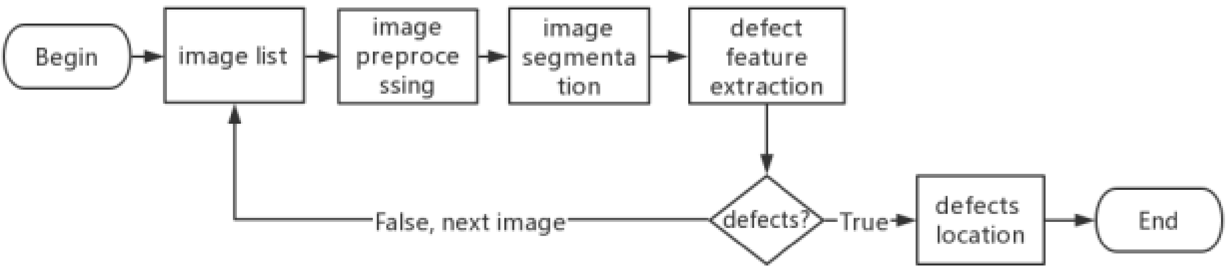


Figure 2.2 Overall Architecture

The defect detection system takes a frame of image from the head of the queue of images to improve the image quality and make it clearer and easier to be recognized by the machine. The image signal is processed by Gaussian filter to suppress the noise which is normally distributed. The mean value and variance of the image are calculated, and the confidence threshold interval of the normal distribution required for image segmentation is set. Canny operator is used to detect the edges of the image and the topological structure analysis algorithm of the digital binary image followed by the boundary is used to extract the defect contour. Finally, the detected defects are marked in the image.

3. Processing Module

Under the influence of the illumination of the real environment and the quality of the camera encoding and decoding the video, it is necessary to suppress the noise in the cloth image under the condition of retaining the details of the original image, that is, image noise reduction.The essence of image noise reduction is to carry out filtering operation on the image: the high-level structure data of the image is segmented as the feature information and the noise doped during image digitization is eliminated.

The linear smoothing filter - Gaussian filter is used to implement the linear smoothing process of the image by filter with size of 3 x 3. It is used to calculate the weighted average of the whole pixels in the image, which are calculated by the weighted average of all the pixels in the adjacent field. The calculated results are stored in the same location in the target image. The size of the result image is the same as the size of the original image.

The main functions of this module are as follows:

After image grey-scale transformation is carried out to improve image contrast, more details can be displayed to improve image quality. In addition, features of interest in image defect detection can be selectively highlighted to suppress unwanted image feature information. Gaussian filter is used to smooth the image. The standard deviation of Gaussian has important influence on the effect of Gaussian filtering. The Gaussian filter's frequency band width is set to the size of σ . Wider the Gaussian filter's the frequency band is, smoother the image is, and the standard deviation and smoothing ability of the Gaussian filter are very strong. Therefore, the actual detection process should be dynamically configured.Parameter to improve the suppression ability of image noise.

The process of image processing module is shown in figure 2.3:

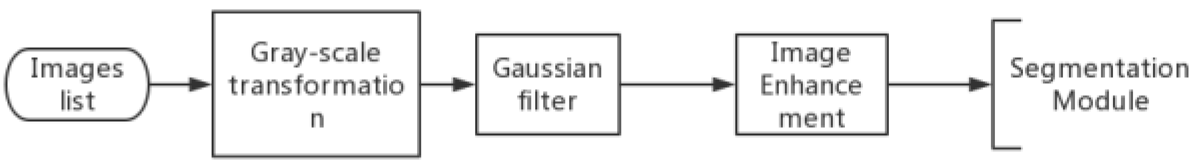


Figure 2.3 The Main Process of Image Processing

Grey-scale transformation: the image is transformed into grey scale, then the image is de-noised by linear filtering. A 3 x 3 kernel is used to convolve with the pixels in the image matrix, and the weighted average grey-value of the pixels surrounding itself determined by the kernel is used to substitute the value of pixels in the convolution centre.

In the BGR colour space, all the ranges of the 3 channel values are from 0 to 255, when a BGR format image is converted to grey-scaled image, the grey-scale values are calculated as follow:

$$\mathrm{dst(I)} = 0.114 \times \mathrm{src(I).B} + 0.587 \times \mathrm{src(I).G} + 0.299 \times \mathrm{src(I).R} \tag{2.1}$$

Gaussian filtering: the kernel of the Gaussian format is convolved with the pixels of the original image to filter out Gaussian noise. Original image can be a single channel or three channels.Each channel is processed independently, and its pixel values can be unsigned 8-bit, 16-bit, single-precision 16-bit, or floating point 32-bit and 64-bit.The Gaussian kernel used for convolution may be different in length and width, but they must be non-negative positive odd Numbers;If the width of the Gaussian kernel is set to 0, the standard deviation in horizontal and vertical directions of the Gaussian kernel vectors is calculated.

The equation for a (2k+1)×(2k+1) size Gaussian filter kernel of is shown as:

$$\mathrm{H_{ij}} = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(i-(k+1))^2+(j-(k+1))^2}{2\sigma^2}\right); 1 \le i, j \le (2k+1) \tag{2.2}$$

The code of generating convolution kernel of filtering and the code of Gaussian filtering is shown as follow:

```
def getGaussKernel(sigma,H,W):

        gaussMatrix = np.zeros([H,W], np.float32)
        cH = (H-1)/2
        cW = (W-1)/2

        for r in range(H):
                for c in range(W):
                        norm2 = math.pow(r-cH,2) + math.pow(c-cW,2)
                        gaussMatrix[r][c] =math.exp(-
norm2/(2*math.pow(sigma,2)))

        sumGM = np.sum(gaussMatrix)
        gaussKernel = gaussMatrix/sumGM
        return gaussKernel
```

```
def gaussBlur(image, sigma, H, W, _boundary ='fill', _fillvalue = 0):

        gaussKernel_x = cv2.getGaussianKernel(sigma, W, cv2.CV_64F)
        gaussKernel_x = np.transpose(gaussKernel_x)
        gaussBlur_x = signal.convolve2d(image, gaussKernel_x, mode = 'same',
boundary = _boundary, fillvalue = _fillvalue)

        gaussKernel_y = cv2.getGaussianKernel(sigma, H, cv2.CV_64F)
        gaussBlur_xy = signal.convolve2d(gaussBlur_x, gaussKernel_y, mode =
'same', boundary = _boundary, fillvalue = _fillvalue)

        return gaussBlur_xy
```

The method of enhance the images is histogram normalization, which is map the range of the pixel value of the input image in the range of [0,255]. Suppose the distribution range of pixel value of input image is Input: [min, Max], and the range after mapping is Output: [0,255].

scale is normalized then the input array elements are shifted, the formula is shown as:

$$dst = alpha \times \frac{src(r,c) - src_{min}}{src_{max} - src_{min}} + beta \tag{2.3}$$

Where src represents the input image matrix, dst represents output image matrix as the same size as src, alpha represents norm value or lower range boundary get by the range normalization and beta is not used for the norm normalization represents upper range boundary of the range normalization.

