



HOKKAIDO UNIVERSITY

Title	Multiple Paired Pixel Consistency Model for Robust Defect Detection in Printed Logotypes
Author(s)	Xiang, Sheng
Degree Grantor	北海道大学
Degree Name	博士(工学)
Dissertation Number	甲第14596号
Issue Date	2021-03-25
DOI	https://doi.org/10.14943/doctoral.k14596
Doc URL	https://hdl.handle.net/2115/81623
Type	doctoral thesis
File Information	ShengXiang_14596.pdf



SSI-DT79185030

Doctoral Thesis

**Multiple Paired Pixel Consistency Model for Robust
Defect Detection in Printed Logotypes**

Sheng XIANG

January, 2021

Division of Systems Science and Informatics
Graduate School of Information Science and Technology
Hokkaido University

Doctoral Thesis
submitted to Graduate School of Information Science and Technology,
Hokkaido University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy.

Sheng XIANG

Thesis Committee: Takayuki TANAKA Professor
Masahiko ONOSATO Professor
Satoshi KANAI Professor

Multiple Paired Pixel Consistency Model for Robust Defect Detection in Printed Logotypes*

Sheng XIANG

Abstract

In modern manufacturing, quality control (QC) is an integral technique. Detection of product flaws plays a crucial role in quality management and ensuring that the product serves the consumer. In this study, we assess how currently available vision systems execute a range of QC tasks on printed items. In particular, we take into consideration the examination of printed characters/text or logotypes for defects, such as holes, scratches, cracks, and foreign artifacts. Printing-defect analysis is currently most done by human testers, and is labor-intensive and time-consuming job. Moreover, the results of an inspection might be unreliable because humans may arrive at different results depending on the time and the mood, skills, and experience of the inspectors. Therefore, human inspection is being replaced by automatic visual inspection systems.

Traditional defect-detection algorithms can be categorized as conventional feature-based and data-driven-based algorithms. Furthermore, conventional feature-based methods can be divided into four categories: statistical-, structure-, filter-, and model-based methods. Data-driven-based methods can be divided into two categories: traditional machine learning methods and deep learning techniques. Both data-driven-based and feature-based algorithms have their advantages and disadvantages. Feature-based methods usually have clear algorithmic implications and are therefore easy to control. Suitable features for various application scenarios can be configured for specific objectives. In comparison, feature-based methods are typically effective and simple to implement since they do not rely on massive quantities of data. However, certain difficulties do occur, such as the failure inability to detect small-sized defects and texture irregularities satisfactorily. Data-driven methods are usually implemented by designing certain learnable parameters of the model and then teaching the data model. Training data usually contains images and corresponding annotations that are manually annotated. While data-driven methods exhibit high precision and generalization, they involve a significant amount of learned data and manual annotations. The training phase often involves substantial computational capital and time.

This work focuses on the identification of printing defects on surfaces embossed using randomly spaced three-dimensional (3D) micro-textures. The embossing processes

*Doctoral Thesis, Division of Systems Science and Informatics, Graduate School of Information Science and Technology, Hokkaido University, SSI-DT79185030, January 6, 2021.

produces a very small variety of convex and concave shapes on the surfaces of metals, plastics, or other materials. Changes in the illumination on certain surfaces have a major effect on their appearance, resulting in difficulty in identifying defects. To realize this goal, the use of the multiple paired pixel consistency (MPPC) model was recommended. We first propose a consistency measure based on the correlation of consistent pixel pairs to obtain a robust defect-free model. We then set in motion a new assessment technique to accurately identify defects. Furthermore, a modification method called position-dependent data inhibition (PDI) is proposed to further improve the robustness and performance of the MPPC model.

This overall dissertation is structured as follows.

Chapter 1 introduces the importance of identification of defects and presents the associated works on defect detection. Some challenges in defect detection are discussed. Furthermore, the motivations and contributions of this study are explained.

Chapter 2 introduces the orientation codes (OCs), the use of which could reduce the influence of illumination fluctuations on defect detection; thus, OCs are used as the basis of the proposed method. First, we implement the original version of the OCs and then expand it by presenting two types of operations: a precise spatial differentiation for calculating the codes with a higher resolution and signed difference between any two codes as preparation for building up a more precise statistical model of their difference. Using these operations, we add a more reliable scheme to explain the statistical relationship between a pair of any logotypes pixels.

Chapter 3 introduces the proposed MPPC model in detail, including the fundamental principle and structure of the MPPC model. First, we observe the defect-free images of logotypes to determine the relationship between any pair of pixels in the logotype. We then introduce kurtosis to obtain the potential distributions. After analyzing the distributions, we explain how to pick the supporting pixel for each target pixel and finally construct the MPPC model for each pixel pair.

Chapter 4 provides a discussion of the method for using the proposed MPPC model of the relationship between pixel pairs in the defect-free logotype for identification of several types of logotype defects. Defect detection can be divided into two main stages. First, the status of each pixel pair is determined. We then identify the normal or abnormal states at the corresponding position defined by the elemental MPPC model.

Chapter 5 is focused on the PDI modification suggested on the basis of MPPC to further strengthen the robustness of the MPPC and stabilize its performance. The basic principles and mechanism of the PDI are explored in depth in this chapter. Finally, we verify the capacity of the PDI by appropriate experiments.

Chapter 6 lays out the experimental setup in detail. In this chapter, comparative experiments for the MPPC and MPPC+PDI models using several real defective images and synthesized or artificial defective images under different conditions such as illumination fluctuation and different noise intensities are designed. We test the robustness and efficiency of our methods through these experiments.

The final chapter outlines the key points of this study and presents a discussion of our algorithms. Finally, the strategy and idea for possible future works are discussed.

Keywords: Multiple paired pixel consistency (MPPC), Position-dependent data inhibition Modification (PDI), Defect detection, Illumination changes, noisy inhibition

Contents

1. Introduction	1
1.1 Research background	1
1.2 Conventional visual-based inspection methods	3
1.2.1 Traditional texture extraction methods	5
1.2.2 Data-driven feature extraction methods	10
1.3 Challenges of defect detection	11
1.4 Motivation and contribution of the study	15
1.5 Organization of dissertation	16
2. Improvement of Orientation Codes	19
2.1 Orientation codes	20
2.2 Signed difference in OC	21
2.3 Precise differentiation	24
2.4 Summary	26
3. Multiple Paired Pixel Consistency Model (MPPC)	27
3.1 Overview	27
3.2 Observing defect-free images of logotypes	28
3.3 Training stage of MPPC	31
3.3.1 Selection of supporting pixels	32
3.3.2 Defect-free model of pixel pairs	35
3.3.3 Computation time in training	36
3.4 Summary	37
4. Defect Detection	39
4.1 Identification pixel pair state	40
4.2 Identification target pixel's state	41
4.3 Parameter discussion	42
4.3.1 Effect of the value Γ on experimental results	42
4.3.2 Effect of the number of supporting pixel N on experimental results	43
4.4 Post-filtering	46
4.5 Summary	48
5. Position-dependent Data Inhibition	49
5.1 χ^2 test for appropriateness	51
5.2 Skewness test for outlier detection	53

5.3	Inhibition index list	54
5.4	Verification of PDI's performance	55
5.5	Summary	57
6.	Experimental Evaluation	59
6.1	Experiment setup	59
6.2	Analysis measurement	61
6.3	Experiments result	63
6.3.1	Different defect sizes	63
6.3.2	Different illumination conditions	65
6.3.3	Different noise intensities	66
6.3.4	Different characters	68
6.3.5	Comparison with other methods	70
6.4	Computation cost	72
6.5	Discussion	75
6.6	Summary	76
7.	Conclusions and Future Works	77
7.1	Conclusions	77
7.2	Future works	78
	Acknowledgements	81
	References	83
Appendix A	Publications lists	91
A.1	Journal Papers	91
A.2	International Conferences	91
A.3	Domestic Conferences	91

List of Figures

1.1	Examples of products whose surfaces are required to be test for quality [1, 2]	1
1.2	Two methods of product defect inspection. (a) Manual inspection [3] and (b) automatic inspection [4]	2
1.3	Categorization of quality control strategies	3
1.4	Summary of visual-based inspection methods	4
1.5	Examples of common types, shapes, and sizes of defects	12
1.6	Example of a printed product. (a) original image and (b)image after sharpening	13
1.7	Imaging effect of the same object under various illumination conditions	14
2.1	An example of logotype printed on embossed surface	19
2.2	Example of printed character. (a) defect-free example and (b) defective example	20
2.3	Sixteen-OC	21
2.4	An example of real logotype image.(a) original image and (b) OC image	22
2.5	Scheme of signed difference in OC	22
2.6	An example is the printed character under different illumination conditions. (a) and (b) the character under strong and low illumination, (c) and (d) the corresponding OC histogram, and (e) and (f) are the corresponding HOG histogram.	23
2.7	Examples of applying the proposed, the Sobel and the Roberts operator. From (a) to (e), Lena image, the magnified parts, the edge detection result by Sobel, by Roberts, and by the proposed operator.	25
3.1	The framework for proposed method	27
3.2	Observing and comparing OC difference histograms	29
3.3	Two representative types of OC difference in pair. (a) and (b) distribution of OC difference for logo-logo pair, logo-background pair.	30
3.4	Kurtosis distribution for two types of pixel pairs	31
3.5	Fundamental definition of image data. Target pixel P and an arbitrary pixel C with their OC sequence varies with different product p_t and q_t	32
3.6	Scheme of MPPC modeling	33
4.1	Scheme of defect detection by the MPPC model	39
4.2	Quantitative analysis of defect detection for different characters	44

4.3	some examples of defect detection result by using different number of supporting pixel. (a) original image, (b) ground truth image. From (c) to (l), defect detection results under the number of supporting pixels from 2 to 20.	45
4.4	Quantitative analysis of defect detection	46
4.5	Example of detection result	47
4.6	Definitions of true positive (TP), false positive (FP), true negative (TN) and false negative (FN) in defect detection	47
5.1	Example of a detection result	49
5.2	Contribution of C on defect detection	50
5.3	Example of OC difference in a randomly selected pixel pair	51
5.4	The critical value p of the χ^2 distribution	52
5.5	Two examples of fitness checks. (a) and (b) the case of with outliers ($\chi^2 = 105.3$) and without outliers ($\chi^2 = 10.2$)	53
5.6	The two distributions of skewness	54
5.7	Effect of applying the outlier removal process	55
5.8	Detailed structure of IIL. (a) three examples and (b) the corresponding inhibition index list.	56
5.9	Some examples of defect detection results	57
6.1	Some examples of real defect images	59
6.2	Two types of experimental images. (a) real defect and (b) synthesized defect.	60
6.3	Abridged general view of the analysis measurements	62
6.4	Some examples of defect detection with different defect sizes	64
6.5	Some examples of defect detection under different illumination conditions	66
6.6	Some examples of defect detection with different amount of additional noise	68
6.7	Some examples of defect detection results for different characters	69
6.8	Some examples of defect detection results for synthesized defect	72
6.9	Some detected defects in the real defect images	73
6.10	Three types of real defects	75

List of Tables

1.1	Some commonly used textural defect detection methods	5
1.2	Some commonly used defect detection methods based on supervised and unsupervised or semi-supervised classifiers	10
3.1	Hardware information	37
4.1	Quantitative analysis of defect detection, when $\Gamma = 10$	42
4.2	Quantitative analysis of defect detection, when the number of supporting pixels=10	43
5.1	Pixel-based performance for different characters	57
6.1	Pixel-level based performance evaluation of defect detection under differ- ent size of defect	65
6.2	Image based performance evaluation for different defect sizes	65
6.3	Pixel-level performance of defect detection under different illumination conditions	67
6.4	Image-level based performance of defect detection under different illumi- nation conditions	67
6.5	Pixel-level based performance of defect detection for different noise inten- sities	68
6.6	Image-level based performance of defect detection for different noise in- tensities	69
6.7	Pixel-level based performance of defect detection for different characters .	70
6.8	Image-level based performance of defect detection for different characters	70
6.9	Pixel-level based performance of defect detection for comparing	71
6.10	Image-level based performance of defect detection for comparing	74
6.11	Computation time in the detecting stage	74

Chapter 1. Introduction

1.1 Research background

Defect detection, which comprises the assessment of the quality of a product, plays an important role in manufacturing industries. With improvements in the manufacturing technology, an increasing number of factories have adopted automatic production lines, and the production efficiency has been greatly improved. As seen in Fig. 1.1, owing to the progress of current production processes, a significant number of goods are manufactured every day. Owing to the improvement in the existing manufacturing technologies, a large number of products are being produced every day, as shown in Fig. 1.1. Over



Figure 1.1: Examples of products whose surfaces are required to be test for quality [1, 2]

the years, the living standards have gradually risen, leading to a continuous increase in the expectations of customers in terms of product quality. Therefore, product quality control has become a very essential issue in production processes. Defect detection plays a significant role in quality control of products. One of the most important aspects of defect detection is finding defects in logotypes. This is because the logotype is the part of the product that is most easily by customers.

In the past decades, the defect detection is often assessed by human inspectors, as shown in Fig. 1.2(a). Manual inspection is a labor-intensive and time-consuming operation. Due to the advanced manufacturing technology, a large number of products are produced every day. Manual inspection is too slow to keep up with the speed



Figure 1.2: Two methods of product defect inspection. (a) Manual inspection [3] and (b) automatic inspection [4]

of the production line. This results in the wastage of resources, which means that the production plant is not running at full capacity. Moreover, the results of such an inspection may be unreliable, since human testers may have varying outcomes depending on the time and the mood, abilities, and expertise of the testers. In addition, human inspection is subjective because there is no consistency level of quality standard that is independent of the operator. Therefore, it is necessary to identify a method realizing high-precision and low-cost defect detection.

To solve the afore mentioned problems, human inspection is being replaced by automatic inspection systems as seen in Fig. 1.2(b). The automatic inspection system has independent and objective detection standards and is not adversely affected by human factors; therefore, it can maintain a stable detection accuracy and has the advantages of speed and not requiring a large number of workers. The automatic detection system can be categorized into destructive and non-destructive systems, as shown in Fig. 1.3. It is preferred to perform a quality inspection without destroying the product. And the automatic detection is usually performed in three stages: sensing, signal processing, and detection result output. First, the signals from the parts of interest are obtained via sensors. Then the signals are then processed to obtain useful information, and finally, according to the effective information, the defect detection is performed. The commonly used sensors include microwave sensors, infrared sensors, and cameras. The camera is most inexpensive and reliable sensor, and the corresponding system is easy to build. Microwave and infrared sensors are not effective for detecting minor flaws and defects that have no uneven surfaces. However, even micro-defects and low-contrast defects can be identified using a camera. In recent years, with the rapid growth of computer vision technologies, more focus has been paid to camera-based visual inspection. This theoretical approach includes using a camera as a monitor to take photographs of the object, and then developing an accurate computer vision system to classify the pixels corresponding to the product surface defect using an algorithm.

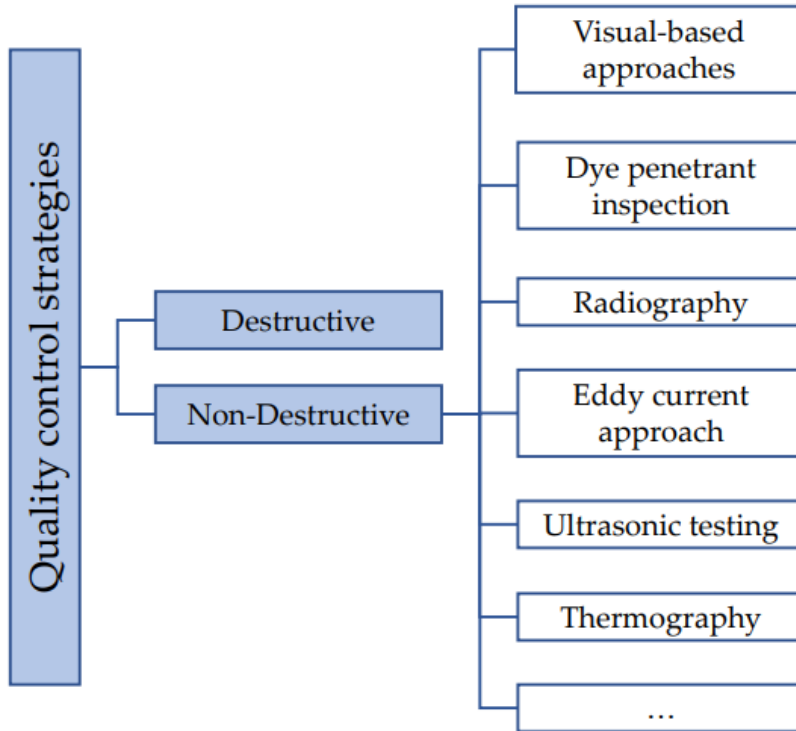


Figure 1.3: Categorization of quality control strategies

1.2 Conventional visual-based inspection methods

One of the most common techniques for visible defect identification in the industry is the visual-based defect detection methods. The primary purpose of visual-based methods is to grasp and artificially represent the natural world and environment. Artificial representation primarily defines the mathematical or logical relation between the input image and some elements, namely environment representation, during the identification of images. The mathematical/logical association aims at converting the input images to the model, which converts the image information into relevant information of significance to the application. This subsection reviews the current techniques of visual-based inspection. In the basic process of the vision-based method, the image of the detected object is first acquired and preprocessed. Then extract effective features from the image; Finally, according to the extracted features, each pixel in the image is divided into defective pixel and non-defective pixel. Feature extraction method is the most important. It can be classified into two categories: traditional texture extraction and data-driven feature extraction. In the past, the majority of the detection algorithms were based on traditional texture extraction. However, with the substantial improvement in computer computing power and the development of deep neural networks, data-driven feature extraction has gained increasing research attention.

The traditional texture extraction and data-driven feature extraction methods have their own merits and demerits. The traditional texture extraction methods usually have practical meanings, such as extracting boundaries. They can be used in various application scenarios, such as with different materials and illumination conditions and for designing appropriate features to realize functions according to different targets. Moreover, these methods are simple to implement, fast, and do not require a huge volume of support data. Their disadvantages are also apparent. They are less portable because they are designed for a particular material or scenario, and the methods may not work effectively if the material of the object being tested is changed. Recently, data-driven approaches have attracted increasing attention. These approaches, in particular the deep neural network algorithm, have tremendous potential in the field of defect detection. They have the benefits of high accuracy and generalization. However, they have the apparent drawback of having a large amount of training data to be mastered. With constant progress in manufacturing technologies and an extremely low rate of defect, it has become difficult to gather adequate samples of defect for learning. It is difficult to label the training data manually.

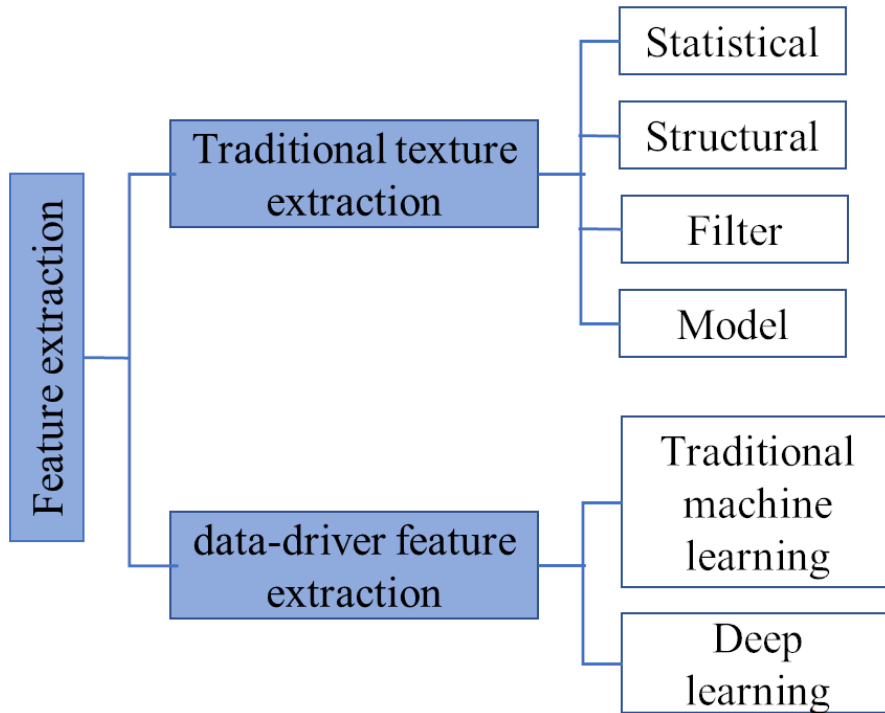


Figure 1.4: Summary of visual-based inspection methods

Approach	Method
Statistical	Histogram properties
	Co-occurrence matrix
	Local binary pattern
	Other gray-level statistics
	Auto-correlation
	Registration-based
Structural	Primitive measurement
	Edge features
	Skeleton representation
	Morphological operations
Filter based	Spatial domain filtering
	Frequency domain analysis
	Joint spatial/spatial frequency
Model based	Fractal model
	Random field model
	Texem model
	Auto-regressive

Table 1.1: Some commonly used textural defect detection methods

1.2.1 Traditional texture extraction methods

Textures offer valuable and unique knowledges for visual-based detection systems. In this subsection, we present some existing texture-extraction-based defect inspection methods. We also add various types of texture analyzes and classifications, as the general role of the defect inspection can be seen as a question of texture analysis.

Xie et al. [5] categorized and presented a comparison of the strategies for obtaining texture features into four groups with regard to different paper in this field, and Ngan et al. [6] performed an extended review on this subject in 2011.

presented a thorough review of texture defect detection and grouped texture analysis strategies into four major groups, including the statistical-, structural-, filter-, and model-based approaches. They [5] focused only on non-color-related methods. In the case of color texture analysis techniques, are color related features. In the following, we introduce the four main categories of methods in details. Table 1.1 shows some commonly used textural defect detection methods.

Statistical approaches

Statistical methods rely on the study of the spatial distribution of pixel values in captured images; they are typically applicable to irregularly textured surfaces. This category of techniques has been of interest in many publications, ranging from low-level

to advanced statistical techniques. Table 1.1 shows some example, such as histogram, autocorrelation, and local binary mode (LBP).

Histogram attributes can effectively support high-order and low-order processing, and the calculation cost is low.

The range, mean, median, geometric mean, harmonic mean, standard deviation, and variance of the histogram properties are widely used. And some methods of histogram comparison statistics are used for the extraction of texture features, such as L1 norm, L2 norm, intersection of histogram, Chi-square and Mallow's or Earth mover's distance. For example, the method introduced in [7, 8, 9] has proven to be a low-cost, low-level defect detection method. And as tonality disruptor [10, 11], they can also be used because their characteristics are invariant to translation and rotation and indifferent to the exact spatial distribution of color pixels. An enhanced Otsu thresholding approach called Weight Object Variance (WOV) have recently proposed by Yuan et al. [12], which can achieve a 94% detection rate. The authors also enhanced Otsu's thresholding method in another study, but there is no specific report on the performance comparison of this method.

One of the most well-known and widely used in texture characteristics is spatial gray-level co-occurrence matrices (GLCMs) [13]. The spatial association of gray-scale pixels to co-occurrence matrices is calculated by these statistical methods. By measuring how many pairs of pixels with certain values and having a given spatial relationship occur in an image, the GLCM function represents an image's texture and then provides a displacement vector. The texture characteristics is defined and then extracted by energy, entropy, contrast, homogeneity, and correlation. Several publications have reported the use of co-occurrence matrices for detecting defects [14, 15, 16]. Although widely used, co-occurrence matrix features have several drawbacks. There does not seem to be a widely agreed solution for maximizing the vector of displacement [17]. In [18, 19], it was stated that the co-occurrence matrices exhibited poorer performance in defect detection compared to the Markov Random Fields (MRF), filtering-based methods. A GLCM based detector with radial base probabilistic neural network was introduced by Capizzi et al. [20] to inspect fruit defects. The GLCM and other methods used by Samy et al. [21] to extract features for training a supervised Support Vector Machine (SVM).

The autocorrelation method can be applied to textures that comprise repeated patterns, such as textiles. The objective of this mathematical algorithm [22, 23] is to determine the correlation between the texture and the regularity of the translation vertex sum with the displacement vector. A research on yarn-dyed fiber failure detector, integrating autocorrelation of the GLCM, has recently been reported by Zhu et al. [24]. However, as this method is sensitive to noise interference and only applicable to regular textures, it is not suitable for the majority of detection tasks such as defect detection in random texture surfaces.

LBP is an inexpensive computational texture operator with a high performance. Ojala et al. [25] initially recommended the LBP operator. It's a tool that calculates the

local contrast with a shift-invariant. In a sliding pane, the operator uses the middle pixel of the window as the threshold of the gray pixel and its neighboring pixels. The effects of the threshold are viewed as binary numbers. The LBP operator is resistant to lighting and rotation shifts and therefore renders it a robust operator. This has been used in a wide range of flaw identification applications for a wide range of materials [26, 27, 28] such as ceramics, wood or other materials. The GLCM and LBP techniques have recently been merged by Zhang et al. [29] to obtain image features for training a back-propagation neural network, and Sindagi et al. [30] proposed an improved LBP for training SVM classifiers for defect detection.

Structural approaches

The structural approach focuses specifically on the spatial placement of components of the texture. Such elements can be extracted from the texture and defined as primitives of texture. A dynamic texture model can be created by applying the arrangement rules of spatial texture primitives. Generally, plain gray-scale zones, line fragments or single pixels are the texture primitives. These components are often used with positioning laws derived from their geometric relations or spatial statistics. Furthermore, defects can be regarded as abnormal structures of an image. Therefore, to achieve defect detection, a structural description and analysis may be done.

Several structural methods have been proposed for defect detection models. Chen and Jain [31] described a model with a skeleton structure for defect detection. First, they threshold the image histogram and then map it to the skeleton representation. Subsequently, some statistical measures are defined based on the position and length histograms of the skeleton. A method called texture blob detection for detecting fabric defects was suggested by Bennamoun and Bodnarova [32]. Song et al. [33] extracted structural features using color and texture information.

First, according to its color information, each image is grouped into fragments; then using morphological smoothing to generate the binary image. The structural features of each spot in the binary image are then extracted, such as the location, fractal perimeter, elongation, and spatial detail. Finally, by color checking and blob defect identification, defect detection is realized. A method based on the multi-scale structural similarity index (MS-SSIM) was recently proposed by Tolba and Raafat [34], which distinguished irregular features with a success rate of 99.1%. Tolba and Raafat [34] recently proposed a multi-scale structure similarity index for identifying irregular features, which have 99.1% success rate. The use of backlight technology in an automated optical failure detection system based on morphology, was suggested by Yun et al. [35]. Finally, the authors [36] introduced a new solution, that of leading the prior information on the least square regression and overcoming the issue of subspace segmentation. Other studies, such as [37, 38], also proposed the use of morphological operations for defect detection.

Filter-based approaches

The image can be represented using the features detected (such as edges, textures and regions). One of the first attempts at image processing was filtering these features, particularly in the case of edge information. The boundary of the spatial pulse intensity change of the image can be regarded as the edge [39]. Multiple filters and algorithms, such as Sobel, Robert, Canny, Deriche, Laws, and Laplacian filters, can be used in order to extract edges from images. In [40, 41], the authors utilized the eigen filters feature detector as a series of structure-independent macro windows. Eigen filters are deemed to be particularly susceptible to noise and local distortion. This makes it less acceptable for inspection of fabrics. However, their low complexity and ability to efficiently integrate different time- and frequency-domain constraints make them common as a general form of characteristics identification in contrast with other approaches.

In the majority of cases, operation in the spatial domain involves noise and complexity. Therefore, The Fourier transform (FT) is then used to translate the image into the frequency domain as defined in [42]; FT provides leverage for easy noise filtering and image processing. Transforming the image into a Fourier domain, filtering it and then transforming it back into a spatial domain is the basic principle. The discrepancy between the initial image and the transformed image can be viewed as a possible defect [43] depending on the application function in the transformation. A central space spectrum was proposed by Chan and Pang [44] based on the premise that the defects normally occur horizontally and vertically. These filters, however, presume a normal texture. Hoffer et al. [45] proposed optical Fourier transform for fabric detection.

Owing its coefficients, the FT is dependent on the entire image. The positioning of defects in the spatial domain cannot be done with this property. Applying a window FT to obtain spatial dependency is the most general approach. And then the Gabor transform (GT) will be created if the window function is a Gaussian function. The GT aims to Spatial and spatial frequency domains [46]. This approach can be separated into two distinct forms. Several filters are stored in the first type in the predetermined frequencies and directions, covering all the potential frequencies in the images and measuring the association between them [47]. However, several measurements will be required to obtain a high precision. The second type requires the design of the best filter for the appropriate recognition region in the parameters, but it is difficult and crucial to achieve the best setting [48]. In texture analysis, the Gabor filter (GF) was first suggested by Turner [49] and Clark et al.

The wavelet transformation (WT) representation was also used as a defect detector [50, 51] with similar characteristics to the GT. WT is based on wavelets of varying frequencies, which are called wavelets for a short time and provides local information in every input image from the horizontal, vertical, and diagonal directions. Some methods have been successfully applied to realize a 98–100% success rate: fuzzy wavelet analysis [52], multi-scale wavelet method [53], WT image restoration scheme [50], and adaptive level selection scheme for analyzing the co-occurrence matrix [54] for defect detection.

Yang et al. [55] developed a feature extractor based on an adaptive wavelet and obtained fabric images based on a Euclidean distance detector. Subsequently, in [56], the authors outperformed the other five WT-based methods by using a new differentiated feature extraction (DFE) method, which realize a classification accuracy of 95.8%.

Model-based approaches

Methods based on statistical are relatively resilient to noise, whereas methods based on structural ignore local knowledge. There are bottlenecks in both approaches, such as messy defects on the texture surface and random background changes. Using the special structure model enhanced by parameter learning, the original texture distribution of the image block is projected to a low-dimensional distribution. Model-based methods can often be better used for detecting various defects. They can be classified into: fractal models, autoregressive (AR) models, and random field models. The following is a briefly discussion on several model-based methods.

The use of the term "fractal" was originally proposed by Mandelbrot [57]; fractal element is a natural geometrical element that is auto-similar and irregular. An exact replica or statistical copy of the whole object is a fragment of a fractal object, which can be extended out to match the whole object. The most noticeable dimensions in the fractal model are the fractal dimension and lacunarity. First the calculation of uncertainty and irregularity; second, the measurement of in-homogenization and systemic changes. The fractal approach performs well on the co-occurrence matrix, GF and MRF in the texture classification in a comparative analysis on Ohanian and Dubes [58]. In practical implementations, however the approach achieved only modest performance. Fractals can appear entirely different, but they can have the same fractal dimension. The fractal model is mainly suitable for natural textures with self-similarity.

AR model is generally based on the linear correlation of pixels [59] to characterize texture features. A multi-resolution pyramid was applied by Serafim [60] to natural image defect segmentation in leather based on a two-dimensional(2D) AR model. A defect detection model based on machine learnings was recently proposed by Zhang et al. [61] that automates report defects into real defect alerts and false alarms. A novel set of variable feature levels, called variable features, is designed to construct a classification model and select 13 basic classifiers as well as two holistic learning methods for the model building. The average accuracy rate realized was 83.36%.

The research of Cross and Jain [62] is based on the concept that texture has an interaction between related random variables in a random or periodic 2D field. Thus, an MRF was used as a texture model for the first time. The configuration of the random field can well reflect the image pixels' spatial similarity. Gayubo et al. [63] used an MRF, inspired by this idea, to restore flat steel defects and remove false features. In addition, to significantly minimize the false detection rate, Xu [64] used context-adaptive hidden Markov tree model; this model is based on the association of wavelet coefficients of flat

steel surface images on various scales to satisfy Markov ' s principle. Recent research results show the immense application potential of MRFs in industrial surface-defect detection.

1.2.2 Data-driven feature extraction methods

In the previous section, we discussed the approaches based on textural-based defect detection. In this section, we will discuss data-driven-based methods. These methods can be divided into two kinds on the basis of their processing dynamics, namely, supervised or unsupervised or semi-supervised methods, as shown in Table 1.2. the machine is trained using well "labeled," in supervised learning. This ensures that any knowledge is already labeled with the right response. This kind of learning is like learning in front of a supervisor. By contrast, in unsupervised learning, the data used for training are not labeled. Thus, the model is not supervised. Instead, it must be able to operate on its own to find knowledge.

Approach	Method
Supervised	K-nearest neighbor
	NNs/Deep learning
	SOM and SVM
Unsupervised/semi-supervised	Statistical/Novelty detection
	Gaussian mixture model

Table 1.2: Some commonly used defect detection methods based on supervised and unsupervised or semi-supervised classifiers

The supervised classification approach comprises a human model in which the application searches for the characteristics of a predefined class. Classifiers must be trained in advance to identify these features under supervision with the predefined detectable features. As an example of a supervised classifier, the K-nearest neighbor (KNN) classifier is a non-parametric learning algorithm, in which the output object of the class is classified, and its local neighborhood is used to make the prediction. The KNN algorithm is one of the simplest machine learning algorithms. Lopez et al.[65] used the KNN to classify ceramics. The image is tiled based on the chromaticity feature, and a high K value is used to realize an excellent performance. Neural networks can learn patterns from training datasets and classify new data based on previous information. Liu and Kang [102] proposed a two-layer feed-forward neural network to divide the pixels of the test image into defect and defect-free areas. A two-layer feed-forward neural network was proposed by Liu and Kang [66] to separate the pixels of the test image into defect- and defect-free regions. The idea here is that the role of defect identification is basically a concern with binary classification. However, substantial computational complexity arises from the vast number of parameters of the neural network. With the customization of the model parameters, convolution and secondary sampling in the convolutional neural

network (CNN) will effectively minimize the size of the model. Therefore, CNN-based architecture has been widely applied in automatic feature extraction [67] and image defect detection in industrial defect detection [68, 69]. A deep CNN for the identification of cracks in concrete and steel surfaces without measuring the defect characteristics was proposed by Cha and Choi [70]. This system is robust against the interference caused by the widely changing situation in the real world. SVM is also used to classify defective and non-defective areas as a generic linear classifier for data binarization classification. In [71], the author suggested that the classifier's effectiveness of defect detection depends on a combination of features and classifiers, and then merged the classifiers with different feature sets to categorize the test images as faulty and regular images.

Automated defect detection, particularly in practical industrial applications, has always been a difficult task. A large collection of tagged images is not always easy to collect, i.e. a collection of input vectors without any corresponding target values consists of the training images. Herein, unsupervised learning is committed to discovering similar sample classes in the input data. The primary purpose of the unsupervised classifier is to detect every unusual part. Detectors were trained on normal samples, and each deviation was considered abnormal. Generally, during the visual inspection, statistical parameter methods are used. Here, The Gaussian normal distribution of the data is the underlying assumption. A Gaussian mixture model with ideal max characteristics was suggested by Gururajan et al. [72] to detect a specific type of defect. A GF preprocessing method for reducing the complexity of structural signals and established an ultra-complete base set through sparse coding has been recently employed by Zhu et al. [73]. Mei et al. [74] developed a system that uses only normal samples for model testing based on unsupervised learning. This approach is applied by reconstruction with a convolutionary automated denoisation network of encoders of picture blocks at various levels of the Gaussian pyramid and then the detection effects are synthesized at different solutions.

1.3 Challenges of defect detection

While traditional detection methods have been very effective, a range of challenges remain to be solved. As the existence of defects in an image is very complicated (such as form, scale, contrast and orientation), one of the most difficult problems is to detect the different types of defect. Not only is the nature of the defect, but its background also increases the difficulty of defect detection. Defect detection under simple backgrounds has been well resolved; but under complex backgrounds, for instance, in the case of complex textures and an uneven contrast, many illusions can be created, which could easily be mistaken as defects. In general, the challenges of defect detection in this study can be summarized as follows.

- **Irregularity of defects** The image defects are very intricate in nature. It primarily comprises three aspects: the defect type, shape, and size. Fig. 1.5 presents

some examples of defects of different types, shapes, and sizes. As can be observed, some of the defects are large whereas other are small. Because the size changes unpredictably, it is difficult to set the size of the detection window, which makes detection difficult. In real-world industrial environments, the factors that cause defects are unpredictable, include thus, the defects caused by these factors can vary greatly in shape. Lines, arrows, circles and curves are some of the most common types. Besides these types, most of the defects are formless and hard to describe. It is impossible to use shape information to find it due to the complex shape of the defects. In addition, on real-world product surfaces, defects are usually slightly concealed, i.e., they have almost the same color and brightness as the defect-free area, which results in a low contrast between defects and defect-free area. The pixel-level positioning between defects and normal pixels has no clear boundary, which makes defect detection very challenging. To sum up, defect detection becomes a potentially challenging problem due to the complexity of defects and the subtle differences between defects and backgrounds.