The functions [19], [20] used in image processing module is show in Table 2.1:

Table 2.1 The functions used in image processing module

Function	Illustrate
cvtColor( )	The function of transform the colour space. The parameter-COLOR_BGR2GRAY: A BGR format image is converted to grey-scaled image
GaussianBlur( )	Smooth an image by the a Gaussian filter. The input image convolves with the appointed Gaussian kernel. The function supports in-place filtering.

zeros( )	Creates an array in the specified size and type, whose all the elements are zero.
normalize( )	All the norm values in the array are normalized in a specified range.

The aim of function zeros( ) is to create void array which have the same rows and columns with the input image, so that the function normalize( ) can output the enhanced image with the same size in the void array.

4. Segmentation Module

The defect segmentation module is the key module of the system. The defect segmentation module includes image grey threshold segmentation, mathematical morphology, etc. the pre-processed images were taken from the queue of images to be examined, and the suspicious defect areas were segmented by the parallel region technology grey threshold segmentation method.This method takes advantage of the grey difference between the suspicious defects (specified objects) and the background, which is pixel level segmentation.The threshold value is selected by calculating the confidence interval of normal distribution. The mathematical morphology methods are used to removes these noise and extract the weight information from the images.

The processes of defect segmentation is shown in Figure 2.4:

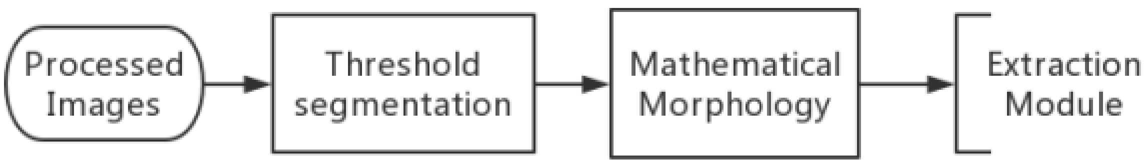


Figure 2.4 The Main Process of Defect Segmentation

In order to extract the defect feature value, the threshold segmentation of the original image should be carried out first, that is, the defect image should be separated from the background image. The key of image threshold segmentation is threshold selection.

Most monochrome fabric images have one characteristic: the grey value of the fabric image almost falls in a small range of grey value, only a part of the cloth due to the texture itself, light conditions and acquisition equipment and other influences fall outside the normal range of grey value. The grey values of the defect image must fall within the range of the normal grey value, so the defect image can be separated from the background image by finding this range and finding the upper and lower thresholds of the range. The confidence interval can be obtained by the method of normal distribution.

The confidence interval method for the normal distribution of the fabric image is to take each pixel on the fabric image as a sample and the grey value of the pixel as a sample value. Firstly, calculate the normal distribution, then the average grey value and standard deviation of pixels are calculated. The upper threshold is: average grey value +3 * standard deviation. The lower threshold is: average grey value -3 * standard deviation. The confidence interval is [lower threshold, upper threshold]. The advantage of the normal distribution method is that the influence of noise on threshold selection can be greatly reduced [20].

The code of dual threshold segmentation is shown as follow:

```
##average and std
(mean , stddv) = cv2.meanStdDev(dst)
##thresholds
Maxthr = mean + 3*stddv
Minthr = mean - 3*stddv
t1, lowthres = cv2.threshold(dst , Minthr, 255,
cv2.THRESH_BINARY)

t2, maxthres = cv2.threshold(dst, Maxthr, 255,
cv2.THRESH_BINARY_INV)

thres = cv2.bitwise_and(lowthres, maxthres)
```

After the noise point processing, there are still some isolated points in the defect image.Most of these isolated spots are small defects in cloth. The idea of dealing with these isolated points is: firstly, the isolated points which are smaller than a fixed size are removed as noise points, where the open and erosion morphological operations are used; then the larger isolated points are then connected as defects with the neighbouring ones, where the close and dilation morphological operations are used.

The functions [22]-[25] used in defect segmentation module is show in Table 2.2:

Table 2.2 The functions used in defect segmentation module

Function	Illustrate
meanStdDev( )	Get matrix's average value and standard deviation of all the elements.
threshold( )	A fixed threshold is used to every matrix element to segment the image.
	A fixed threshold are used to segment the matrix. The function filtering out

	the pixels whose value is smaller or larger the a fixed value. The function usually used to remove a noise and get binary images out of a grey-scale image. The function support the most types of threshold by set different parameter.
THRESH_BINARY	The parameter to determine the type of the threshold is binary threshold processing.
THRESH_BINARY_INV	The parameter to determine the type of the threshold is inverted binary threshold processing.
bitwise_and( )	Calculate per-element bitwise logical AND between the two arrays (or an array and a scalar). i.e. result = array1 & array2
ones( )	Creates an array in the specified size and type, whose all the elements are one.
dilate( )	A specific structuring element created by function ones(), then it is used to dilate the input image .
erode( )	A specific structuring element created by function ones(), then it is used to erode the input image .
morphologyEx( )	Implement more complex morphological operations. The function realize the more complex morphological operations by combining the basically erosion operation,dilation operation and some simple operations . Through independent processing the every channel to realized the morphological operation for the image composed of multi-channel.
arcLength( )	The function returns the result of the perimeter of a contour.
contourArea( )	The function returns the result of the area of a contour.

In the threshold segmentation process: The aim of function meanStdDev( ) is to get the 2 parameters of the image: the mean value and the standard deviation so that the lower threshold and upper threshold can be determine. THRESH_BINARY and THRESH_BINARY_INV are two specific parameters of the function threshold( ) which determine the type of the threshold. THRESH_BINARY means binary threshold processing and it is used to segment image with lower threshold. THRESH_BINARY_INV means inverted binary threshold processing and it is used to segment image with upper threshold. The function bitwise_and ( ) adds the two result get from binary threshold processing and inverted binary threshold processing so that the final segmentation result has been calculated. The threshold is set to normal distribution confidence interval. After the image is processed, the amount of useless data is greatly reduced, so as to highlight the image contour required.

In the mathematical morphology process: The function ones( ) is used to create the kernel of all kinds of the morphological operations, the function morphologyEx( ) is used to achieve the complex morphological process, 2 specific parameters MORPH_OPEN and MORPH_CLOSE are used to implement open operation (remove the discrete noise points) and close operation (connect the discrete defects). After noise processing, there are still some isolated points in the image.Most of these isolated spots are small fabric defects. Call the functions to calculate every isolated point's perimeter (function arcLength( )) and area (contourArea( )), then remove the noise according to the perimeter and area.

5. Extraction Module

The defect detection thread takes a frame of image from the pre-processed image queue, first obtains the trusted normal surface area of the threshold value, then performs threshold segmentation, and then detects the defect edge. If there is an edge contour, defect contour information is obtained and the defect contour are marked in the image.

The main process of defect extraction is shown in Figure 2.5:

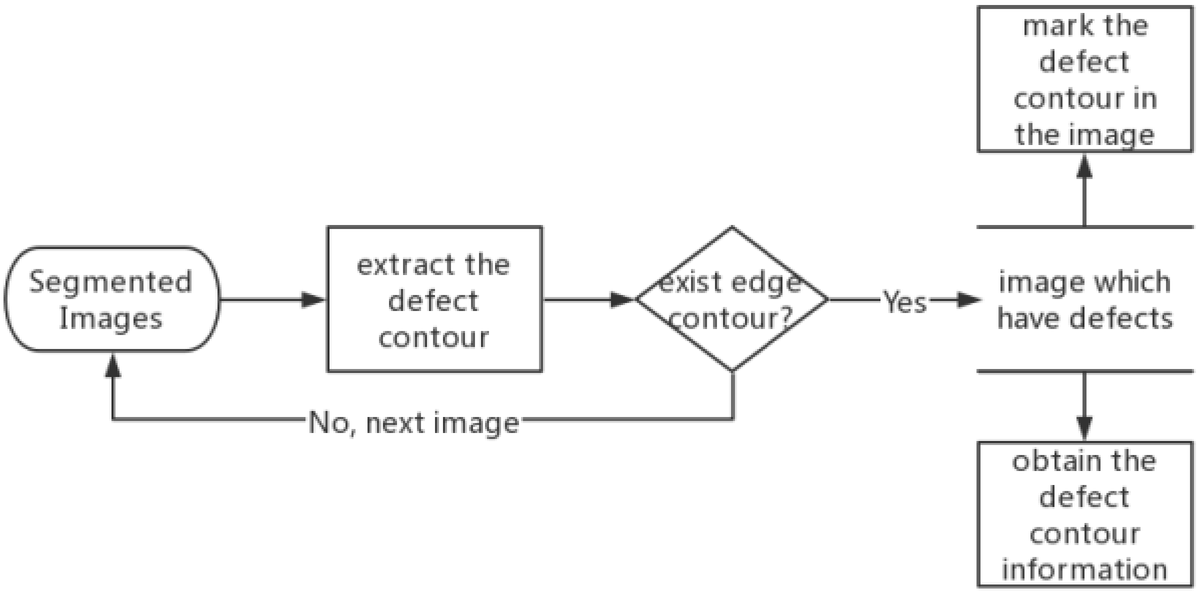


Figure 2.5 The Main Process of Defect Extraction

Canny find the pixels in the image with large changes in grey value. If there are not any pixel with large changes, the image does not exist defect, however, if the pixels with large changes exist, connect them together and these connected lines are the edges of the defect. The defect contour information would be obtain and it will be marked in the image.

The functions used in defect extraction module is show in Table 2.3:

Table 2.3 The functions used in defect extraction module

Function	Illustrate
Canny( )	Through Canny algorithm [18] to detect all the edges in the image . Through the principle of the Canny algorithm, this function retrieves all the edges in the source image then marks the edges in the output map image. The Canny algorithm need 2 thresholds. The smaller threshold is used to connect the edges. The largest threshold is used to retrieve strong edges.
findContours( )	Finds contours of a binary image. The function's source image requires a binary image, then returns the information of all the contours of the image[26]. The contours are a common and helpful way to analysis shape and detect and recognise object.
boundingRect( )	Returns the information that the minimum bounding rectangle which contains the contour in a grey-scale image.
drawContours( )	Returns and draw the outlines of contours (thickness≥0) or filled contours (thickness<0).

The function canny( ) is used to detect the edge contours of the defects, and if contours exist, their information will be obtain by the function findContours( ), and the defects are marked in the image using the minimal up-right integer rectangle of its own contour.

6. Chapter Summary

This chapter mainly introduces the overall architecture of the detection system, general design the image preprocessing module, defect segmentation module, defect feature information extraction module in the system as well as the interaction mechanism among the modules. Under the framework design of the detection system, the image preprocessing, defect detection and defect feature information extraction modules are designed and realized in detail and realized by using python in combination with the open source machine vision library. Part of the code in the modules is shown to explain and display the way of implementation.

Chapter 3: Test results and analysis

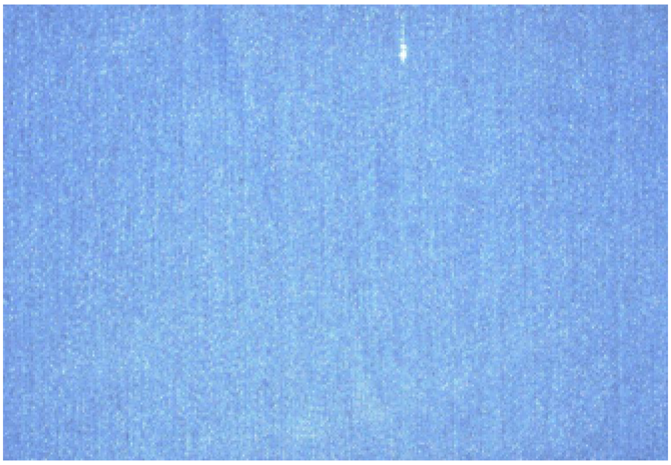
The operating environment of the system consists of image processing, defect detection and defect extraction software services. The detection software runs on a Windows flat 64-bit system, which is written in python and requires open source machine vision library (OpenCV) support.

All the figure provided by 2019 guangdong industry innovation competition. Fabric images in Nanhai textile workshop of Foshan were collected, large-scale data sets of high quality fabric defects were made and released, meanwhile the fine annotation were provided.

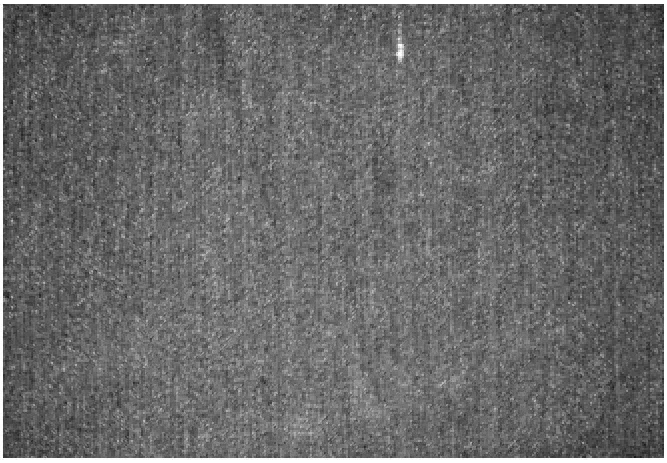
1. Single Defect

Several common defects of fabric are detected as follow:

Hole defect: the hole formed in the warp and weft yarn is a kind of defect easily detected by machine vision.The detection steps of such defects are as follows: grey transformation of the original image, Gaussian smoothing and finally test results. The figure of is shown:



Original image of the Broken hole



grey-scale transformation and Gaussian smoothing

Threshold segmentation

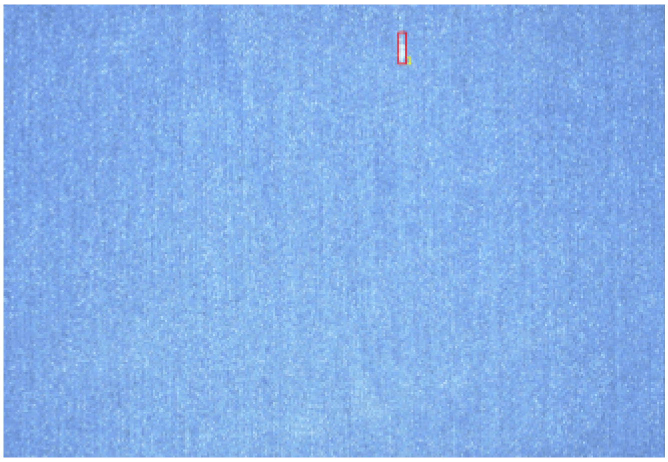


Image marked the defect

00a07f82f195265d0953081713.jpg:

Defect 1
Length: 155
Area: 584
[[[867 61]]

[[883 61]]

[[883 130]]

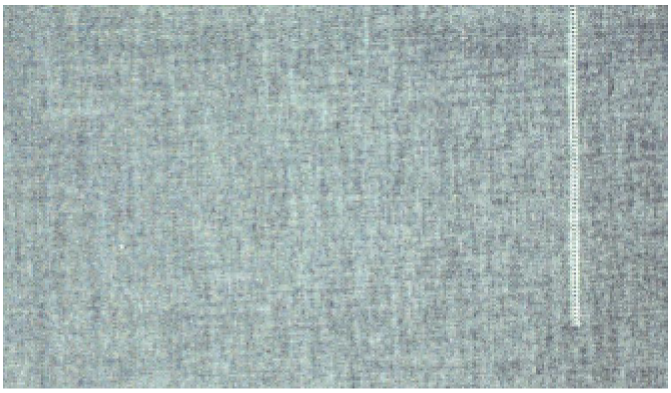
[[867 130]]]

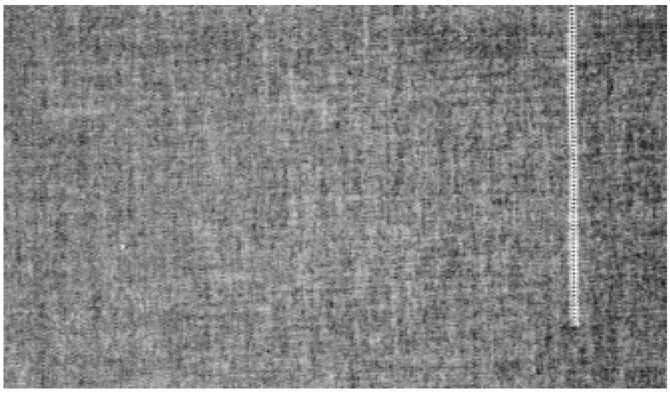
The defect contour information

Figure 3.1 Figure of Hole Defect Detection

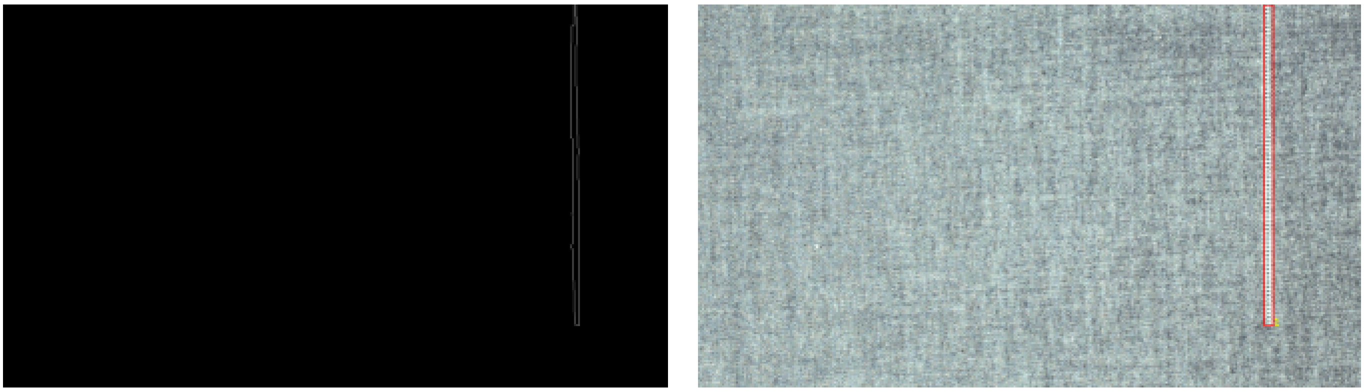
The defect contour information of the figure is save in the .txt file as shown, which consist of the location of all the contour points.

Warp missing: a piece of fabric or a piece of warp is missing because the warp is broken during weaving and not handled in time.The figure of is shown:





Original image of the Warp missing　　　　grey-scale transformation and Gaussian smoothing



Threshold segmentation　　　　Image marked the defect

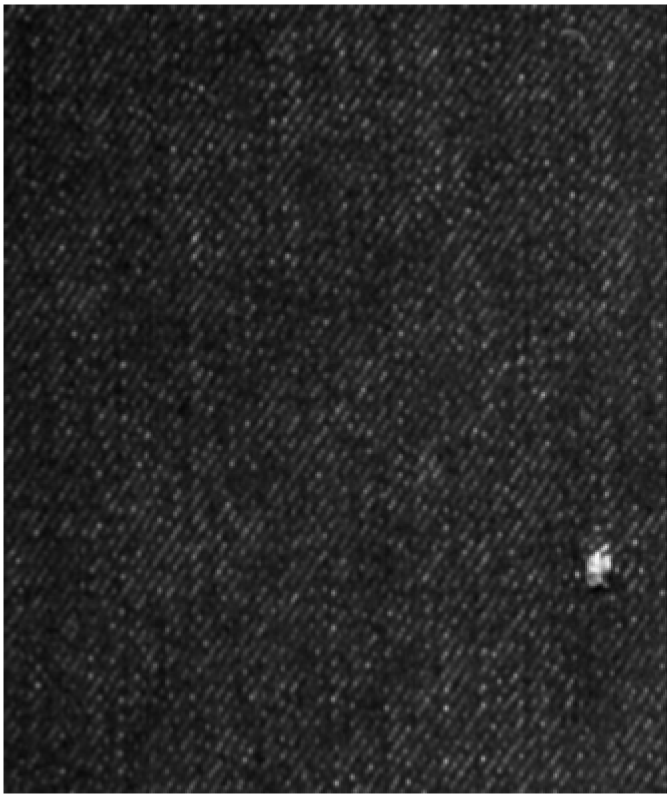
```
00c145d6979084d21323312592.jpg:

Defect     1
Length:    1406
Area:      6826
[[[1210    0]]

 [[1229    0]]

 [[1229  685]]

 [[1210  685]]]
```

The defect contour information

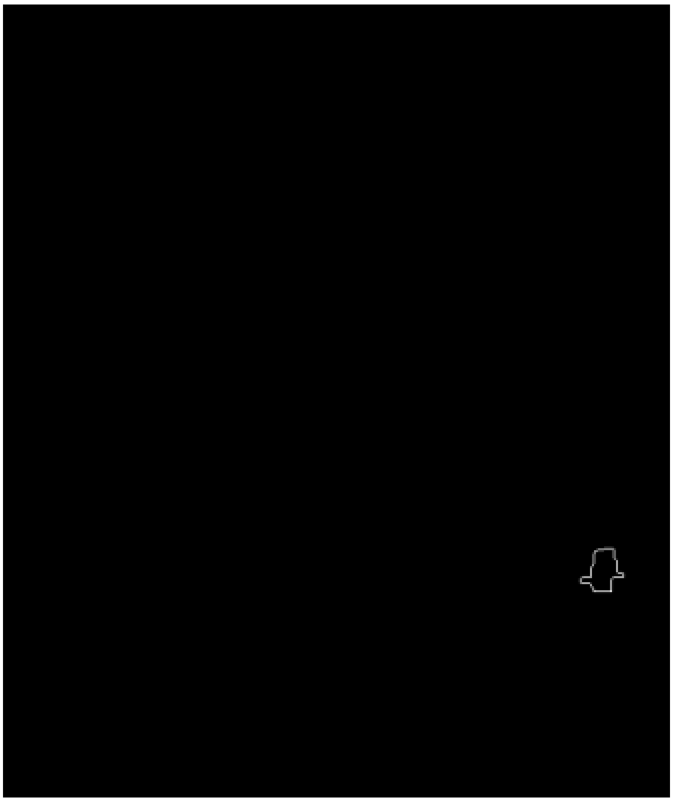
Figure 3.2 Figure of Warp Missing Detection