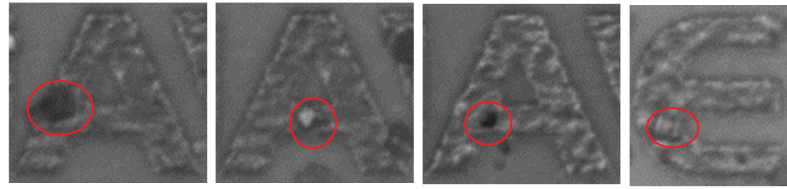


Figure 1.5: Examples of common types, shapes, and sizes of defects

- Irregularity of texture** The result of the inspection is not only affected by the nature of the defect itself, but also by the characteristics of the background. Products and thus the surfaces of products are usually made of materials such as wood, metal, and plastic, because these materials have a good appearance and feel. Wood is a popular material for the component. It has an irregular and complicated texture. Small defects in the texture can then be covered so that they are noticeable. Metal is also a common surface material. Small patterns are often engraved in the surface, which comprises bumps and dents. There are smooth metal goods, no sculpted styles, but the smooth surface is reflective. Engraved patterns or high reflectivity in metal surface can make defect detection very difficult. The other common product is plastic. Plastics are widely used in manufacturing because they are inexpensive and practical. They are usually colorful. For example, plastic bottles may come in more than a dozen colors and have complex pictures and text on them. As these surfaces have complex textures, high reflectivity, and colors, defect detection is challenging. Fig. 1.6 presents an example of the printed product used in this study.
- Fluctuation of illumination conditions** Realizing precise control of the actual

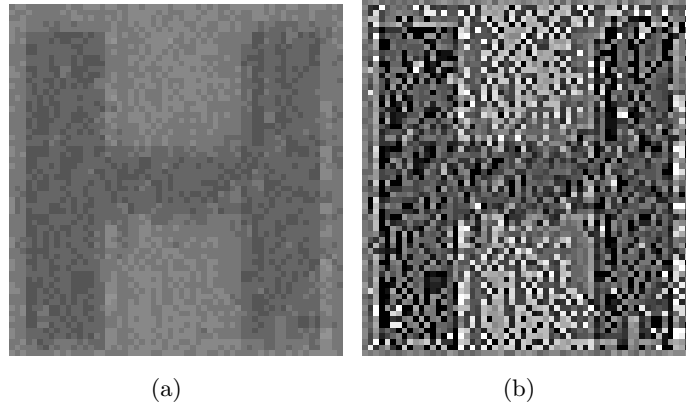


Figure 1.6: Example of a printed product. (a) original image and (b) image after sharpening

factory illumination conditions are difficult. Illumination conditions are mainly affected by two factors: one is the angle of the light source, and the other is the intensity of the illumination. The angle of the light source in the general factory is fixed at a well-adjusted angle. Therefore, we consider the second factor, i.e., the intensity of the illumination. Multiple sets or phases of visual inspection are normal in factories, whose state can vary over time. The following factors will induce illumination intensity changes.

- Natural light influences illumination conditions.
- Life of light sources is limited. Life is confined to light sources. We cannot ensure that the circumstances are as they were before the transition when we replace them.
- The lighting conditions of each manufacturing line cannot be guaranteed to be the same.
- Alternating current, allowing the light to fluctuate, is found in real lighting. The shutter speed of the camera used to detect is very high since assembly lines run very quickly. This results in a noticeable change in the illumination of the captured photo.

Furthermore, we deal with printed logotypes on a surface that is embossed with 3D micro-textures distributed randomly, as shown Fig. 1.7. The 3D micro-structures, which are uniformly embossed onto the surface, produce a faint shadow under illumination and appear as a random texture in the corresponding image. Illumination variations on certain surfaces significantly impact the appearance of certain surfaces and create issues in the identification of defects. From Fig. 1.7, we can observe that, even for the same object, the texture of the logo surface can be significantly different under different illumination intensities. This phenomenon poses great challenge for defect detection.

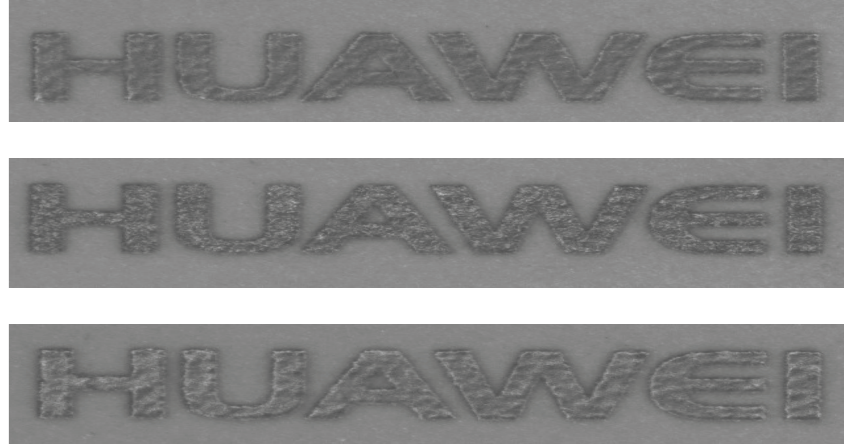


Figure 1.7: Imaging effect of the same object under various illumination conditions

- Lack of training samples** As mentioned earlier, many traditional methods comprise the use of some machine learning methods to extract the characteristics of defects for distinguishing them. Although the concept is appropriate, the quality of the training samples is closely associated with the results of the test. Therefore, the number of training samples should then be adequate, and as many conditions as possible should be covered by the samples. Therefore, the issue is how to collect sufficient samples. There exist two main problems in collecting sufficient samples. One is that the risk of errors is substantially decreased with the ongoing development of manufacturing technologies. Consequently, adequate defect samples are difficult to obtain. The collection takes substantial time even though adequate samples are obtained. Another problem is that the defect samples are required to be manually and individually marked with defect information. This work requires a large number of personnel with extensive testing experience. It is a time-consuming and expensive work. Another problem with respect to machine learning methods is poor portability. These methods all learn specific objects, and when they are applied to other objects, the detection results may be unsatisfactory. In summary, for many applications, collecting a sufficient number of defect samples to include the different defects is very difficult.
- Operational efficiency** The actual production line has a requirement for high detection efficiency. As the production becomes increasingly efficient, the time required for complete inspection of the product keeps decreasing. Even if the algorithm can provide good detection results, if the detection speed is slow, it cannot be used in actual industry applications. Therefore, when designing the algorithm, it is best to take into consideration the time complexity.

1.4 Motivation and contribution of the study

From the aforementioned discussion, we can observe that although the micro 3D texture surface appears uniform, it is actually full of random textures and fine particles owing to uneven material distribution. These random textures and fine particles are easily affected by light fluctuations during the imaging process. In addition, real-world defects such as scratches and stains may be minute and have a low contrast with the background. Furthermore, real-world flaws such as bruises and stains can be minute and have a poor background contrast. In addition, a shortage of adequate flawed samples. These circumstances make it very difficult to use traditional defect-detection methods for defect detection in surfaces and logotypes. In this dissertation, a robust defect-detection method called the multiple paired pixel consistency (MPPC) model for handling printed logotypes printed on micro 3D texture surfaces is presented to overcome the aforementioned issues. The MPPC model is established based on the orientation code of the image, instead of directly using the gray image. This is because the orientation code can greatly reduce the impact of light fluctuations; we use spatial information to find consistency in texture. Furthermore, a scheme called "position-dependent data inhibition" is proposed for improving the performance of the model, wherein two statistical techniques can be used for the detection and elimination of outliers. This scheme is expected to serve as a general type of data inhibition in other statistical systems.

This dissertation's contributions are summarized as follows.

- We realize a method of describing fundamental characteristics in pixels on logotypes in embossed surfaces by identifying a slight difference in statistics in both types.
- Robustness under illumination fluctuation is realized mainly via the use of orientation codes instead of directly addressing the brightness. Our model can thus tolerate fluctuations in illumination conditions.
- Our proposed model realizes the adaptive threshold adjustment function of the noise level by modeling coding noise and sensor noise. When we select the pixel pairs for the preparation of the mathematical models, we use an adaptive threshold dependent on the variance characteristic of the target selected. This is not possible with other methods. For example, when the contrast of the image is very poor, it is often accompanied by an increase in the noise level. Some methods cannot work well under such conditions; however, our proposed method can adapt to the change in the noise level and adjust the threshold. This is very important for defect detection.
- Sensitivity to small and vague defects can be realized. For instance, the single Gaussian model of pixel pairs is more sensitive, robust, and efficient than the Gaussian mixture model. The aforementioned sensitivity refers to the accuracy of

the defect detection. A new definition of a small differential operator can be used to raise the spatial sensitivity.

- Comparing with some possibility to use deep learning approaches, the proposed method is focused not only on the classification of defect images but also on the so-called pixel-wise extraction or measurement of the defect regions. We design a scheme for handling defective characteristics at the pixel scale. In addition, in our approach, only a small number of positive samples are used to classify or train pixel-wise elemental models, while in deep learning approaches, two balanced training sets of defect-free and faulty images may be required, which are typically difficult to obtain due to the excellent quality control recently realized.
- A modification performed using PDI can improve the robustness of the proposed method. In the training data, noisy data involving defects could inhibit. Therefore, the proposed method will also reduce the expense of preparing the training samples.

1.5 Organization of dissertation

This overall dissertation is structured as follows.

Chapter 1 introduces the importance of identification of defects and presents the associated works on defect detection. Some challenges in defect detection are discussed. Furthermore, the motivations and contributions of this study are explained.

Chapter 2 introduces the orientation codes (OCs), the use of which could reduce the influence of illumination fluctuations on defect detection; thus, OCs are used as the basis of the proposed method. First, we implement the original version of the OCs and then expand it by presenting two types of operations: a precise spatial differentiation for calculating the codes with a higher resolution and signed difference between any two codes as preparation for building up a more precise statistical model of their difference. Using these operations, we add a more reliable scheme to explain the statistical relationship between a pair of any logotypes pixels.

Chapter 3 introduces the proposed MPPC model in detail, including the fundamental principle and structure of the MPPC model. First, we observe the defect-free images of logotypes to determine the relationship between any pair of pixels in the logotype. We then introduce kurtosis to obtain the potential distributions. After analyzing the distributions, we explain how to pick the supporting pixel for each target pixel and finally construct the MPPC model for each pixel pair.

Chapter 4 provides a discussion of the method for using the proposed MPPC model of the relationship between pixel pairs in the defect-free logotype for identification of several types of logotype defects. Defect detection can be divided into two main stages. First, the status of each pixel pair is determined. We then identify the normal or abnormal states at the corresponding position defined by the elemental MPPC model.

Chapter 5 is focused on the PDI modification suggested on the basis of MPPC to

further strengthen the robustness of the MPPC and stabilize its performance. The basic principles and mechanism of the PDI are explored in depth in this chapter. Finally, we verify the capacity of the PDI by appropriate experiments.

Chapter 6 lays out the experimental setup in detail. In this chapter, comparative experiments for the MPPC and MPPC+PDI models using several real defective images and synthesized or artificial defective images under different conditions such as illumination fluctuation and different noise intensities are designed. We test the robustness and efficiency of our methods through these experiments.

The final chapter outlines the key points of this study and presents a discussion of our algorithms. Finally, the strategy and idea for possible future works are discussed.

Chapter 2. Improvement of Orientation Codes

Herein, we deal with a printed logotype on a surface embossed with randomly distributed 3D micro-textures, as shown in Fig. 2.1. This type of surface is made using embossing processes for creating tiny mixed convex and concave patterns on the surfaces of metal, plastic, or other materials. Because they have an attractive appearance, good handling, and excellent slip resistance, such surfaces have been widely used in numerous products. Their three dimensional (3D) micro-structures, which are uniformly embossed on the surface, produce slight shadows under illumination and appear as a random texture in an image. The changes in illumination on such surfaces have a significant influence on the appearance, which causes difficulties in defect detection. Therefore, we need to consider reducing the impact of changes in illumination conditions when designing algorithms.

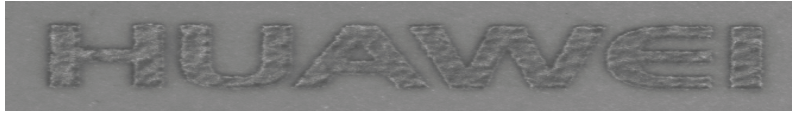


Figure 2.1: An example of logotype printed on embossed surface

Fig. 2.2 shows an example of printed characters that is defect-free and defective one, respectively. From the leftmost column, they are the raw image, gradient image, and OC image. And here we use three types of image, the original, the gradient and the OC images to represent these two characters. From Fig. 2.2, we could see that the original images and gradient images of typical defect-free and defective images have large difference or fluctuation in the whole part of logotype, which makes difficulty in its utilization for the defect detection. Instead of the gradient feature, we have proposed Orientation Code Matching (OCM) [75] as a robust matching or registration. Comparing the OC images of defect-free and defective parts shown in Fig. 2.2, we find there is a certain similarity in the defect-free part, and some prominent variation in the defective one. So, we use OC as the basis of our algorithm.

In this chapter, we first introduce the original version of the Orientation Codes (OCs) and then extend it by providing two types of operation, namely, a precise spatial differentiation for calculating the codes with a higher resolution and the signed difference between any two codes as a preparation for developing a more precise statistical model of their differences. Using these operations, we introduce a more precise scheme to describe

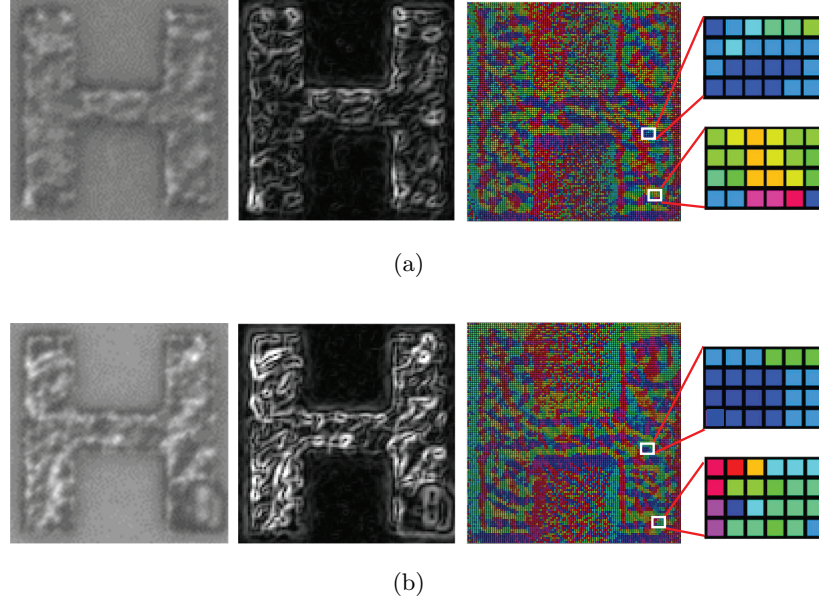


Figure 2.2: Example of printed character. (a) defect-free example and (b) defective example

the statistical relationship in a pair of any pixels on the logotypes.

2.1 Orientation codes

The orientation codes were proposed for filtering in order to extract robust features based on only orientation information involved in gradient vectors from any pictures. It has strong robustness to image representation and can resist the illumination fluctuation in the real world since it utilizes only the gradient orientation as a discriminative feature, which represents the orientation angle rather than the strength of gradient.

Let $I(i, j)$ be the brightness of pixel (i, j) . Then its partial derivatives in horizontal and vertical directions are written as $\nabla I_x = \partial I / \partial x$ and $\nabla I_y = \partial I / \partial y$, respectively. The orientation angle θ can be computed by using the $\tan^{-1}$ function. The expression is shown in following.

$$\theta = \tan^{-1} (\nabla I_y / \nabla I_x) \quad (2.1)$$

The actual orientation is determined after checking signs of the derivatives, thus making the range of θ to be $[0, 2\pi)$. Here, we use the Sobel kernels for calculating approximations of the horizontal and vertical gradient, which is based on convolving the image with two small, separable and integer valued 3×3 kernels in horizontal and vertical directions and is therefore computationally efficient for real-time system application:

$$\nabla I_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I(x, y) \quad (2.2)$$

and

$$\nabla I_y = \begin{bmatrix} 1 & 2 & 1 \\ -0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I(x, y) \quad (2.3)$$

where $*$ denotes the two-dimensional convolution operation. Then, The orientation code or OC is obtained by quantifying the orientation angle θ into N levels with a constant width $\Delta\theta (= 2\pi/N)$. The OC can be expressed as follows.

$$C_{i,j} = \begin{cases} \left\lceil \frac{\theta_{i,j}}{\Delta\theta} \right\rceil & |\nabla I_x| + |\nabla I_y| \geq \Gamma \\ N & \text{otherwise} \end{cases} \quad (2.4)$$

where Γ is a threshold level for ignoring pixels with low-contrast neighborhoods. That is, pixels with neighborhoods of enough contrast are assigned OC from the set $\{0, 1, \dots, N-1\}$ while we assign the code N to ignored pixels. In this paper $\Gamma = 10$ and $N = 16$. An example of a set of OCs is shown in Fig. 2.3. And Fig.2.4 shows an example of real

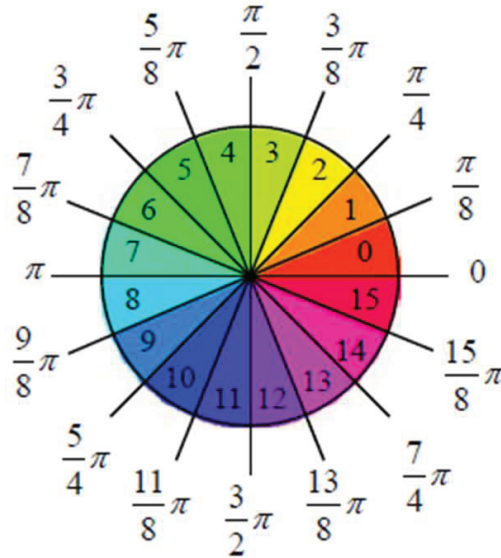


Figure 2.3: Sixteen-OC

printed logotypes image.