Spot defect: cloth surface or yarn with oil, colour, etc.These defects usually appear as dots and stars.The figure of is shown:

Original image of the Spot defect



grey-scale transformation and Gaussian smoothing



Threshold segmentation

Image marked the defect

1bf0ece2a497dc170802285273.jpg:

Defect 1
Length: 118
Area: 512
[[[430 405]]

[[462 405]]

[[462 437]]

[[430 437]]]

The defect contour information

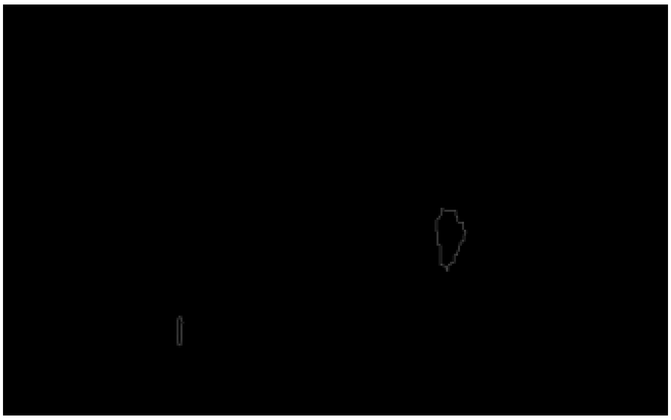
Figure 3.3 Figure of Spot Defect Detection

Adhesive: any other fabric that adheres to the surface of a piece of cloth during finishing or transportation.

The figure of is shown:

Original image of the Adhesive fabric

grey-scale transformation and Gaussian smoothing



Threshold segmentation

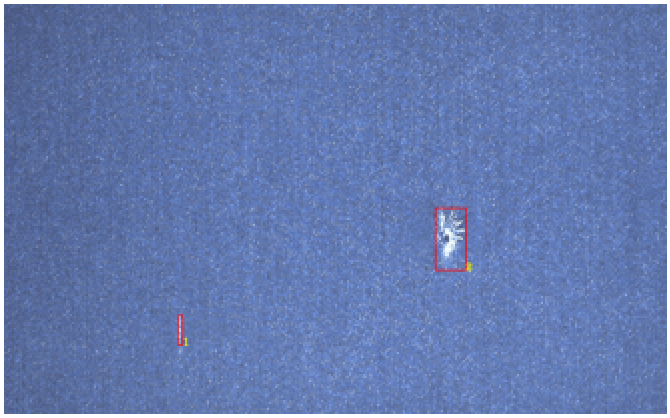


Image marked the defect

03bdb2d5d2f2183d0919134502.jpg:

Defect 1
Length: 151
Area: 501
[[[420 748]]

[[431 748]]

[[431 818]]

[[420 818]]]

Defect 2
Length: 421
Area: 6490
[[[1038 490]]

[[1111 490]]

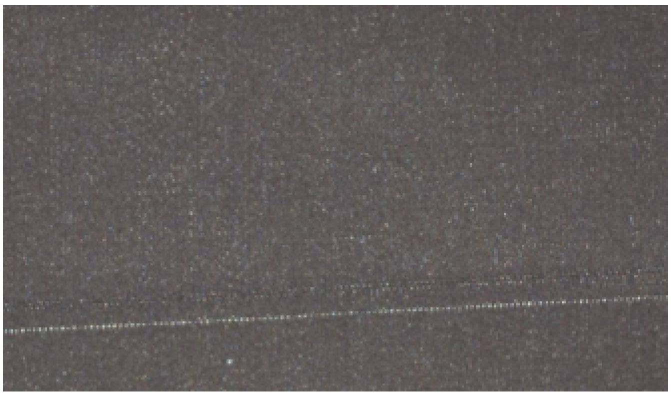
[[1111 639]]

[[1038 639]]]

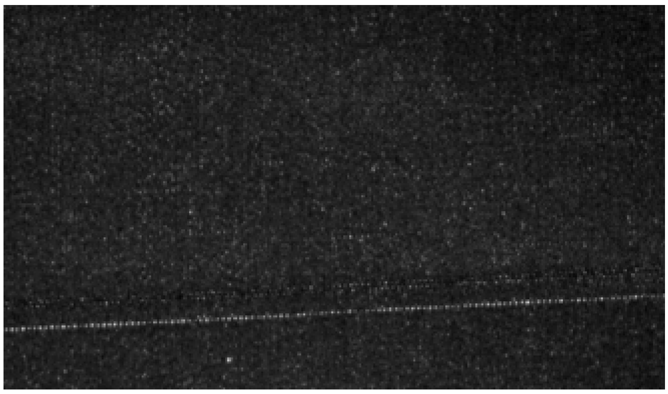
The defect contour information

Figure 3.4 Figure of Adhesive Fabric Detection

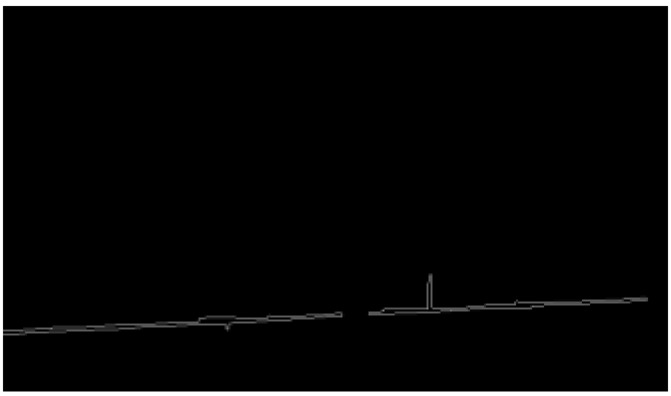
Scratch defect: the fabric has one or more yarns different from the adjacent normal yarns, showing multiple yarns, missing yarns or detailed scratches. The fabric has an obvious straight and strong appearance.The figure of is shown:



Original image of the Scratch defect



grey-scale transformation and Gaussian smoothing



Threshold segmentation

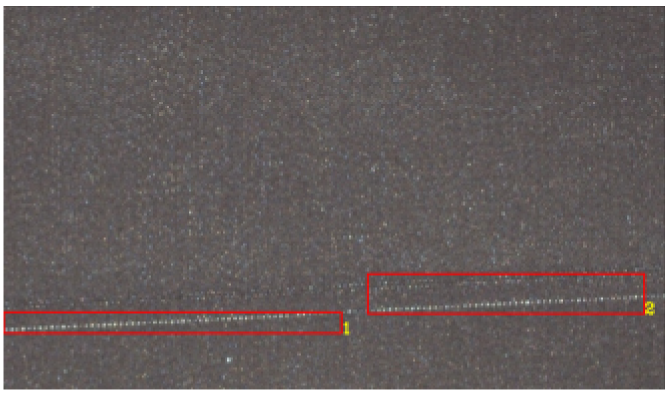


Image marked the defect

```
1a2755ba2f40d9781248089113.jpg:

Defect    1
Length:   1053
Area:     2293
[[[  0 455]]

 [[502 455]]

 [[502 486]]

 [[  0 486]]]

Defect    2
Length:   947
Area:     1766
[[[541 398]]

 [[952 398]]

 [[952 457]]

 [[541 457]]]
```