2.2 Signed difference in OC

Here we propose a somewhat new definition in OC space which is better suited for the more detailed statistical design than the original one. In contrast with the previous definition [75], the definition used here has not only positive differences but also negative ones. The negative part can contribute to a higher resolution. But with this comes a

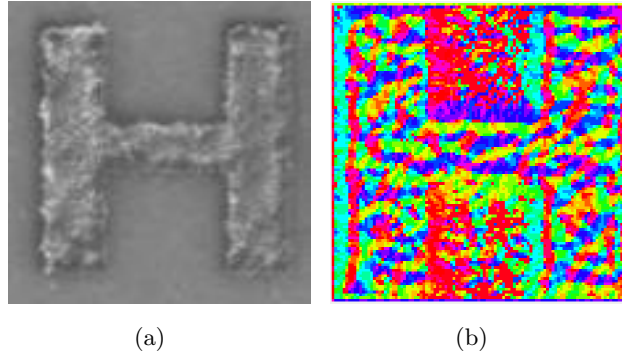


Figure 2.4: An example of real logotype image.(a) original image and (b) OC image

problem, because of the increased resolution, we need more pixels to achieve a stable histogram. We expect it to give a more complete and precise distribution of OC differences that facilitates statistical handling. The expression is

$$\Delta(a, b) = \begin{cases} \min\{|a - b|, N - |a - b|\} & (a - b) < -\frac{N}{2} \text{ or } 0 \leq (a - b) < \frac{N}{2} \\ -\min\{|a - b|, N - |a - b|\} & (a - b) \geq \frac{N}{2} \text{ or } -\frac{N}{2} < (a - b) < 0 \end{cases} \quad (2.5)$$

where a and b represent the orientation code to be compared or subtracted, for instance from a target and a reference image, respectively, and N shows the invalid-pixel code.

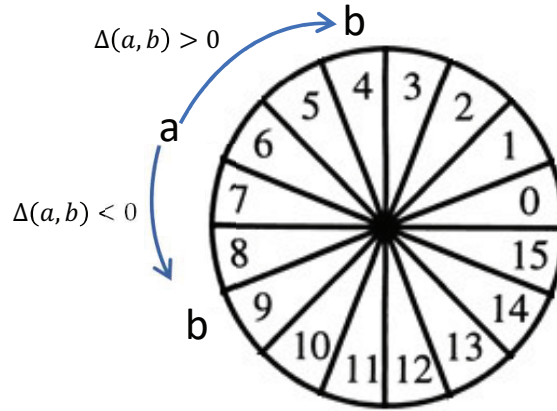


Figure 2.5: Scheme of signed difference in OC

Because our directional code design is circular, we have two values when we calculate the OC difference, which are $|a - b|$ and $N - |a - b|$. In the definition of the magnitude of the difference, we use the minimum difference to represent the difference between the two orientation codes. In addition, the definition of positive and negative values of the difference value is shown in Fig. 2.5 which schematically shows the above-mentioned definition of the difference which can distinguish between clockwise and anti-clockwise orientation from a to b along the circle defining any angles by considering cyclicity.

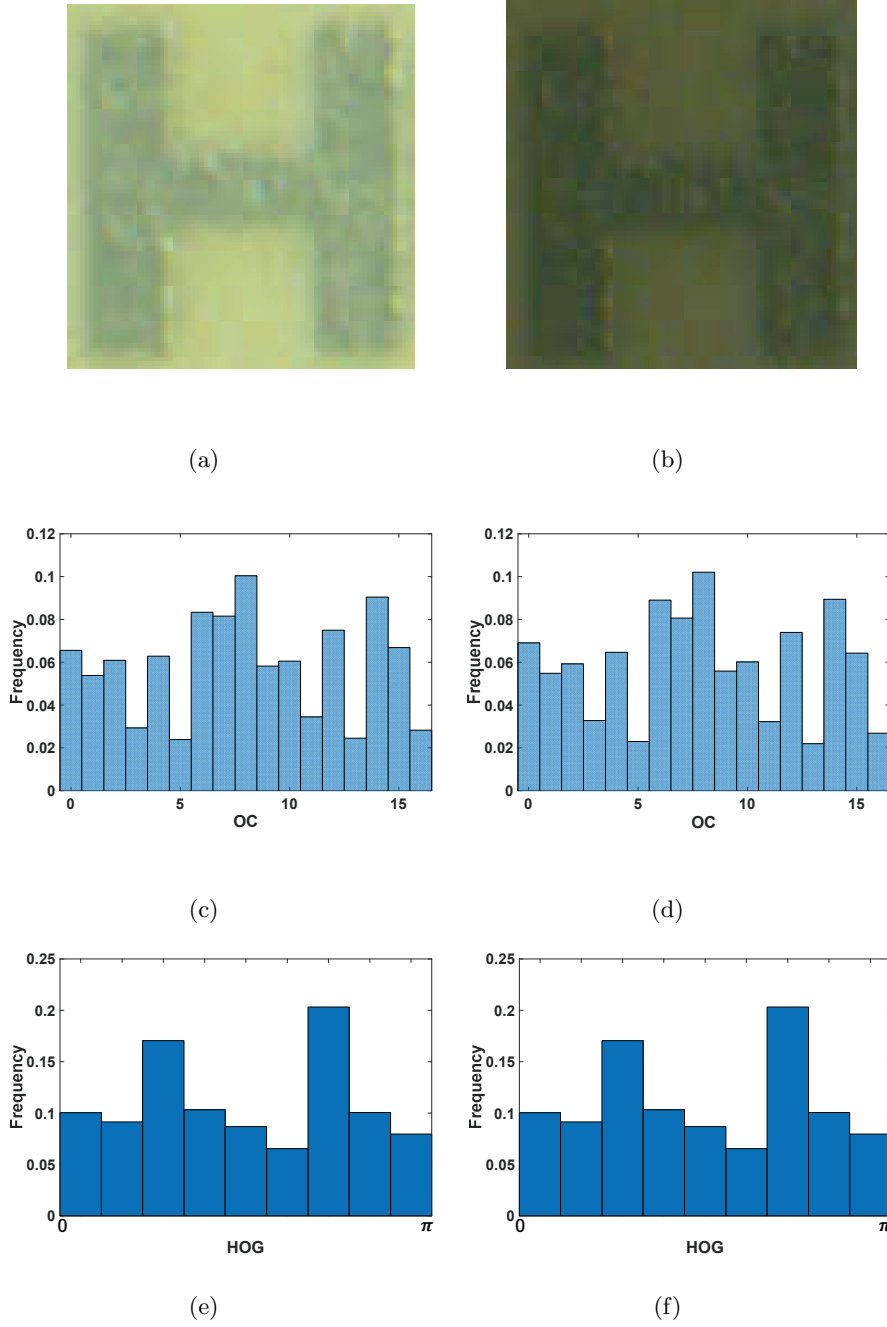


Figure 2.6: An example is the printed character under different illumination conditions. (a) and (b) the character under strong and low illumination, (c) and (d) the corresponding OC histogram, and (e) and (f) are the corresponding HOG histogram.

A similar scheme, the histogram of gradient (HOG) [76], was proposed and successfully applied in many researches. Comparing with it, OC is rather simpler and has some merits. In the OC approach, we do not require any gradient strength information but only orientation angles. We experimented with two pictures of the same object under

different illumination conditions as shown in follows. Then use OC and HOG to create histograms for the two images, respectively. The histograms are shown below.

From the histogram, we can see that even if the illumination changes drastically, the histograms formed by the two methods almost no change at all. Furthermore, we introduce the intersection to compare the similarities of the two histograms produced under the same method. The similarity of OC is 0.9828, and the similarity of HOG is 0.9253. Here we can see that the two methods performed very close, and they are very robust to illumination fluctuations. From these fundamental experiments, we can get that they have some similar performances in robustness to illumination fluctuation. In the calculation of angles, while HOG has $[0, \pi]$, OC has the range of $[0, 2\pi]$, where we aim to detect any defect rather finely by use of this wider resolution in combination with a more precise differentiation operator shown in the next section.

2.3 Precise differentiation

In general, the Sobel operators [77] of larger domains, for instance, three by three or five by five, give a stable estimation of the direction angle. Here, we utilize three by three kernels as an example. The operator uses two three by three kernels which are convolved with the original image to calculate approximations of the derivatives, one for horizontal changes, and one for vertical. If we define $\mathbf{A}$ as the source image, and G_x and G_y are two images which at each point contain the vertical and horizontal derivative approximations respectively, the computations are as follows:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \mathbf{A} \quad (2.6)$$

and

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ -0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A} \quad (2.7)$$

However, in this paper, we aim to detect very small defects in the real production from factories, where their manufacturing processes continue to advance, and the defects occurred in any actual production lines become smaller and smaller. Therefore, we proposed a new gradient or differentiation operator for getting a higher resolution or definition to realize better spatial sensitivity rather than stability.

In the next equation 2.8, we define two convolution matrices for the gradient operator in a two by two region.

$$G_x = \begin{bmatrix} -1 & 1 \\ -2 & 2 \end{bmatrix} * \mathbf{A} \quad (2.8)$$

and

$$G_y = \begin{bmatrix} 2 & 1 \\ -2 & -1 \end{bmatrix} * \mathbf{A} \quad (2.9)$$

In the definition of convolution matrices, a pixel of interest is defined at the lower left position in this two by two region and it may have the weight of -2 and then the neighboring pixels along the x and y axes have the one of +2 while the neighboring pairs of pixels have -1 or +1 weights for getting some smoothing effect for de-noising.

Fig. 2.7 shows three typical examples for evaluating the sensitivity by comparing the proposed operator with Roberts operator of two by two domains as a representative small operator and Sobel operator of three by three domains by use of a standard image Lena. Comparing the results in part 2 and 3 especially, we see the proposed operator could represent rather finer structures in their corners, lid, and lashes than Roberts operator and Sobel operator, while we could have not so significant increase in the noise levels.

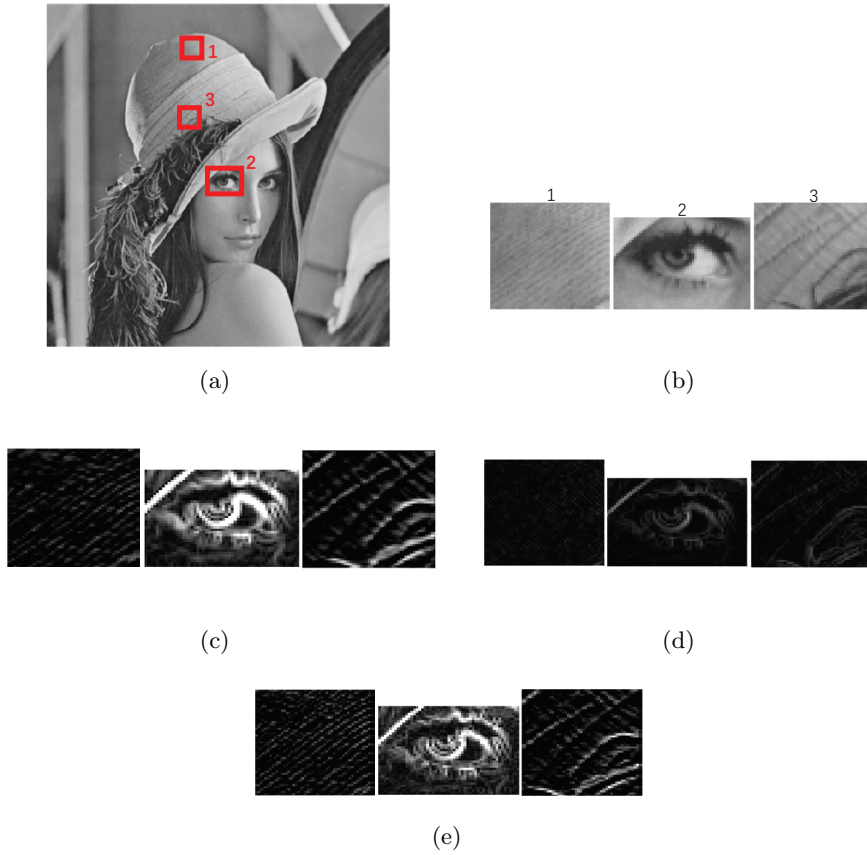


Figure 2.7: Examples of applying the proposed, the Sobel and the Roberts operator. From (a) to (e), Lena image, the magnified parts, the edge detection result by Sobel, by Roberts, and by the proposed operator.

2.4 Summary

In this chapter, we extend the original version of orientation codes by providing two types of operation, namely, a precise spatial differentiation for calculating the codes with a higher resolution and the signed difference between any two codes as a preparation for developing a more precise statistical model of their differences. Using these operations, we introduce a more precise scheme to describe the statistical relationship in a pair of any pixels on the logotypes.

- Firstly, we introduced why we utilize the orientation code as the basis of our method. After coding with OC, we found a certain similarity in the defect-free regions and a great difference in the defective regions.
- Then, we presented the original version of orientation code, which has strong robustness to image representation and can resist the illumination fluctuations proved in many experiments.
- Third, we proposed a somewhat new definition in OC difference which is better suited for the more detailed statistical design than the original one.
- Finally, a new gradient or differentiation operator was proposed for getting a higher resolution or definition to realize better spatial sensitivity.

Chapter 3. Multiple Paired Pixel Consistency Model (MPPC)

3.1 Overview

In this chapter, we propose a novel framework to build a defect-free model for defect detection, which can robust against the illumination change, such as highlighting and shadowing. We name it Multiple Paired Pixel Consistency [78].

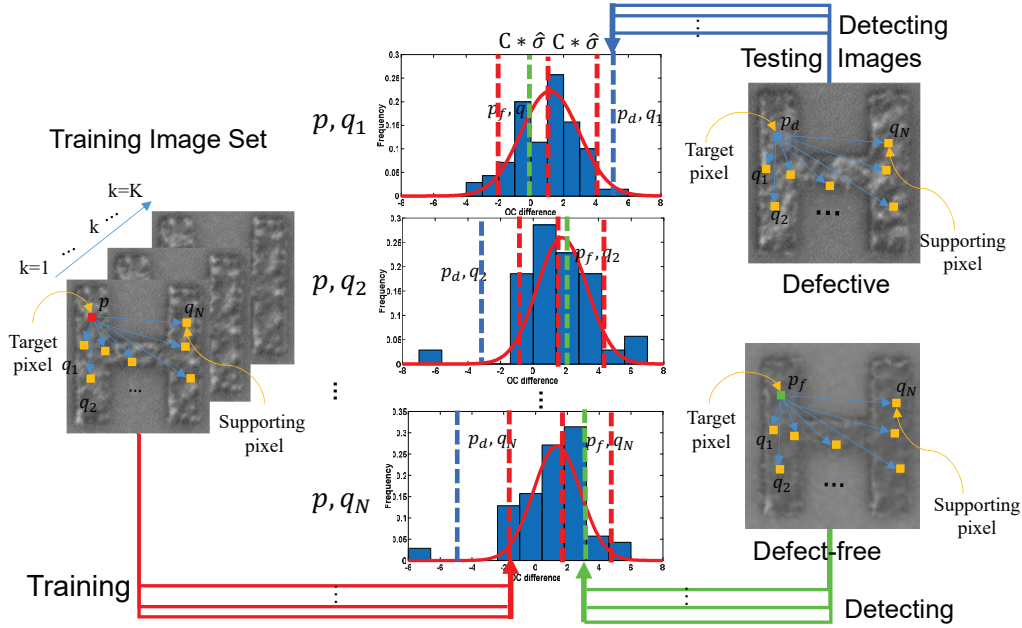


Figure 3.1: The framework for proposed method

In this study, in order to model the defect-free image, spatial pixel pairs with a great degree of consistency are employed to represent each other by using the stable OC differential increment between a pixel pair which is much more reliable than the OC of a single pixel.

A pixel pair consist of each pixel itself and a selected pixel. Hereafter we call the pixel itself as *target pixel* and the selected pixel as *supporting pixel*. The target pixel refers to

all pixels in a logotype area. The supporting pixels are neither randomly selected in the scene, nor defined in advance around the target pixel. Instead, the supporting pixels are selected based on their statistical stability with the target pixels.

The proposed method consists of two major components, as illustrated in Fig. 3.1: training stage and detecting stage. Training stage can be divided into two steps. They are to select supporting pixels which has a great degree consistency with target pixel, Pearson product-moment correlation coefficient is used to quantify the consistency between the pixel pairs. And to make the model of each pixel pair. Here, we use single Gaussian model to fit. Detecting stage also can be divide into two steps. They are to identify whether each pixel pair match its model, and then to determine the status of each target pixel.

This chapter is organized as follows, Section 3.2 introduces a basic experiment for observation of defect-free image. Section 3.3 explain the detail of modeling the defect-free image.

3.2 Observing defect-free images of logotypes

We recall that our aim is to detect defects in logotypes on embossed surfaces which may behavior as random textured ones. Since any logotypes are printed in the same pattern, in spite of the randomness of the surfaces, it is expected for them to have a similar texture at the same positions as a fundamental assumption for the design of a statistical model.

As the first step for modeling the statistical behavior of pixels on logotypes on embossed surfaces, we consider the relationship between any pair of two pixels both on the logotype. In this dissertation, the objects have not so strong colors, so we handle their monochrome versions here. By use of the precise operator of differentiation and the orientation codes mentioned above, we can expect to extract a stable statistical characteristic even under fluctuating illuminations.

Fig. 3.2 shows two pairs of two pixels of which one pair contains two pixels both on the logotype but another one consists of a pixel on the logotype and one on the background. Here our interest is to investigate what kind of relation any pair of two pixels in different places has in orientation code difference. We randomly select P on logotype as a target pixel, an arbitrary pixel Q_L on the logotype part, and an arbitrary pixel Q_B in the unprinted or background part.