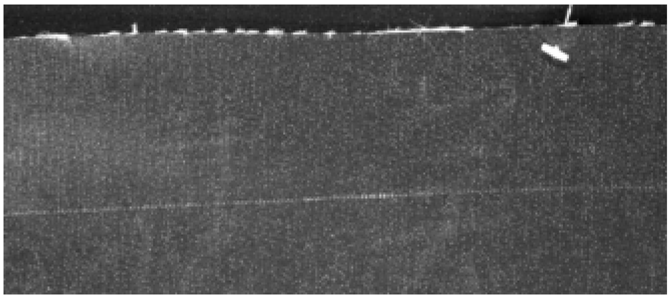
The defect contour information

Figure 3.5 Figure of Scratch Defect Detection

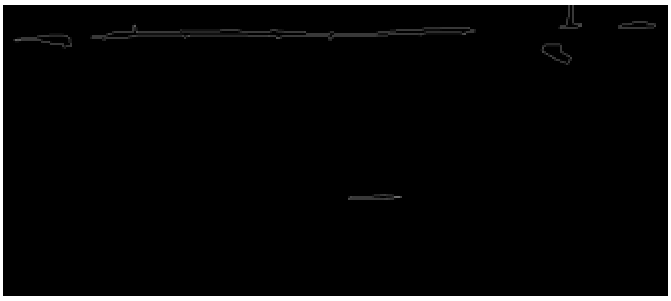
2. Multiple defects

Process the single defect, the detection system can basically detected the exact location of defects and mark it in the original image. But it comes to the various defects rise on a piece of cloth simultaneously, the system is hard to detect every defects in a precise way, as follow:

Original image of the Multiple defects



grey-scale transformation and Gaussian smoothing



Threshold segmentation

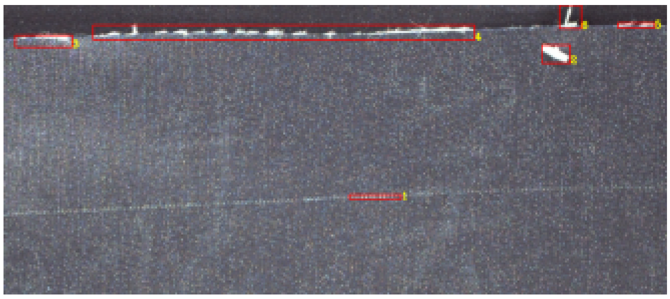


Image marked the defect

```
0acba66f6c097a5e1425276893.jpg:

Defect     1
Length:    244
Area:      644
[[[778 429]]

 [[895 429]]

 [[895 439]]

 [[778 439]]]

Defect     2
Length:    176
Area:      1363
[[[1212   90]]

 [[1276   90]]

 [[1276  131]]

 [[1212  131]]]

Defect     3
Length:    297
Area:      1438
[[[ 26  68]]

 [[154  68]]

 [[154  95]]

 [[ 26  95]]]

Defect     4
Length:    1827
Area:      6285
[[[ 201   45]]

 [[1060   45]]

 [[1060   78]]

 [[ 201   78]]]

Defect     5
Length:    176
Area:      595
[[[1383   40]]

 [[1466   40]]

 [[1466   51]]

 [[1383   51]]]

Defect     6
Length:    191
Area:      787
[[[1252    0]]

 [[1301    0]]

 [[1301   53]]

 [[1252   53]]]
```

The defect contour information

Figure 3.6 Figure of Multiple Defects Detection

The effect of detecting multiple defects is not flaw, both the polishing scratch at the bottom of the figure and middle chromatic aberration are ignored by the system. The difference between the grey-scale value of them and the grey-scale value of the fabric itself is much less than the difference between the grey-scale value of damaged edges, adhesive object and the grey-scale value of the fabric itself. The most possibility is their defect edge contours are filtered out by the setting thresholds.

3. Chapter Summary

In order to test the function of the system, samples of different defects were selected for testing and analysis.Image preprocessing, defect detection and feature extraction are all running normally. Through the analysis of these test data, common and single fabric defects can be successfully detected, which proves that the system is effective and can achieve the goal of fabric defects detection.

Conclusion

This project focuses on the design and implementation of the fabric defect detection system based on machine vision. Firstly, it introduces the application of machine vision in fabric defect detection and the development trend of relevant detection technologies at home and abroad. The image preprocessing, defect detection and defect extraction techniques were analyzed. The design idea and implementation scheme of a fabric defect detection system based on machine vision are proposed. The image is processed with Gaussian filter to smooth and reduce noise, and the image edge detection technology and Canny are used to extract the defect features. The main principle of this system is based on machine vision instead of artificial vision to do cloth defect detection and recognition. The image sequence is transmitted to the image processing module to identify the morphological information of cloth and its suspected defects. The defect detection module performs relevant calculation on these image signals to extract the features of the defects, mark and obtain the location of the defects.

The software of this system is installed on the Windows 10 operating system, which needs the support of the open source visual framework OpenCV. Through the research and analysis of relevant detection technology, the framework design and coding implementation of the system, the fabric defect detection system proposed in this paper is finally realized. In order to test and verify the detection ability and practicability of the system, pictures of different defects were selected for detection and test. The test results show that the system can detect cloth defects in video images in real time, and detect suspicious defects in each frame image in turn.Through the analysis of the experimental results, the single fabric defect can be detected and identified, which achieves the basic purpose of designing the system.

1. Achievements

Compared with the traditional cloth defect detection technology, this system has the following advantages and innovations:

(1) better real-time detection. Multi-thread technology is used to process image preprocessing, defect detection and defect extraction in real time.

(2) more accurate test results.By quoting Canny edge detection method, the common defects on cloth fabric can be detected, and the feature extraction is more comprehensive.

(3) The system is cross-platform.The system is developed based on python language and can be used across platforms. Machine vision framework OpenCV can be also used cross-platform, so that the whole system has a good cross-platform capability.

2. Deficiency and Outlook

The fabric defect detection system proposed in this paper can detect the single defect of plain cloth, but it still exists the following problems and difficulties:

(1) Sometime multiple defects exist in the same piece of fabric, the selection of threshold value is a difficult problem to identify all the defects at the same time.

(2) Fabric with large pattern colour difference will cause interference to defect detection.

(3) There are various of fabric defects, so how to classify the detected defects is a considerable problem.

(4) Under the influence of multiple factors such as production environment, lighting and garment production process, the signal and noise of the detection system are relatively low, and the relatively weak signals are difficult to be detected or cannot be effectively distinguished from the noise.How to adapt to different illumination changes, the influence of noise and other external interference is one of the problems to be solved.

(5) Due to the diversity and variety of defects, the shape of the same type of defects is also different, and the relationship between the causes of the numerous defects and their external manifestations is still unclear.As a result, the description of the defects in the collected image information is not complete, and the success rate of feature extraction of the defects is not high. In addition, it is difficult to find the standard defect image as a reference, which brings difficulties to the detection of the defects, resulting in a low detection rate.