Fig. 3.3 shows two typical examples of orientation code difference histograms, which were made from a data set of 170 logotype images. We can see that one of the logo-background pairs (P, Q_B) as an example of typical logotype defects, such as lack of printing, in Fig. 3.3(b) has a wide variance rather than the one in Fig. 3.3(a) for the logo-logo pairs (P, Q_L) , where the profile has a clear peak as another prominent feature in comparison with the latter one, and we also find a symmetric property around the mean or center value. From this observation and the rareness of defects in real factories,

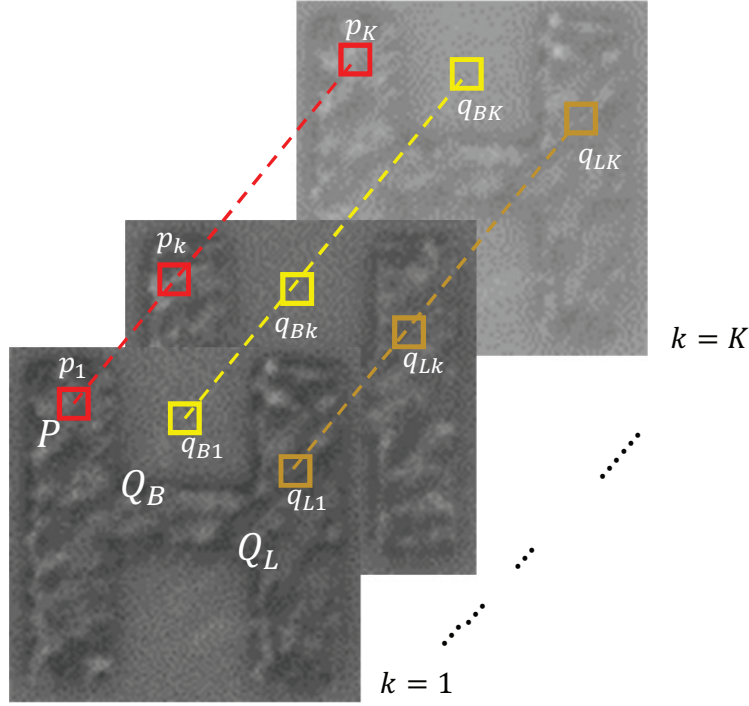
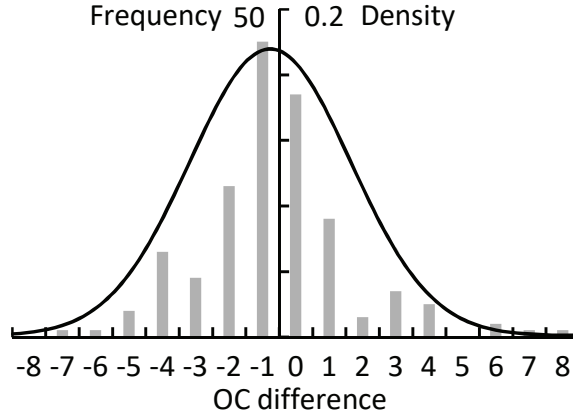


Figure 3.2: Observing and comparing OC difference histograms

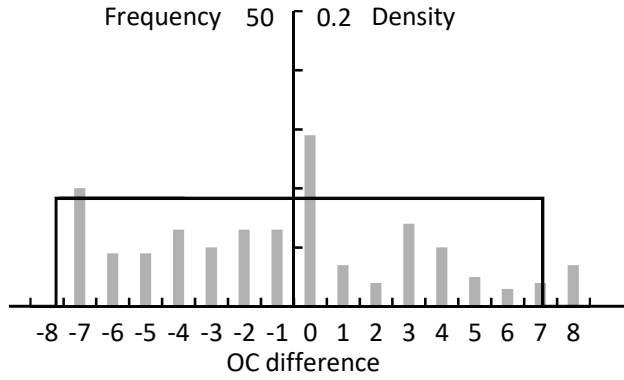
we assume that they basically can be fitted by a single Gaussian distribution.

Then we want to confirm the rationality of our above hypothesis from the following two aspects. One is we want to use statistics to analyze whether it is reasonable for us to fit a single Gaussian distribution model. Here we utilize the kurtosis to analysis. The other is that the generality of the phenomenon that the OC difference of logo-logo pairs found above can have an obvious peak and are symmetrical around the middle value. Their kurtoses are calculated for finding their potential distributions. We found that the kurtosis of the histogram in Fig. 3.3(a) was 0.0065 that is very close to 0, which makes it reasonable to fit or model it by a single Gaussian distribution. The histogram can be modeled by the single Gaussian distribution $\frac{1}{2.3\sqrt{2\pi}} \exp\left(-\frac{(x-(-0.65))^2}{2*2.3^2}\right)$ with their mean -0.65 and the standard deviation 2.3. On the other hand, for the histogram of logo-background pairs (P, Q_B) in Fig. 3.3(b), the calculated kurtosis was -1.2, which shows it can be modeled by a uniform distribution. Fig. 3.3(b) can be fitted by the uniform distribution $\frac{1}{6.275-(-8.275)}$ since their mean and, variance were -1 and 17.6, respectively.

In order to verify the universality or reproducibility of the above mentioned phenomena, for randomly selected ten target pixels P we constructed ten sets of 20 pixels Q , each of which included ten of the suitable pixels and another ten of the unsuitable ones based on the covariance-based measure introduced γ defined in Eq. 3.8. Fig. 3.4 shows the distribution of their kurtoses with respect to the γ values, resulting in the two



(a)



(b)

Figure 3.3: Two representative types of OC difference in pair. (a) and (b) distribution of OC difference for logo-logo pair, logo-background pair.

clear clusters of the suitable or highly correlated pairs and the unsuitable pairs of P and Q , each of which included 100 kurtoses. The former cluster located at the upper right around 0 shows that they have Gaussian distributions as their statistical models, while the lower left one around -1.2 for the latter cluster shows their models must be uniform distributions.

We utilize the signed difference introduced in Section 2.3 in these statistical analyses of histograms and these models have very important roles in the next section. From these observations, we ascertain the following very different behaviors of these pairs: logo-logo pixel pairs may generally have a very high correlation due to logotype printing, while logo-background pairs have no clear correlative relation. We expect the latter kind of pair may possibly be used to represent typical defects, such as lack of printing.

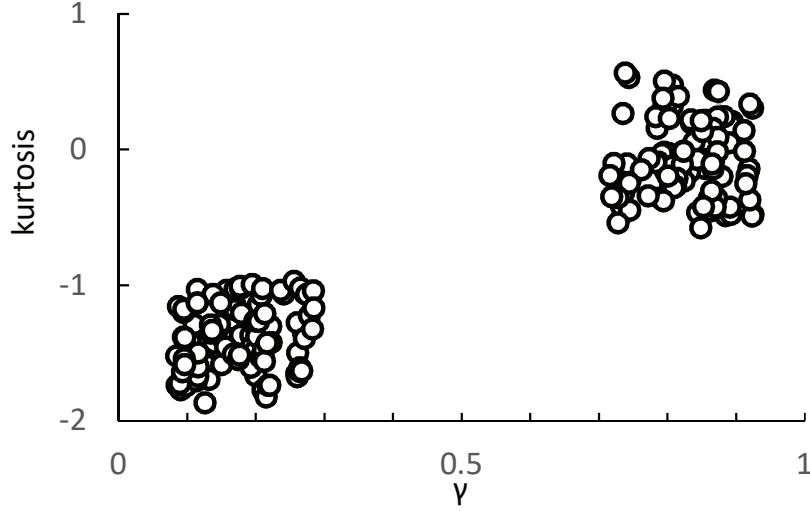


Figure 3.4: Kurtosis distribution for two types of pixel pairs

3.3 Training stage of MPPC

In this section, we introduce the training stage of proposed method based on previous introduced observation. The distribution between target pixel P and arbitrary pixel Q_L in logotype area has an obvious peak and symmetry profile which can be modeled by a single Gaussian distribution. But the Fig. 3.3(b) shows the histogram from the target pixel P and arbitrary pixel Q_B in background part which has neither an obvious peak nor symmetrical relation.

Here, we use Orientation code image rather than gray-scale image. Since Orientation code can be strongly robust against illumination fluctuation. Because in the real-world scene the illumination condition is difficult to control.

Fig. 3.5 shows the fundamental definitions of the image data. Suppose we are given a training image sequence $B = \{I_1, I_2, \dots, I_K\}$ with a total of K images, and each image has $M = U \times V$ pixel positions. In the three-dimensional space $\Gamma = \{(u, v, k) \mid 1 \leq u \leq U, 1 \leq v \leq V, 1 \leq k \leq K\}$, we have $U \times V \times K$ intensity values within a OC value range $[0, N]$.

In the following, the OC value of each pixel position varies with different product are regarded as samples from a stochastic process. We define P as a target pixel at location (u, v) . The location of P varies to cover all pixels of an image, and its orientation code sequence varies with different product is denoted as $\{p_k(u, v)\}_{k=1,2,\dots,K}$. In the same way, we define $C(u', v')$ as arbitrary pixel with orientation code sequence $\{c_k(u', v')\}_{k=1,2,\dots,K}$ at the location (u', v') . We have omitted most (u, v) and (u', v') in the following discussion.

Fig. 3.6 shows the schematic structure of the proposed model ‘Multiple Paired Pixel

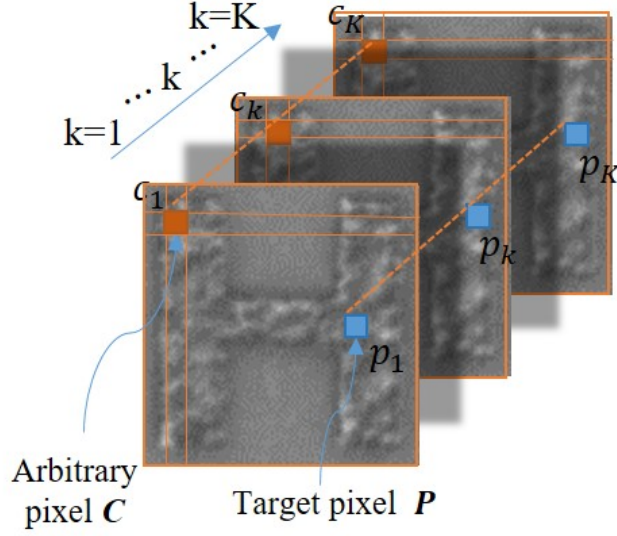


Figure 3.5: Fundamental definition of image data. Target pixel P and an arbitrary pixel C with their OC sequence varies with different product p_t and q_t

Consistency (MPPC)' which can represent one statistical characteristic in orientation code difference between two elemental pixels in the pairs. The main idea of this statistical modeling of pictures is called 'CP3' that has been proposed previously by Liang [79] for robust background subtraction. We propose an extension of this scheme by introducing the cohesive relationship of orientation codes in logo-logo pairs which are defined between each target pixel P on a logotype and the set of its supporting pixels also on the same logotype, which should be selected to have higher consistency or correlation with the target pixel. In other words, all of them have similar trends of change as the target pixel has, for which we can make a statistical model by fitting a single Gaussian distribution to the orientation code difference histogram of these pairs of high consistency.

3.3.1 Selection of supporting pixels

For modeling a relation between any pixels in defect-free logo, we use spatial pixel pairs with high consistency. A pixel pair consists of each pixel itself (target pixel) and a selected pixel (supporting pixel). And the target pixel refers to all pixels in the logotype area.

For formalization in this paper, we use capital letters in bold face, such as $\mathbf{Q}$, to represent any set, simple capital ones to represent any pixel, lower ones to show any orientation codes of the pixels, respectively.

In this section, we introduce how to select the supporting pixels from the $M - 1$ candidate pixels for each target pixel P . For an arbitrary pixel pair (P, Q) , we can have

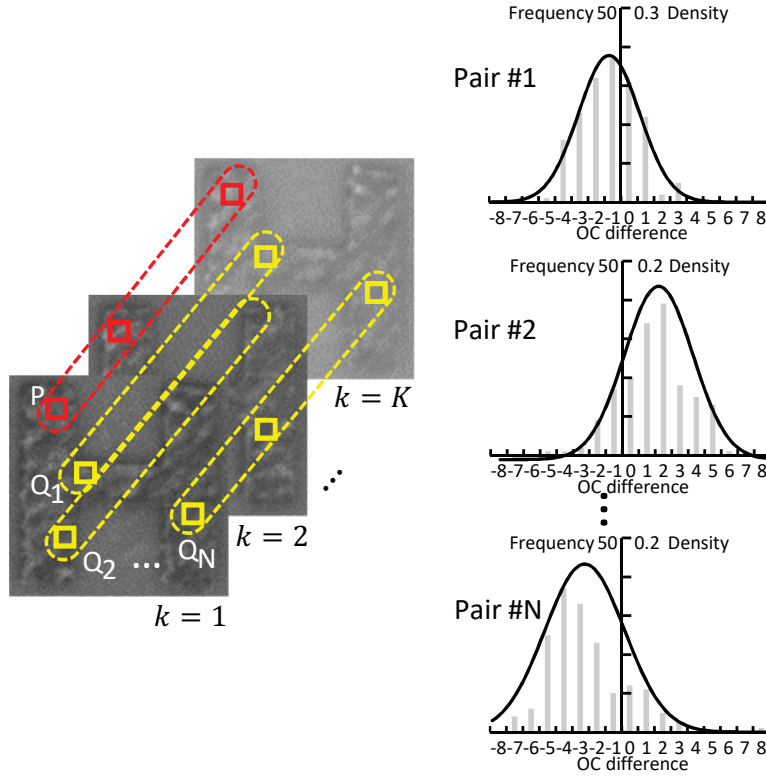


Figure 3.6: Scheme of MPPC modeling

two OC sequences as follows.

$$P = \{p_1, p_2, \dots, p_K\} \quad (3.1)$$

and

$$Q = \{q_1, q_2, \dots, q_K\} \quad (3.2)$$

The expectation value of P and Q are

$$\bar{P} = 1/K \sum_{k=1}^K p_k \quad (3.3)$$

and

$$\bar{Q} = 1/K \sum_{k=1}^K q_k \quad (3.4)$$

And their variances are

$$\sigma_P^2 = \frac{1}{K} \sum_{k=1}^K (p_k - \bar{P})^2 \quad (3.5)$$

and

$$\sigma_Q^2 = \frac{1}{K} \sum_{k=1}^K (q_k - \bar{Q})^2 \quad (3.6)$$

The covariance of (P, Q) can be defined as following:

$$C_{P,Q} = \frac{1}{K} \sum_{k=1}^K (p_k - \bar{P}) (q_k - \bar{Q}) \quad (3.7)$$

If $C_{P,Q} > 0$ means P and Q has positive covariance value, which indicates they may have a high consistency probability. We call this kind of pixel pair co-occurrence pixel pair hereafter. And in order to measure the independent consistency quantitatively, we use Pearson product-moment correlation coefficient:

$$\gamma_{(P,Q)} = \frac{C_{P,Q}}{\sigma_P \cdot \sigma_Q} \quad (3.8)$$

where σ_P and σ_Q are the standard deviation of the OC sequence P and Q , respectively. Eq. 3.8 is used to estimate the linear dependence of spatial pixel pairs, and $\gamma_{P,Q}$ is ± 1 which represent the case of a perfect positive/negative linear correlation. And this normalized correlation coefficient does not involve the restraint between P and Q . Therefore, the pixels Q with any mean values are likely to have large $\gamma_{P,Q}$ with P . And $C_{P,Q}$ is the covariance of P and Q .

And in the practice, Eq. 3.8 can be calculated based on a correlation matrix instead of calculating pixel-by-pixel serial processing. The correlation matrix is the covariance matrix of the standardized random variables $\tilde{p}_k = p_k / \sigma_P$. With total of M pixel positions in logotypes, the image sequence can be arranged progressively as a column vector set $\chi^M = \tilde{p}_k(m)_{m=1,2,\dots,M}$.

The correlation matrix is as follows

$$\begin{aligned} \gamma(\chi^M) &= C(\chi^M, (\chi^M)^T) \\ &= \begin{bmatrix} \tilde{p}_k(1) - \bar{P}(\tilde{p}_k(1)) \\ \tilde{p}_k(2) - \bar{P}(\tilde{p}_k(2)) \\ \vdots \\ \tilde{p}_k(M) - \bar{P}(\tilde{p}_k(M)) \end{bmatrix} \cdot \begin{bmatrix} \tilde{p}_k(1) - \bar{P}(\tilde{p}_k(1)) \\ \tilde{p}_k(2) - \bar{P}(\tilde{p}_k(2)) \\ \vdots \\ \tilde{p}_k(M) - \bar{P}(\tilde{p}_k(M)) \end{bmatrix}^T \end{aligned} \quad (3.9)$$

where $C(\cdot)$ is the covariance operation. The correlation matrix is symmetric so that each row and each column of the $\gamma(\chi(M))$ is an array of $\gamma_{P,Q}$ for each target pixel $P(u, v)$.

For each target pixel $P(u, v)$, we can have $M - 1$ candidate pixel at difference positions, where M shows the logo image size. Namely, we need calculate $M - 1$ values of $\gamma_{(P,Q)}$. Then the supporting pixels Q_n can be selected in the array $\gamma_{(P,Q)}$ which has the highest N components. The expression of supporting pixels is in the following.

For each target pixel $P(u, v)$ with $M - 1$ values of $\gamma(P(u, v), Q_i(u', v'))$ need to be calculated. Pixel $Q_i(u', v')$, having a high value of $\gamma(P, Q_i)$ with target pixel $P(u, v)$,

can be selected as the reference block Q_n . Collectively, the reference blocks that satisfy the requirement of having the highest N components in the array $\{\gamma(P, Q)\}$ can be nominated as the candidate blocks. Taking the candidates $Q_1, Q_2 \in Q_i$ by way of example, we order them so that $Q_1 > Q_2$, if $\gamma(P, Q_1) > \gamma(P, Q_2)$.

The supporting pixel set $\{Q_i\}_{i=1,2,\dots,N}$ is defined as follows:

$$\mathcal{Q} = \{Q_i(u_i, v_i) \mid \gamma_{\mathbf{P}, \mathbf{Q}_i} \geq \gamma_{\mathbf{P}, \mathbf{Q}_{i+1}}\}_{i=1,2,\dots,N} \quad (3.10)$$

where N is the parametric number of supporting pixels Q_n for each target pixel P . And the selection threshold of supporting pixel is an adaptive threshold depending on the variation of P . In general, we can expect that if the pixel pair (P, Q_i) keeps a high correlation coefficient, then the supporting pixel Q_i can provide some reliability to estimate the current state of the target pixel P .

3.3.2 Defect-free model of pixel pairs

In this section, we discuss how to produce a limited number of supporting pixels Q_n^P and build the model.

In the Section. 3.3.1, the selected Q_n using an adaptive threshold resulted in an indeterminate number of N . In order to propose an executable algorithm, we need to confirm a limited number of supporting pixels. Because the spatial distribution of Q_n is not regular, we cannot choose a few points based on spatial methods to represent Q_n^P from Q_n . In order to solve this problem, we introduced K-mean clustering method. Firstly, we utilize K-means clustering to divide Q_n to K classes. And we use the nearest pixel from the center of the class to represent Q_k^P .

In the previous section, we have designed how to choose supporting pixels. And from many experiments, we know that the histogram of OC difference between the target pixel P and supporting pixel Q , both of which in logotype part has an obviously peak. And it can be modeled by a single Gaussian distribution model. According to this feature, we assume that for each supporting pixel Q_n keeps a bivariate OC differential increment with target pixel P ,

$$\Delta(P, Q_n) \sim N(\hat{\mu}_n, \hat{\sigma}_n^2) \quad (3.11)$$

and we utilize the Eq. 2.5 to calculate the statistic relation of OC difference in the pixel pair. The estimated parameters $\hat{\mu}$ and $\hat{\sigma}$ are represented as,

$$\hat{\mu}_n = \frac{1}{K} \sum_{k=1}^K \Delta(p_k, q_k) \quad (3.12)$$

$$\hat{\sigma}_n = \left(\frac{1}{K} \sum_{k=1}^K (\Delta(p_k, q_k) - \hat{\mu})^2 \right)^{\frac{1}{2}} \quad (3.13)$$

After the training stage, the above two parameters $\hat{\mu}_n$ and $\hat{\sigma}_n$ are recorded for the following detection stage. And the defect-free model is a look-up table consisting of $[u'_n, v'_n, \hat{\mu}_n, \hat{\sigma}_n]$ for supporting pixel $\{Q_n^P\}_{n=1,2,\dots,N}$. Another two parameters u'_n and v'_n represent the position of supporting pixel Q_n^P . The pseudo-code of training stage is shown in Algorithm 1.

Algorithm 1: MPPC for defect-free modeling

Input: $B = \{I_1, I_2, \dots, I_K\}$ with total K images
Output: look-up table of $\{Q_k^P\}_{k=1,2,\dots,K}$ consisting of $[u'_n, v'_n, \hat{\sigma}_n, \hat{\mu}_n]$

- 1 **Initialization**
 - 2 Build the vector set $\chi^M = \{\tilde{p}_k(m)\}_{m=1,2,\dots,M}$
 - 3 **Compute correlation matrix:**
 - 4 $\gamma(\chi^M) = C(\chi^M, (\chi^M)^T)$
 - 5 **Select supporting pixels**
 - 6 **for each** $P(u, v)$ **do**
 - 7 Computer γ based on Eq. 3.8.
 - 8 $\mathcal{Q} = \{Q_i(u_i, v_i) | \gamma_{\mathbf{P}, \mathbf{Q}_i} \geq \gamma_{\mathbf{P}, \mathbf{Q}_{i+1}}\}_{i=1,2,\dots,N}$
 - 9 Compute $\hat{\mu}_n$ based on Eq. 3.12
 - 10 Compute $\hat{\sigma}_n$ based on Eq. 3.13
 - 11 And record $[u'_n, v'_n, \hat{\sigma}_n, \hat{\mu}_n]$ for $\{Q_n^P\}_{n=1,2,\dots,N}$
-

3.3.3 Computation time in training

Because our algorithm is applied in the actual production line, so the computational efficiency of the algorithm will have certain requirements. In this section, we will discuss the issues of our proposed method's computation time. We will discuss the following two aspects, one is based on the image, one is based on the pixel.

Now we introduce the computation time in training stage. In the training stage, we utilize 100 defect-free images to train the defect-free model. The size of each picture is 110 * 111 pixels. Here we propose two versions of the program. One is written in MATLAB, the other is written in C++. Their development environment is MATLAB 2016a and Visual Studio 2013, respectively. In C++ version, we use the library of Opencv2.4.11. And Table. 3.1 shows the hardware information of used computer. The CPU of the computer used is Inter Xeon 3.5GHz, and the memory is 64G.

In the MATLAB version, we use the timing function in MATLAB to timing. In the c + version, we used two timing functions for timing. One is OpenCV timing function, the other is QueryPerformanceCounter function. In the MATLAB version, the entire training process took 9548.69 seconds. In the C++ version, the entire training process took 157.33 seconds which use OpenCV timing to timing and took 153.45 seconds which

use QueryPerformanceCounter function to timing. Converting to pixel level, training stage takes 0.78 seconds per pixel and 100 images in MATLAB, 12.8 milliseconds per pixel and 100 images in C++.

Table 3.1: Hardware information

CPU	Intel(R)
	Xeon(R) E5-1620 v3 @3.50 GHz
Memory	64G

3.4 Summary

In this chapter, we proposed a novel framework to build a defect-free model for defect detection called Multiple Paired Pixel Consistency (MPPC), which can robust against the illumination fluctuations.

- Firstly, we utilize kurtosis to determine the potential distribution and analyses the statistical relationship of any pair of pixels.
- And then we based on this knowledge presented detailed introductions of the modeling procedure of MPPC.
- With the dispersed distribution in structure, MPPC can reduce the likelihood of erroneous detection and lead to more reliable and robust results in foreground detection, this will be verified in the next chapter.
- Finally, we presented the computation cost in training stage.

Chapter 4. Defect Detection

In this chapter, we now discuss how to utilize the proposed MPPC model of the relationship between pixel pairs in the defect-free logotype for detecting many sorts of logotype defects. Since the MPPC model can represent the statistical behavior of the relation of an individual target pixel to the supporting pixels around it, we utilize it to statistically test whether a target pixel is recognized as a reasonable sample from the distribution registered in the LUT or not. The scheme for defect detection algorithm is shown in Fig. 3.1. In this stage, we can divide it to the following steps: (a) the step to identify whether the pixel pair (P, Q_n^P) matches its Gaussian model. (b) the one for observing the matching ratio, namely, to count the number of pixel pair (P, Q_n^P) that can match their Gaussian models from total N supporting pixels for each target pixel P .

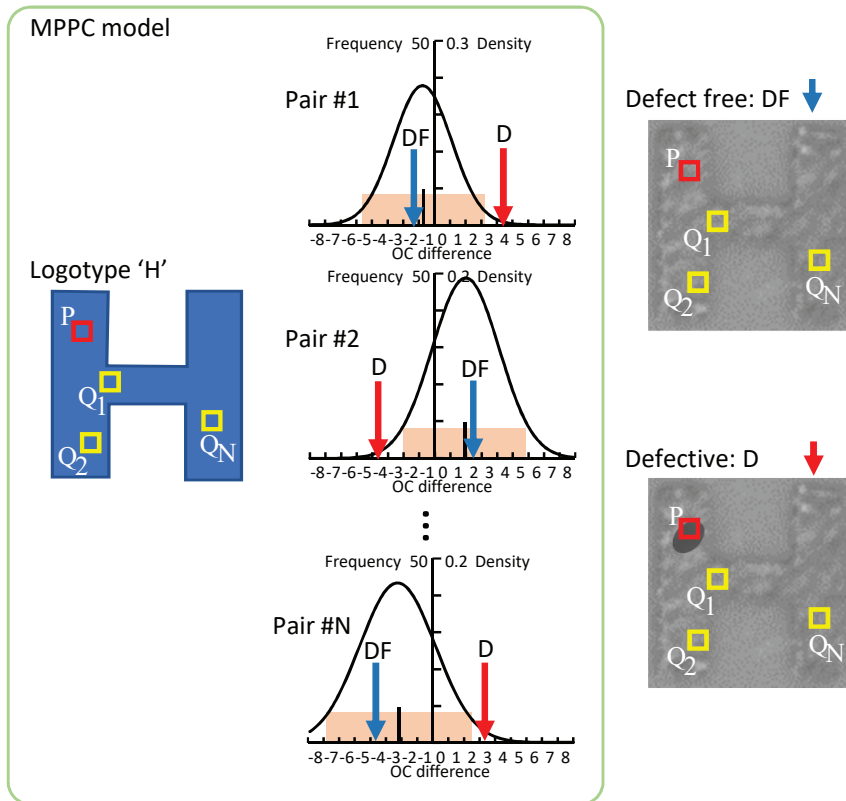


Figure 4.1: Scheme of defect detection by the MPPC model

Defects have several types, such as dust or particles, scratches, misprinting, hairs, spits, etc., of which characteristics are randomness in alignment, texture, shape, or size. Because of high performance quality control in production lines, these defects may be very small, and furthermore we should also keep in mind their very low probability of occurrence. In addition, in this study, we need to handle the randomness in embossed surfaces too. Thus, these fundamental characteristics lead us to approaches based on pixel-wise evaluation for detection. Afterward, we may proceed to some next steps to recognize them as aggregated regions that reveal some common characteristics. From these considerations and the investigation in Section 3, the two features of our MPPC model, spatial sparseness and high consistency in correlative relation, can be utilized for handling defects. The former may prevent any defect from occupying P and some supporting pixels simultaneously and then by use of the latter feature we expect to have some evaluation scheme for recognizing whether P is occupied by any defect or not.

4.1 Identification pixel pair state

Based on the multiple paired pixel consistency model of defect-free logotype established above, MPPC can obtain the spatial information and the consistency information among different samples of target pixel P , and then compare the OC difference between the target pixel P and supporting pixel Q_n^P to judge the state of target pixel P , as shown in Fig 4.1. Once there is an outlier in the consistency relation of pixel pairs, we regard this situation as the abnormal state of pixel pairs, and the target pixel can be estimated as a defective pixel.

In this step, we should evaluate the N pixel pairs' state $(P, \{Q_n^P\}_{n=1,2,\dots,N})$ in turn. For each pixel pair (P, Q_n^P) , we design a binary function $\beta(Q_n^P)$ for identifying the normal/abnormal state which can be estimated by the following expression.

$$\beta(Q_n^P) = \begin{cases} 1 & \|\Delta(p, q_n) - \hat{\mu}_n\| \geq C \cdot \hat{\sigma}_n \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

where p and q_n are the OC of two elements in the n -th pair defined around the target pixel P , respectively. Here we use a binary normal distribution of differential increments of pixel pairs instead of the traditional single Gaussian pdf-based identification function of a single pixel. In a single Gaussian pdf-based method, the ideal threshold should be changed following the latest intensity variable. For instance, when the illumination fluctuations become intense, the standard deviation should become larger. In our proposed method, the stable differential increment of a pixel pair provides an adaptive observation so that $\check{\gamma}_n$ is only related to the noise acting on each pixel. So, we do not need an adjustable C to adapt to changes caused by illumination changes. And C is a constant parameter which can be set from 1.0 to 3.0 to define an area for acceptance of identify relation of 68% – 99% of its probability. In Eq. 4.1, when $\beta(Q_n^P)$ is 1 or 0,

we can assume the stochastic relation defined in the trained model in the previous step could be violated or hold. We can get the N bits of $\beta(Q_n^P)$ for each target pixel P .

4.2 Identification target pixel's state

After identifying the normal/abnormal state of the pixel pair, N bits of $\beta(Q_n^P)$ are produced for the following decisions of each P . To identify the defective/defect-free state of it, we defined a rate $\xi(P)$ of maintenance as follows.

$$\xi(P) = \frac{1}{N} \sum_{n=1}^N \beta(Q_n^P) \quad (4.2)$$

If $\xi > T$, where T is the threshold parameter introduced in advance. That can be adjusted to achieve desired results, the target pixel can be considered as defective pixel. Otherwise, P is considered as a defect-free pixel. For instance, if $T = 0.5$, and the number of abnormal Q_n^P is larger than $N/2$, namely, $T = \frac{1}{N} (\text{floor}(\frac{N}{2}) + 1)$, then P should be a defective pixel. The procedure for computing $\xi(P)$ for every target pixel is performed by a bits counting operation, along with a look-up table for calculate $\beta(Q_n^P)$. The pseudo-code of the defect detection is shown in Algorithm 2.

Algorithm 2: Defect detection

Input: Test image J , set thresholds C and T

Output: Defective/Defect-free

```

1 for Each logotype pixel  $P$  in  $J$  do
2   Initialization:
3   Load LUT  $[u'_i, v'_i, \hat{\sigma}_i, \hat{\mu}_i]$  of  $\{Q_i^P\}_{i=1,2,\dots,N}$ 
4   Pixel pair identification:
5   for  $i=1,2,\dots,N$  do
6     if  $|\Delta(p, q_i) - \hat{\mu}_i| > C * \hat{\sigma}_i$  then
7        $\beta(Q_i^P) = 1$ 
8     else
9        $\beta(Q_i^P) = 0$ 
10  Defect decision:
11  Compute the mismatched ratio.
12   $\xi(P) = \frac{1}{N} \sum_{i=1}^N \beta(Q_i^P)$ 
13  if  $\xi(P) > T, (0 < T < 1)$  then
14     $P \rightarrow \text{defective}$ 
15  else
16     $P \rightarrow \text{defect-free}$ 

```

4.3 Parameter discussion

In this section, we discuss the influence of some parameters on defect detection results. They are Γ which is a pre-specified threshold level for ignoring the pixels with low contrast neighborhood and N which is the number of supporting pixels.

4.3.1 Effect of the value Γ on experimental results

The threshold value Γ plays an important role in suppressing the effects of noise and has to be selected according to the problem at hand; too large values can cause the suppression of texture information. Mentioned in reference [75] they used a small value of 10 for Γ , which for most of their experience is beneficial.

Table. 4.1 shows the defect detection quantitative analysis when $\Gamma = 10$. Here we utilize three evaluation metrics, *Precision* (also known as positive predictive value), *Recall* (also known as sensitivity) and *F-measure* for quantitative analysis. *Precision* can be seen as a measure of exactness or fidelity, and *Recall* can be seen as a measure of completeness of defect. *F-measure* is a weighted harmonic mean of the *Precision* and *Recall*. In the study of defect detection, we believe that the integrity of the defective parts extracted is very important. That is, the demand for recall is higher. From Table. 4.1, we can see that the value of recall is not so high. So we want to raise the value of recall, and we think that 10 may not be the optimal value of Γ . In the following, we discuss the effect of the value Γ on experimental results and find the optimal value of Γ .

Table 4.1: Quantitative analysis of defect detection, when $\Gamma = 10$

Character	Recall	Precision	F-measure
H	0.67	0.55	0.61
U	0.72	0.54	0.61
A	0.70	0.55	0.61
W	0.72	0.53	0.61
E	0.67	0.54	0.60
I	0.69	0.56	0.62

In this fundamental experiment, the six sets images which consist of characters "H", "U", "A", "W", "E", "I" were used. Each set has 160 defect-free images that cover three different conditions in illumination corresponding to three groups of dark, normal and bright, respectively. Due to difficult to collect real defective product from factories, here we use artificial defect. We cut out 10*10 pixels from the background area. Then add this part to logotype part. This artificial defect is used to simulate the defect of missing printing.

The purpose of this experiment is to examine the defect detection performance of the proposed method using different value of Γ , and to find the optimal value of Γ . Here we set the value of Γ from 0 to 15 one by one for testing. Namely, $\Gamma = [0, 1, \dots, 15]$. The

Table 4.2: Quantitative analysis of defect detection, when the number of supporting pixels=10

Character	Recall	Precision	F-measure
H	0.71	0.55	0.63
U	0.74	0.54	0.63
A	0.73	0.54	0.62
W	0.74	0.53	0.62
E	0.70	0.54	0.61
I	0.71	0.56	0.63

quantitative analysis of each character is shown in Fig. 4.2. From (a) to (d), they are the result of characters "H", "U", "A", "W", "E", and "I". The blue line indicates the trend of *Recall*, gray line indicates the trend of *F-measure*, and orange line indicates the trend of *Precision*.

And from Fig. 4.2, we can observe that the parameter Γ mainly affects *Recall*. For instance, in Fig. 4.2(b) when the Γ smaller than 9, the *Recall* is stable at around 0.65. According to the definition of *Recall*, this result means the number of True Positive (TP) pixels are a little larger than the number of False Negative (FN) pixels which indicate that too small value of Γ cannot provide a robust detection. And when the value of Γ become larger, the *Recall* grows rapidly, but the *Precision* has a slight downward trend. And when the value of Γ bigger than 12, the *F-measure* become stable. And when the Γ bigger than 14, the *Recall* become stable. Therefore, we think $\Gamma = 12$ maybe is the optimal value for this time.

4.3.2 Effect of the number of supporting pixel N on experimental results

The number of supporting pixels N is important to establish a reliable model. A larger number of supporting pixels can provide more information to make subsequent defect extraction more efficient. But using too many numbers of supporting pixels cause two problems. One is that it can result in over-fitting of the extracted information, resulting in more noise in the experimental results; and another one is that it will cost more computational time.

Table. 4.2 shows the quantitative analysis of defect detection when using 10 supporting pixels. Here we also utilize previous introduced three metrics for quantitative analysis. They are *Precision*, *Recall* and *F-measure*, respectively. From the Table. 4.2, we can see that the value of *Recall* is not so high. So, we assume that the number of supporting pixels may influence the performance of defect detection. In the following, we discuss the effect of the number of supporting pixels N and find the optimal value of N .

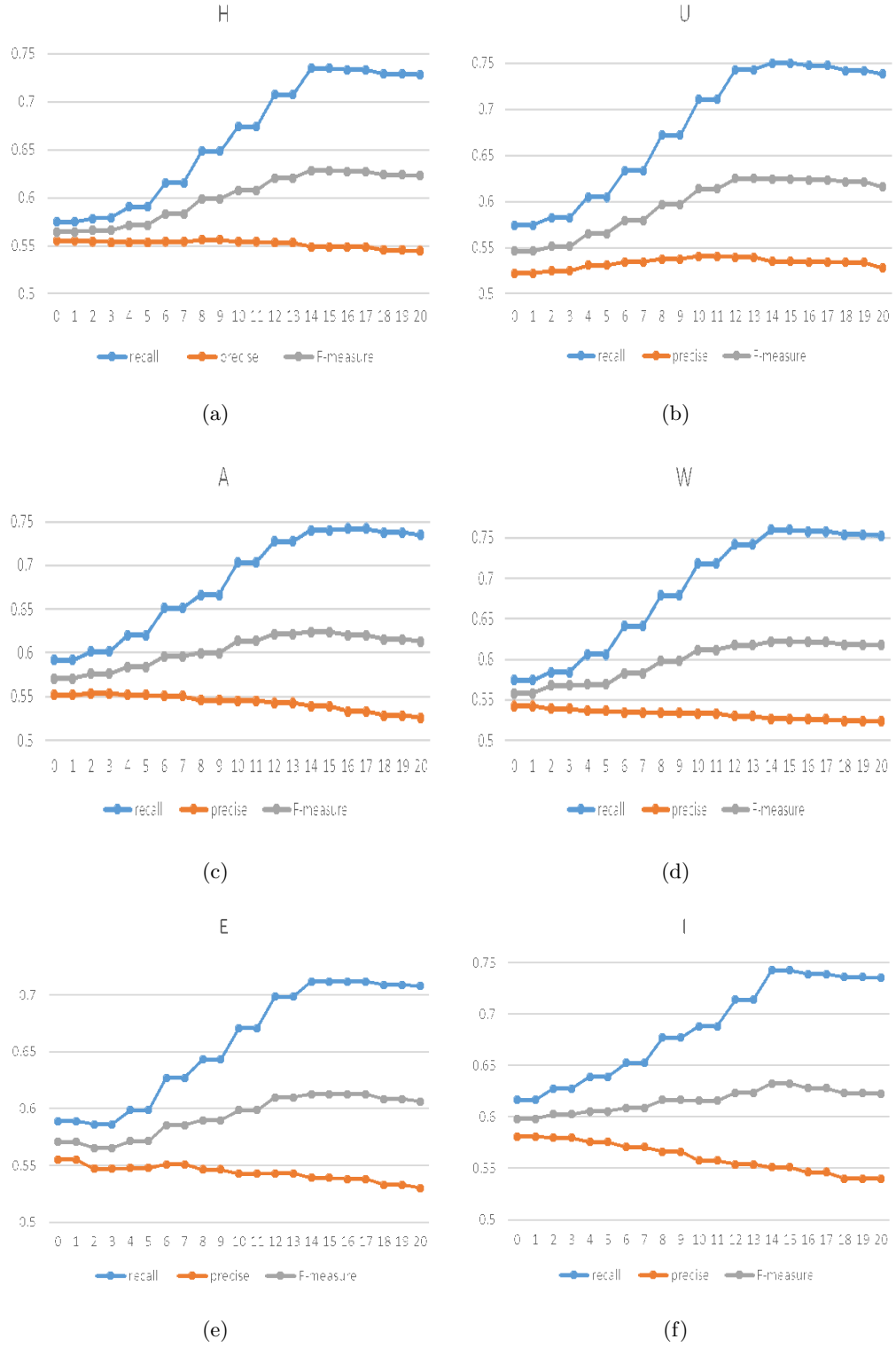


Figure 4.2: Quantitative analysis of defect detection for different characters

A basic experiment is designed to examine the defect detection performance using different number N , and to find the optimal value of them. Here we set the values in the range of $[2, 4, \dots, 20]$.