The technology of machine vision is complex, and the mechanism of human vision is also extremely complex. Human beings are unable to describe their visual processes in terms of self-observation. So building a machine vision system is a very difficult and difficult task. However, it can be foreseen that with the continuous development and maturity of machine vision and artificial intelligence technology, it will be more and more widely used in the manufacturing industry in the near future.

References and Bibliography

[1] J. Sobus, B. Pourdeyhimi, J. Gerde, and Y. Ulcay, "Assessing Changes in Texture Periodicity Due to Appearance Loss in Carpets: Gray Level Co-occurrence Analysis," *Textile Research Journal*, vol. 61, no. 10, pp. 557–567, 1991.

[2] W. Mengtao, L. Yue yang, D. Shuai, "Research Progress of Defect Detection Method Based on Machine Vision," *Advanced Textile Technology*, vol. 27, no. 5, Sept. 2019.

[3] K. Sakhare, A. Kulkarni, M. Kumbhakarn, and N. Kare, "Spectral and spatial domain approach for fabric defect detection and classification," *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, 2015.

[4] W. Jasper, J. Joines, and J. Brenzovich, "Fabric defect detection using a genetic algorithm tuned wavelet filter," *Journal of the Textile Institute*, vol. 96, no. 1, pp. 43–54, 2005.

[5] D. Xia, G. Jiang, and P. Ma, "Warp-Knitted Fabric Defect Segmentation Based on Non-Subsampled Wavelet-Based Contourlet Transform," *Autex Research Journal*, vol. 17, no. 4, pp. 319–327, 2017.

[6] J. Jing, "Automatic Defect Detection of Patterned Fabric via Combining the Optimal Gabor Filter and Golden Image Subtraction," *Journal of Fiber Bioengineering and Informatics*, vol. 8, no. 2, pp. 229–239, 2015.

[7] B. Michael, "Uster fabriscan cloth inspecting system," *Textile World*, vol.151, pp.58-60, 2000.

[8] F.G. Stong, "EVSI2Tex2000 Cloth Inspecting System," *Text World*, vol.151, 2001.

[9] M. Aditi and M.Gopi, "Convolution," in Introduction to Visual Computing:Core concepts in Computer Vision, Graphics, and Image Processing, Boca Raton: CRC Press, 2018, pp.41-56.

[10] H. Cao & Z. Wu, "Linear filtering analysis and simulation of noise image," *Information Technology and Informatization*, Aug, 2017.

[11] Z. Chen, "Research and implementation of SIFT algorithm based on image registration," *WuHan Univ of Tech.*, WuHan, 2008.

[12] I. Bao, Y. Zhang, Y. Wang & Y. Luo " Textile image enhancement based on discrete gaussian filter," *Textile journal*, vol.26, Apr, 2005.

[13] J. Xu, S. Ding, Z. Shi & W. Jia, "New theory and new method of image segmentation," School of Comp. Sci.and Tech., China university of mining and technology, 2010.

[14] S. Yang, "Adaptive and fast algorithm for threshold segmentation of gray image", ChongQing Univ., 2014.

[15] Threshold Types, Online documentation. [Online]. Available: http://www.opencv.org

[16] Image Filtering, Online documentation. [Online]. Available: http://www.opencv.org

[17] L. Cui, "Research on the key techniques in the identification of cloth defects," Xidian University, 2013.

[18] J. Canny, "A Computational Approach to Edge Detection," *Readings in Computer Vision*, pp. 184–203, 1987.

[19] Graph API: Converting image from one color space to another, Online documentation.[Online]. Available: http://www.opencv.org

[20] Operations on arrays, Online documentation. [Online]. Available: http://www.opencv.org

[21] X. Jia, "Fabric defect detection based on open source computer vision library OpenCV," *2010 2nd International Conference on Signal Processing Systems*, 2010.

[22] Matrix Reductions, Online documentation. [Online]. Available: http://www.opencv.org

[23] Operations on arrays, Online documentation. [Online]. Available: http://www.opencv.org

[24] Structural Analysis and Shape Descriptors, Online documentation. [Online]. Available:

http://www.opencv.org

[25] Miscellaneous Image Transformations, Online documentation. [Online]. Available: http://www.opencv.org

[26] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, p. 396, 1985.

Appendix 1

```
#-*- coding: UTF-8 -*-
import cv2
import numpy as np
import math
from scipy import signal
from PIL import Image
from PIL import ImageEnhance


def powerTrans(gray, c, y):
   gray_power = np.uint8(c * (gray ** y))
   return gray_power

def plf(x, X, Y):
   x = np.float32(x)
   y = 0
   for i in range(len(X)):
      t = 1
      for j in range(len(Y)):
         if j != i:
            t = t * ((x - X[j]) / (X[i] - X[j]))
      y = np.uint8(y + t * Y[i])
   return y

def fft2Image(src):
   r,c = src.shape[:2]
   rPadded = cv2.getOptimalDFTSize(r)
   cPadded = cv2.getOptimalDFTSize(c)
   fft2 = np.zeros((rPadded,cPadded, 2),np.float32)
   fft2[:r,:c, 0] =src
   cv2.dft(fft2,fft2,cv2.DFT_COMPLEX_OUTPUT)
   return fft2

def amplitudeSpectrum(fft2):
   real2 = np.power(fft2[:,:,0],2.0)
   Imag2 = np.power(fft2[:,:,1],2.0)
   amplitude = np.sqrt(real2+Imag2)
   return amplitude

def graySpectrum(amplitude):
         amplitude = np.log(amplitude +1.0)
         spectrum = np.zeros(amplitude.shape, np. float32)
         cv2.normalize(amplitude, spectrum, 0, 1, cv2.NORM_MINMAX)
         return spectrum
```

```
def grayReversal(gray):
   gray_reversal = 255 - gray
   return gray_reversal

def getGaussKernel(sigma,H,W):

        gaussMatrix = np.zeros([H,W], np.float32)
        cH = (H-1)/2
        cW = (W-1)/2

        for r in range(H):
                for c in range(W):
                        norm2 = math.pow(r-cH,2) + math.pow(c-cW,2)
                        gaussMatrix[r][c] =math.exp(-norm2/(2*math.pow(sigma,2)))

        sumGM = np.sum(gaussMatrix)
        gaussKernel = gaussMatrix/sumGM

        return gaussKernel

def gaussBlur(image, sigma, H, W, _boundary ='fill', _fillvalue = 0):

        gaussKernel_x = cv2.getGaussianKernel(sigma, W, cv2.CV_64F)
        gaussKernel_x = np.transpose(gaussKernel_x)
        gaussBlur_x = signal.convolve2d(image, gaussKernel_x, mode = 'same', boundary =
_boundary, fillvalue = _fillvalue)

        gaussKernel_y = cv2.getGaussianKernel(sigma, H, cv2.CV_64F)
        gaussBlur_xy = signal.convolve2d(gaussBlur_x, gaussKernel_y, mode = 'same',
boundary = _boundary, fillvalue = _fillvalue)

        return gaussBlur_xy

def calcGrayHist(image):

   rows,cols = image.shape[:2]
   grayHist = np.zeros([256], np.uint64)
   for r in range(rows):
      for c in range(cols):
         grayHist[image[r][c]] +=1

   return grayHist
```

```
def otsu(image):

    rows,cols = image.shape
    grayHist = calcGrayHist(image)
    uniformGrayHist = grayHist/float(rows*cols)
    zeroCumuMoment = np.zeros([256], np.float32)
    oneCumuMoment = np.zeros([256], np.float32)

    for k in range(256):

            if k == 0:
                    zeroCumuMoment[k] = uniformGrayHist[0]
                    oneCumuMoment[k] = (k)*uniformGrayHist[0]

            else:
                    zeroCumuMoment[k] = zeroCumuMoment[k-1] + uniformGrayHist[k]
                    oneCumuMoment[k] = oneCumuMoment[k-1] + k*uniformGrayHist[k]

    variance = np.zeros([256], np.float32)

    for k in range(255):

            if zeroCumuMoment[k] == 0 or zeroCumuMoment[k] == 1:
                    variance[k] = 0

            else:
                    variance[k] = math.pow(oneCumuMoment[255]*zeroCumuMoment[k] - 
oneCumuMoment[k],2)/(zeroCumuMoment[k]*(1.0 - zeroCumuMoment[k]))

    threshLoc = np.where(variance[0:255] == np.max(variance[0:255]))
    thresh = threshLoc[0][0]
    threshold = np.copy(image)
    threshold[threshold > thresh] = 255
    threshold[threshold <= thresh] = 0

    return (thresh,threshold)
```

```
def createLPFilter(shape, center, radius, lpType, n=2):

   rows, cols =shape[:2]
   r,c = np.mgrid[0:rows:1, 0:cols:1]
   c-= center[0]
   r-= center[1]
   d = np.power(c,2.0) + np.power(r,2.0)
   lpFilter = np.zeros(shape, np.float32)

   if (radius<=0):
           return lpFilter

   if (lpType == 0):
      lpFilter = np.copy(d)
      lpFilter[lpFilter<pow(radius,2.0)] = 1
      lpFilter[lpFilter<=pow(radius,2.0)] = 0
   elif (lpType == 1):
      lpFilter = 1.0/(1.0+np.power(np.sqrt(d)/radius,2*n))

   elif (lpType == 2):
      lpFilter = np.exp(-d/(2.0*pow(radius,2.0)))

   return lpFilter

def histNormalized(InputImage,O_min = 0,O_max = 255):
   I_min = np.min(InputImage)
   I_max = np.max(InputImage)
   rows,cols = InputImage.shape

   OutputImage = np.zeros(InputImage.shape,np.float32)

   cofficient = float(O_max - O_min)/float(I_max - I_min)
   for r in range(rows):
      for c in range(cols):
         OutputImage[r][c] = cofficient*( InputImage[r][c] - I_min) + O_min
   return OutputImage
```