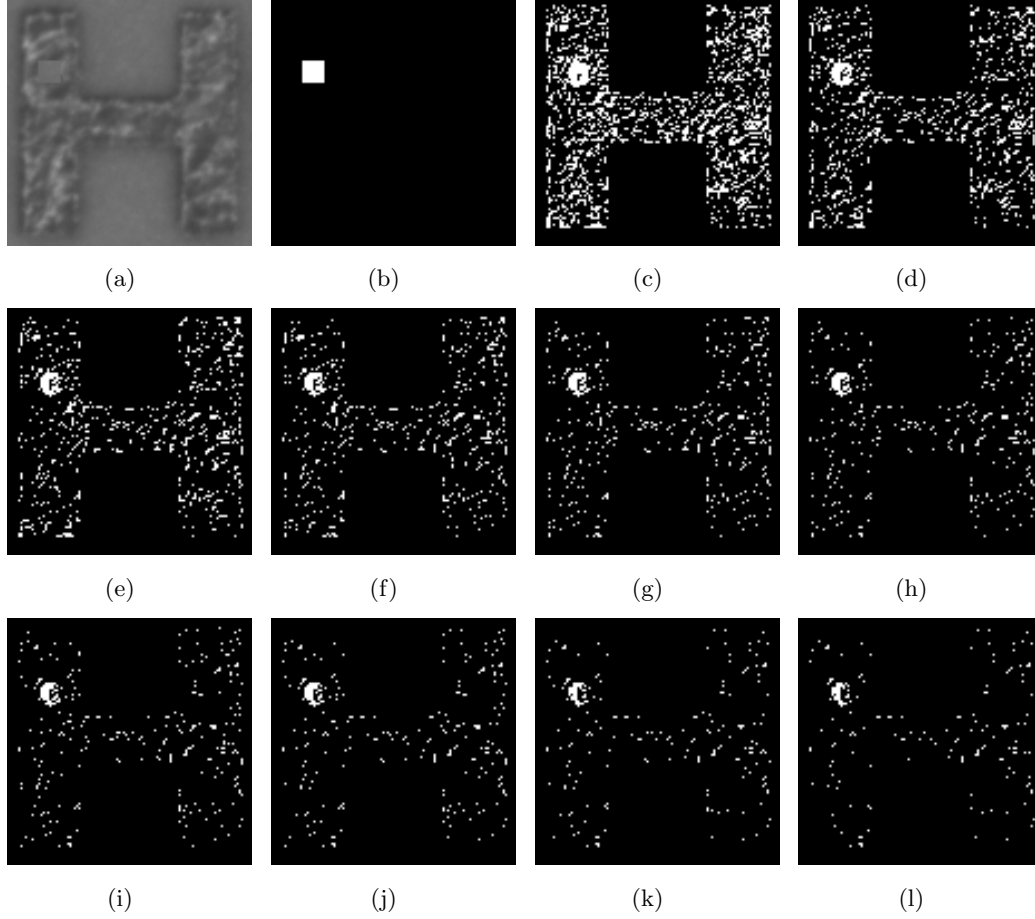


Figure 4.3: some examples of defect detection result by using different number of supporting pixel. (a) original image, (b) ground truth image. From (c) to (l), defect detection results under the number of supporting pixels from 2 to 20.

Fig. 4.3 shows some defect detection results. As can be seen from Fig. 4.3, as the number increases, the defect extraction integrity decreases somewhat and the noise level is also declining, and furthermore, when the number is too small, the extraction of the defect performance is rather high, while the noise such as a lot of false positive pixels have appeared. Maybe it is caused insufficient training models. When the number becomes larger, the noise level in the detection result is obviously decreased, but the extracted part of defect is not complete. This may be due to the training model is over-fitted.

And Fig. 4.4 shows the quantitative analysis of character "H". And Fig. 4.4 we can observe that the parameter N mainly affects Recall. As the number of supporting pixels increases, Recall drops slightly, and Precision increase slightly. When $N = 2$, Precision is around 0.5. According to the definition of Precision, the result means the number

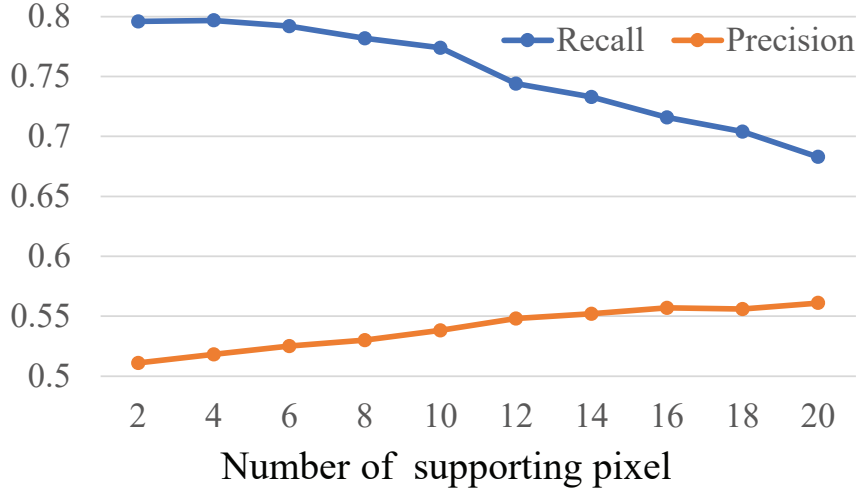


Figure 4.4: Quantitative analysis of defect detection

of false positive pixels (FP) and true positive pixels (TP) are similar which indicated too small number of supporting pixels cannot provide robust detection. The value of N is up to 10, Recall can maintain a relatively high level, and Precision gradually increases. After the value of 10, as N increases, Recall has a significant drop, and Precision is almost unchanged. So, we think 10 is the optimal value of N in this study.

4.4 Post-filtering

Fig. 4.5 shows an example of a detection result, where the white and black dots represent detected defective pixels and defect-free ones, respectively. (a) is the ground truth image, and (b) is the detection result. We used a synthesized image in which the real logotype ‘H’ was superimposed a background region of the same shape and size as the white square in the ground truth image to check the fundamental performance of the proposed detection algorithm here. One can find that not so small number of over-detections in the background and under-detection in the defect region. Over-detection means that the pixels are of pseudo-defective characteristics but defect-free in reality, while under-detection pixel means the opposite case. Observing the detection result, we can also see the high density of white dots in the true defective region and the sparseness of white dots in the defect-free area, which indicate the effectiveness of the algorithm proposed in the previous chapter.

Fig. 4.6 outlines the definitions of true positive (TP), false positive (FP), true negative (TN) and false negative (FN) in the defect detection.

Here, we introduce the idea of connected component labeling [80], where any connected region that has the size less than a threshold T_h will be removed as FPs, and

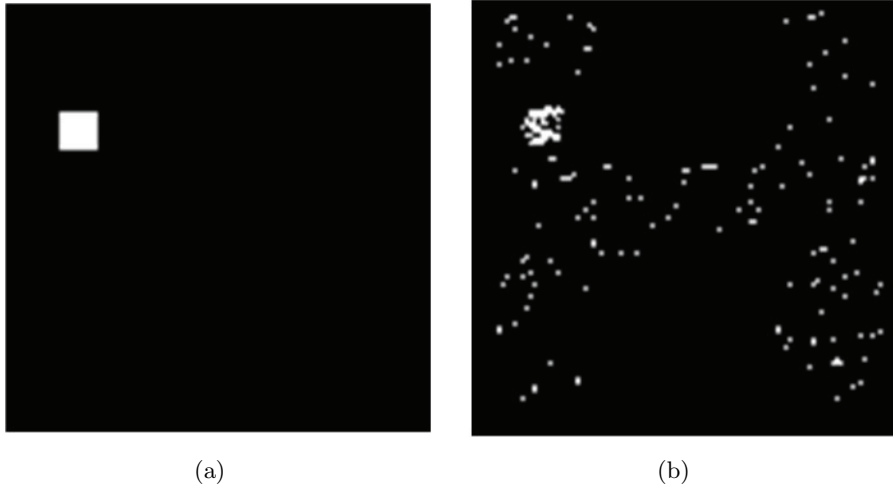


Figure 4.5: Example of detection result

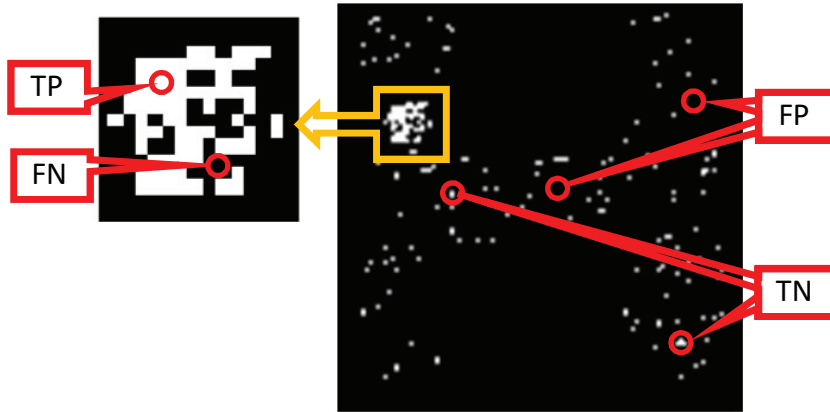


Figure 4.6: Definitions of true positive (TP), false positive (FP), true negative (TN) and false negative (FN) in defect detection

T_h is a threshold that can be adjusted to achieve the desired result. We utilized 100 experimental results from MPPC to check the size of the connected components in the defect-free areas. We observed that over 99% of the population of the connected components of a size of less than 5 pixels. Based on this, we estimated a reasonable value of T_h , which is likely to be 5 in this study. For FN pixels, however, we utilize the closing morphological operator [81], which is defined as

$$Y \cdot M = (Y \oplus M) \ominus M \quad (4.3)$$

where Y is any binary image from MPPC; $\oplus$ and $\ominus$ are dilation and erosion operator, respectively, and M is a mask matrix of the size 5×5 . This size is determined experimentally and meets our requirements, where we used the 3×3 , 5×5 , 7×7 to filter 100 experimental results. The average values of recall and precision for 3×3 , are 0.74 and 0.96; for 5×5 , are 0.86 and 0.95; for 7×7 , are 0.90 and 0.82. The best result is

5×5 here. From these experiences, we found that the smaller filter keeps many holes unchanged, while the larger sizes can easily lead to excessive filtering for small defects. Moreover, all the images used in this study have a size of approximately 100×100 pixels and a resolution of 15 pixels/mm. The physical size of each character is approximately 7mm. Usually, a defect is less than 10 pixels in length (or less than 1mm) and is usually a hard target for human inspectors to perform visual inspection with their naked eyes.

4.5 Summary

In this chapter, we introduced how to utilize the proposed MPPC model of the relationship between pixel pairs in a defect-free logotype for detecting many types of logotype defects.

- Firstly, we introduce the defect detection based on the MPPC model. This stage can be divided to two main steps: (a) identify the state of pixel pair, whether the pixel pair matches its MPPC model; (b) determine the state of pixel, defect-free or defective.
- And then we discussed the influence of some parameters on defect detection results and found the optimal parameters through experiments.
- Finally, we present a novel filtering method to improve performance.

Chapter 5. Position-dependent Data Inhibition

In this chapter, we will introduce a modification method for improve the performance of defect detection. First, we introduce the problem in the detection results of the MPPC algorithm, that is, the defects will appear in the detection results. Then we introduced why this happens. Finally, we propose an improved algorithm for this situation. The improved algorithm mainly consists of two parts. The first part is to use the chi-square test to determine whether this situation exists or not, and then if it exists, we use skewness to detect the location of the noise and remove it.

Fig 5.1 shows an example of the detection of an artificial defect involving a square

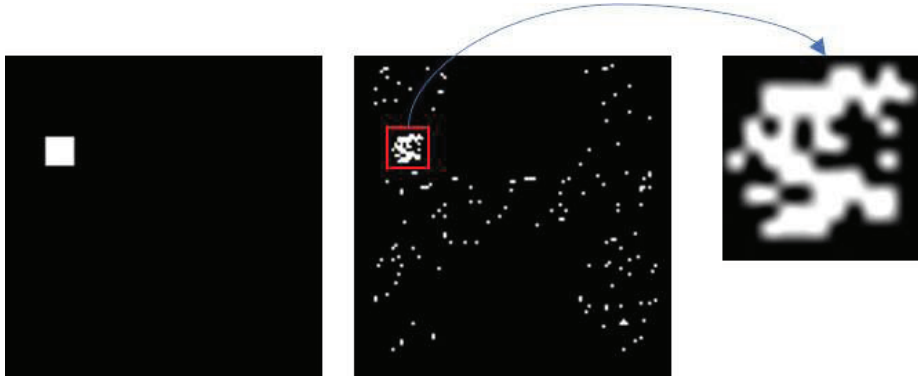


Figure 5.1: Example of a detection result

shape embossed in a real image. From left to right, they are ground truth, detected defect, and magnified version. The proposed method can detect many pixels in the defective area, but it sometimes fails to discriminate defective pixels from non-defective ones. We wish to improve the integrity of defect detection, that is, to fill the holes in the detected defects, because any defect must be generally aggregated in connected regions. After analyzing this phenomenon, we found that two factors may cause this phenomenon. One is that the parameters of our settings may not be optimal. Another possibility is that there may be noise data in our training data, so we cannot establish a stable and effective MPPC model.

With this in mind, we have tried to select suitable parameters in the original MPPC model; however, there is some difficulty in adjusting parameters C and T in Eq. ???. So an example is provided to better understand their direct and important influence on

detection. First, C was adjusted, as shown in Fig.5.2. shows the contribution of C on

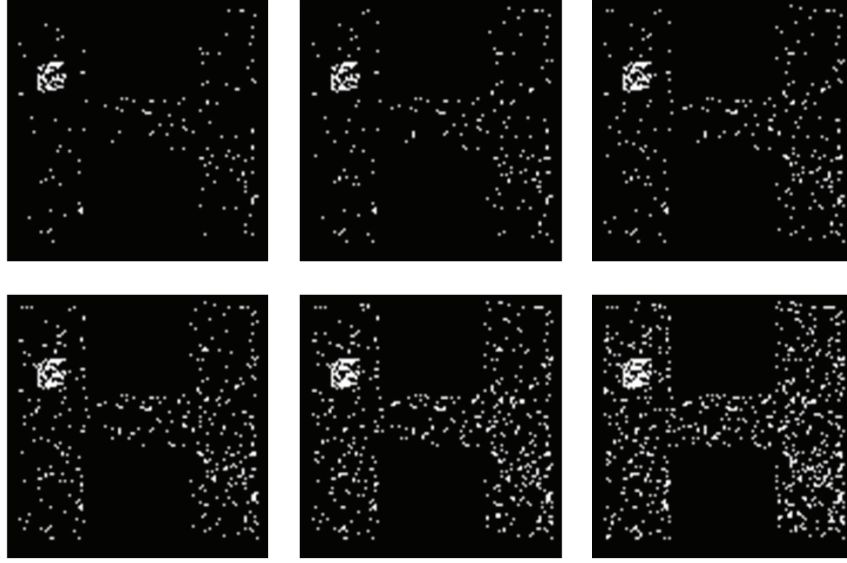


Figure 5.2: Contribution of C on defect detection

defect detection. From top left to bottom right, they are the result, when $C=2.0, 1.8, 1.6, 1.4, 1.2$, and 1.0 . The figure indicates a trade-off between the decrease of holes in the defective part and the increase of noise along its periphery. This observation indicates that it is difficult to select any general value of C that can provide acceptable results, even for given image data.

Therefore, we propose a general approach to the problem by introducing data (or resource) management that has been utilized in the production management field [82]. As a typical example, a certain analogy exists when testing a set of many parts are delivered by suppliers that may include occasional faulty parts. In general, just before the assembly process, managers should accept qualified parts or reject defective ones using inspection and statistical results. We apply this to testing or selecting ill-conditioned data images or samples using two statistical tests.

As a different approach to modify a model such as MPPC, Zhou [83] proposed a replacement algorithm to modify the positional structure of the pixel pair, but it was essentially tailored to their technique. In this study, we propose a more general procedure, named position-dependent data inhibition (PDI), to improve performance in defect detection. To design an effective and economical scheme, an approach based on defining bad samples not from data images rather for each pair of pixels $\{P, Q\}$ is presented.

Detecting defects in the materials examined in this study is not an easy task, even for professional inspectors who are trained to find small and/or obscure defects; furthermore, judgment may involve variations depending on the individual inspector. Another problem is that the number of training samples is not significantly large in general. If

this trial set of samples contains inaccurate or noisy data, they may have a non-negligible effect on constructing a useful model for defect detection.

Fig 5.3 presents an example which has some outlying values in the left zone and does

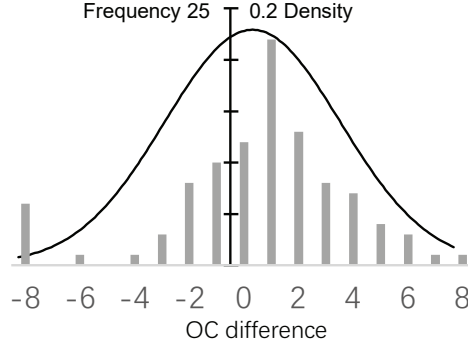


Figure 5.3: Example of OC difference in a randomly selected pixel pair

not achieve good fitness to the single Gaussian distribution. To realize the proposed inhibition process, two schemas are required: a judgment, and a localization for inappropriate or outlying data. If the fit is poor, the data contain some noise. In this case, we need to proceed with the second step, which comprises filtering inconsistent image data based on inhibition index list (IIL). From the resulting filtered image, a single elementary Gaussian model can be created to fit the target and supporting pixel pair again.