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
#   final.py
#
#   Copyright 2020 Administrator <Administrator@HE-CHENGYANG>
#
#

def main(args):
	return 0

if __name__ == '__main__':
	import os
	import cv2
	import sys
	import math
	import numpy as np
	from PIL import Image
	from PIL import ImageEnhance
	C=0
	figure_address='source_images'
	for root, dirs, files in os.walk(figure_address):
	    for d in dirs:
	        print(d)
	for file in files:
		print(file)
		img_path = root+'/'+file
		img = cv2.imread(img_path,1)
		print(img_path, img.shape)

		#process
		C = C+1
		gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
		gauss = cv2.GaussianBlur(gray, (3, 3), 0)

		dst = np.zeros(gray.shape, np.uint8)
		cv2.normalize(gauss, dst, 255, 0,cv2.NORM_MINMAX, cv2.CV_8U)
		cv2.imwrite("process/processed"+str(C)+'.png',dst)
```

```
        ##average and std
        (mean , stddv) = cv2.meanStdDev(dst)
        ##thresholds
        Maxthr = mean + 3*stddv
        Minthr = mean - 3*stddv
        t1, lowthres = cv2.threshold(dst , Minthr, 255, cv2.THRESH_BINARY)

        t2, maxthres = cv2.threshold(dst, Maxthr, 255,
cv2.THRESH_BINARY_INV)

        thres = cv2.bitwise_and(lowthres, maxthres)
        t3, thres = cv2.threshold(thres, 127, 255, cv2.THRESH_BINARY_INV)
        openkernel = np.ones((2,2), np.uint8)
        opening = cv2.morphologyEx(thres, cv2.MORPH_OPEN, openkernel)
        closekernel = np.ones((9,9), np.uint8)
        closing = cv2.morphologyEx(opening , cv2.MORPH_CLOSE, closekernel)

        openkernel = np.ones((3,3), np.uint8)
        opening = cv2.morphologyEx(closing, cv2.MORPH_OPEN, openkernel)

        erodekernel = np.ones((3,3), np.uint8)
        erode = cv2.morphologyEx(opening, cv2.MORPH_ERODE, erodekernel)

        contours,hierarchy =cv2.findContours(erode,  cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
        for i in range(len(contours)):
            length = cv2.arcLength(contours[i], True)
            area= cv2.contourArea(contours[i])
            if length < 25 or area < 15:
                cv2.drawContours(erode,[contours[i]],0,0,-1)

        closekernel = np.ones((50,50), np.uint8)
        closing = cv2.morphologyEx(erode, cv2.MORPH_CLOSE, closekernel)

        dilatekernel = np.ones((4,4), np.uint8)
        dilate = cv2.morphologyEx(closing , cv2.MORPH_DILATE, erodekernel)


        (mean , stddv) = cv2.meanStdDev(dilate)
        edges =cv2.Canny(dilate,mean/6,mean)
        cv2.imwrite("process/edges"+str(C)+'.png',edges)
        Con,hierarchy =cv2.findContours(dilate, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
```

```
        contour = Con[:]

        i = 0
        fname =open("information/"+str(C)+'.txt','w')
        fname.write(file+": "+'\n')
        if len(Con) == 0:
            fname.write('No defect in this image\n')
        else:
            for i in range(len(Con)):
                x,y,w,h =cv2.boundingRect(Con[i])
                Con[i] = np.array([[[x,y]],[[x+w,y]], [[x+w,y+h]], [[x,y+h]]] )
                length = cv2.arcLength(contour[i], True)
                area = cv2.contourArea(contour[i])
                cnt=Con[i]
                right = tuple(cnt[cnt[:,:,0].argmax()][0])
                top = tuple(cnt[cnt[:,:,1].argmin()][0])
                bottom = tuple(cnt[cnt[:,:,1].argmax()][0])
                point=tuple([right[0],bottom[1]])
                cv2.putText(img,str(i+1), point,
cv2.FONT_HERSHEY_COMPLEX,0.75,(0,255,255),2)
                fname.write('\n')
                fname.write('Defect'+'\t'+str(i+1)+'\n')
                fname.write('Length:'+'\t'+str(round(length))+'\n')
                fname.write('Area:'+'\t'+str(round(area))+'\n')
                fname.write(str(Con[i])+'\n')

        fname.close()
        dst_img = cv2.drawContours(img,Con,-1,(0,0,255),2)
        cv2.imwrite("results/final"+str(C)+'.png',dst_img)
        cv2.waitKey(0)
    sys.exit(main(sys.argv))
```

Appendix 2

Because the project only have simulation & analysis, no part is influenced by Coronavirus (COVID-19).