5.1 χ^2 test for appropriateness

We first introduce how to evaluate the appropriateness of the trained model, in other words, how to judge the fitness quality of the single Gaussian probability function to the OC difference histogram calculated for each pixel pair. In elementary modeling, two histograms are created: the OC difference histogram, H_O , and then, the expected histogram, H_E , is calculated from the fitted probability function. Here, we introduce Pearson's χ^2 test [84] to measure the goodness of fit which is defined as

$$\chi^2 = \sum_{n=1}^N \frac{(O_n - E_n)^2}{E_n} \quad (5.1)$$

where N , O_n , and E_n are the number of bins, an observed value in the n -th bin of H_O , and the corresponding expected value in H_E , respectively. The definition of E_n is

$$E_n = \left\{ \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(n-\mu)^2}{2\sigma^2}} \right\}_{n=1,2,\dots,N} \quad (5.2)$$

where the mean, μ , and standard deviation, σ , are calculated from the corresponding pixel pair, P and Q . A reasonable value for p value can be determined from the χ^2 table

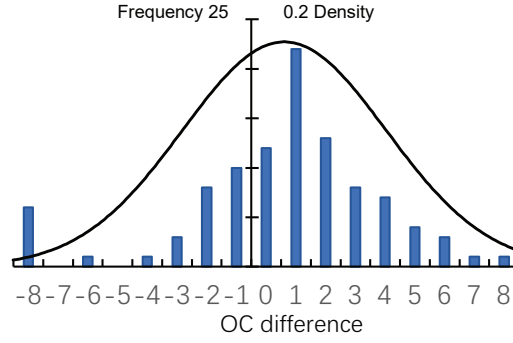
as shown in Fig. 5.4. For instance, in this study, there are 17 bins in our histogram, so the degree of freedom is $16(= 17 - 1)$. Then we set the value of $\aleph$ to 0.05, and we can get the $p = 26.30$ by looking up the table.

Degrees of Freedom	Probability of a larger value of χ^2								
	0.99	0.95	0.90	0.75	0.50	0.25	0.10	0.05	0.01
1	0.000	0.004	0.016	0.102	0.455	1.32	2.71	3.84	6.63
2	0.020	0.103	0.211	0.575	1.386	2.77	4.61	5.99	9.21
3	0.115	0.352	0.584	1.212	2.366	4.11	6.25	7.81	11.34
4	0.297	0.711	1.064	1.923	3.357	5.39	7.78	9.49	13.28
5	0.554	1.145	1.610	2.675	4.351	6.63	9.24	11.07	15.09
6	0.872	1.635	2.204	3.455	5.348	7.84	10.64	12.59	16.81
7	1.239	2.167	2.833	4.255	6.346	9.04	12.02	14.07	18.48
8	1.647	2.733	3.490	5.071	7.344	10.22	13.36	15.51	20.09
9	2.088	3.325	4.168	5.899	8.343	11.39	14.68	16.92	21.67
10	2.558	3.940	4.865	6.737	9.342	12.55	15.99	18.31	23.21
11	3.053	4.575	5.578	7.584	10.341	13.70	17.28	19.68	24.72
12	3.571	5.226	6.304	8.438	11.340	14.85	18.55	21.03	26.22
13	4.107	5.892	7.042	9.299	12.340	15.98	19.81	22.36	27.69
14	4.660	6.571	7.790	10.165	13.339	17.12	21.06	23.68	29.14
15	5.229	7.261	8.547	11.037	14.339	18.25	22.31	25.00	30.58
16	5.812	7.962	9.312	11.912	15.338	19.37	23.54	26.30	32.00
17	6.408	8.672	10.085	12.792	16.338	20.49	24.77	27.59	33.41
18	7.015	9.390	10.865	13.675	17.338	21.60	25.99	28.87	34.80
19	7.633	10.117	11.651	14.562	18.338	22.72	27.20	30.14	36.19
20	8.260	10.851	12.443	15.452	19.337	23.83	28.41	31.41	37.57
22	9.542	12.338	14.041	17.240	21.337	26.04	30.81	33.92	40.29
24	10.856	13.848	15.659	19.037	23.337	28.24	33.20	36.42	42.98
26	12.198	15.379	17.292	20.843	25.336	30.43	35.56	38.89	45.64
28	13.565	16.928	18.939	22.657	27.336	32.62	37.92	41.34	48.28
30	14.953	18.493	20.599	24.478	29.336	34.80	40.26	43.77	50.89
40	22.164	26.509	29.051	33.660	39.335	45.62	51.80	55.76	63.69
50	27.707	34.764	37.689	42.942	49.335	56.33	63.17	67.50	76.15
60	37.485	43.188	46.459	52.294	59.335	66.98	74.40	79.08	88.38

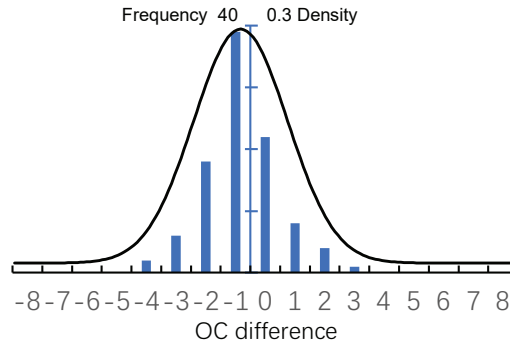
Figure 5.4: The critical value p of the χ^2 distribution

In the evaluation step, if $\chi^2 < p$, the fit is good so data inhibition is not required, but if $\chi^2 \geq p$, the fit is considered poor, and then, the pixel pair of interest must be supported by some ill-conditioned data, i.e., outlying data, which have a significantly larger OC difference. This assessment is required to certify the test's validity with respect to real-world data images.

Fig. 5.5 presents two typical examples of fitness checks involving real data images, which indicate that the algorithm can be applied to our data. Any outlying data should be excluded using the procedure described in the next section until $\chi^2 < p$ is achieved.



(a)



(b)

Figure 5.5: Two examples of fitness checks. (a) and (b) the case of with outliers ($\chi^2 = 105.3$) and without outliers ($\chi^2 = 10.2$)

5.2 Skewness test for outlier detection

The next problem is how to find the defective data or position of noisy data. In MPPC modeling, as any statistical characteristics in each pixel pair can be represented by a single Gaussian distribution, which is symmetrical in nature, it is necessary to utilize a statistical procedure that is sensitive to some unbiased data in the histograms. The *skewness* [85] is a statistical measure used to evaluate unbiasedness around the mean in histograms. We utilize the normalized moment coefficient of skewness, γ_1 , defined as:

$$\gamma_1 = E \left[\left(\frac{X - \mu}{\sigma} \right)^3 \right] \quad (5.3)$$

where X , μ , and σ represent the observed frequencies, and their mean and standard deviation, respectively, and E is the expectation operator. Consider the two distributions in the Fig. 5.6. In each diagram, the values on the right side of the distribution gradually differ from the values on the left. These tapering sides, called tails, provide a visual

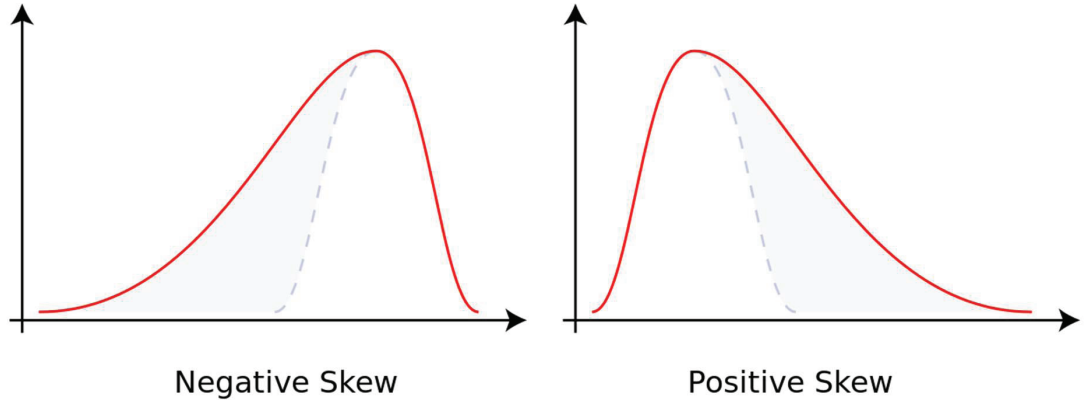


Figure 5.6: The two distributions of skewness

means of determining which of the two skewness of a distribution:

- Negative skewness: long left tail; The mass of the distribution is concentrated on the right side of the graph. Although the curve itself appears to be tilted to the right or to the right, the distribution is considered to be tilted to the left. To the left is to draw the left tail, usually to the left of a typical data center. A left-skew distribution is usually shown as a right-skew curve
- Positive skewness: long right tail; The mass of the distribution is concentrated on the left side of the graph. Although the curve itself appears skewed or tilted to the left, the distribution is said to be skewed to the right. In contrast, "right" means to draw the right tail, and usually to the right of the typical data center. A right-skew distribution is usually shown as a left-skew curve

According to the value of γ_1 , if $\gamma_1 < 0$ any outlying data will be located to the left or in a smaller zone from the mean or center, while otherwise outliers can be found to the right or in a larger zone. We can apply this procedure along with registering the data or image ID into the IIL and exclude the corresponding data from fitting until the remaining OC values in the difference histogram satisfy the fitness condition specified by the χ^2 value. Fig. 5.7 demonstrates the effect of the outlier removal process when applying the two procedures.

5.3 Inhibition index list

A list-based structure for the representation of the IIL is developed and we describe the modification algorithm that uses the IIL. Let $K = \{1, 2, \dots, K\}$ be the set or list of all possible indexes for the data images, which is used initially to calculate the mean and standard deviation value for any pixel pairs. Repeated application of the two steps involving judgment and localization creates an IIL that includes the indexes of inhibited data for every pixel pair.

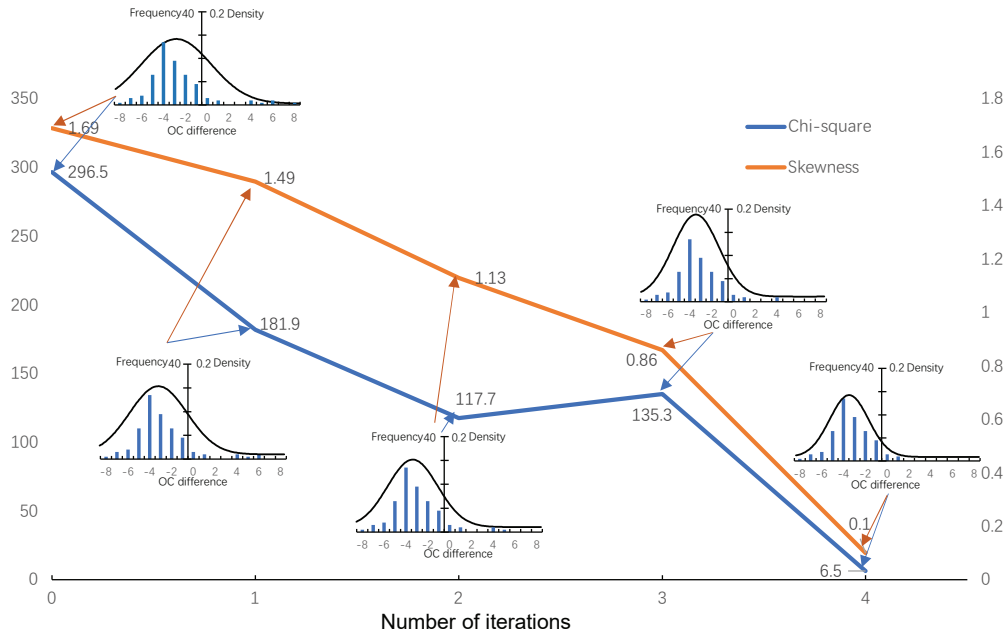


Figure 5.7: Effect of applying the outlier removal process

Fig. 5.8 shows the detailed structure of IIL in contrast with the total index list K . In Fig. 5.8 (a), three P pixels, $P_{1,1}$, $P_{23,40}$, $P_{43,40}$ are present all over a logotype area, and their sets of supporting pixels Q , where their pairing is represented by the subscripts P and superscripts of Q . Supporting pixels disperse unlimitedly, and sometimes they are shared by different P such as $Q_3^{1,1}$ and $Q_1^{43,40}$. Fig. 5.8 (b) presents a list named the inhibited index list (IIL), in which each row corresponding to a pixel pair contains the indices of inhibited images detected by use of the above-mentioned algorithm. For example, the pair consisting of $P_{23,40}$ and $Q_1^{23,40}$ has more than two inhibited images including 8 and 17, and so on. Note that $Q_3^{1,1}$ and $Q_1^{43,40}$ represent the same pixel, although each pair of inhibited images is different in general.

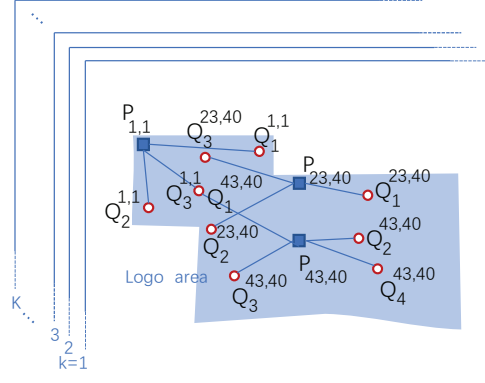
During every step of the modification, the mean and the variation of OC differences adopted for Gaussian fitting should be updated as

$$\mu = \frac{1}{|K \setminus \text{IIL}|} \sum_{K \setminus \text{IIL}} \Delta(P, Q) \quad (5.4)$$

$$\sigma^2 = \frac{1}{|K \setminus \text{IIL}|} \sum_{K \setminus \text{IIL}} \{(\Delta(P, Q) - \mu)^2\} \quad (5.5)$$

5.4 Verification of PDI's performance

In this section, we evaluate the performance of defect detection by MPPC with PDI. Here, the three sets of images, which consist of 25 images for character "A", 20 for



(a)

P	Q	1234567...	...	...K
1, 1	1	3	10	27 ...
	2	5	9 ...	
	3	2	13	29 ...
	...			
23,40	1		8	17 ...
	2			32 ...
	3	6	14 ...	
	...			
43,40	1	4		19 ...
	2		7	17 ...
	3		9	25 ...
	4			21 ...
	...			

(b)

Figure 5.8: Detailed structure of IIL. (a) three examples and (b) the corresponding inhibition index list.

“E”, and 15 for “U” were used, which contained real defects. We used four cases of real defects in different positions for “A,” four for “E,” and three for “U.” Fig. 5.9 shows some examples of defect detection using the proposed method, where we have separated defects from the background. From top to bottom they are the test images, their ground truth images, detected results by MPPC, and detected results by MPPC with PDI, respectively. Actual handling of background defects was not considered in this paper because the focus is to design an effective technique for defect detection in the logotype regions. The examples indicate that the proposed method can locate most parts of an area containing defects using only MPPC, and that after introducing the PDI, the defective area could be detected more completely.

In this paper, we focused on the logotype area and have excluded the background area (unprinted area). However, in Fig. 5.9, the first, second, third, and last column

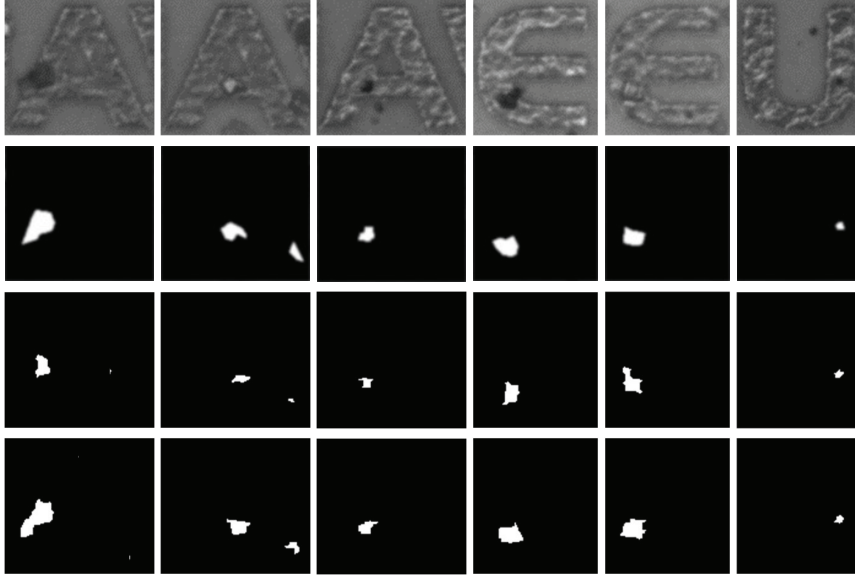


Figure 5.9: Some examples of defect detection results

contain some defects in the background area. To solve this issue, we will extend our algorithm to perform defect detection in the background area in the future.

Table 5.1: Pixel-based performance for different characters

Character	Method	Recall	Precision	F-measure
A	MPPC	0.74	0.96	0.84
	MPPC+PDI	0.89	0.95	0.92
E	MPPC	0.87	0.89	0.88
	MPPC+PDI	0.92	0.87	0.89
U	MPPC	0.80	0.92	0.86
	MPPC+PDI	0.88	0.90	0.89

Table 5.1 shows the pixel-based performance evaluation. From this table, all F-measure values increased in three characters, and furthermore a substantial increase occurred in all of the recall values because the shapes of the defects could appear multiple times in the detected images, resulting in the improvement of their integrity or reproducibility.

5.5 Summary

Chapter 5 proposed the modification scheme for MPPC, called Position-dependent Data Inhibition (PDI). It contains three main parts: (1) detection the modeling failure case; (2) detection noisy data. (3) inhibit the noisy data and remodeling.

- Firstly, we analyzed the missing detection problem of defect results by MPPC. We found that the missed detection was caused by a problem in the modeling process.
- And then we introduced the χ^2 test to check the modeling quality. If the value of χ^2 less than threshold, we judge this modeling process is failed, we need other processes to deal with this problem.
- If the modeling process is failed, we use skewness to detect the location of the noise. And then generate the inhibition index list for inhibiting noisy data and update the MPPC model.
- Finally, we verified the performance of PDI with reliable experiments.

Chapter 6. Experimental Evaluation

In this chapter, we would like to introduce the evaluation of experiments for our methods including five main contents: (2) Experimental setup; (1) Analysis measurements; (3) Experimental results and (4) Computational cost. In the end, the discussion will be presented to have a summary of the experimental results for our methods MPPC and MPPC+PDI.

6.1 Experiment setup

Fig. 6.1 shows a part of the real defect images for the experiments here, which were collected in the factories. All defect images have the corresponding ground truth images,

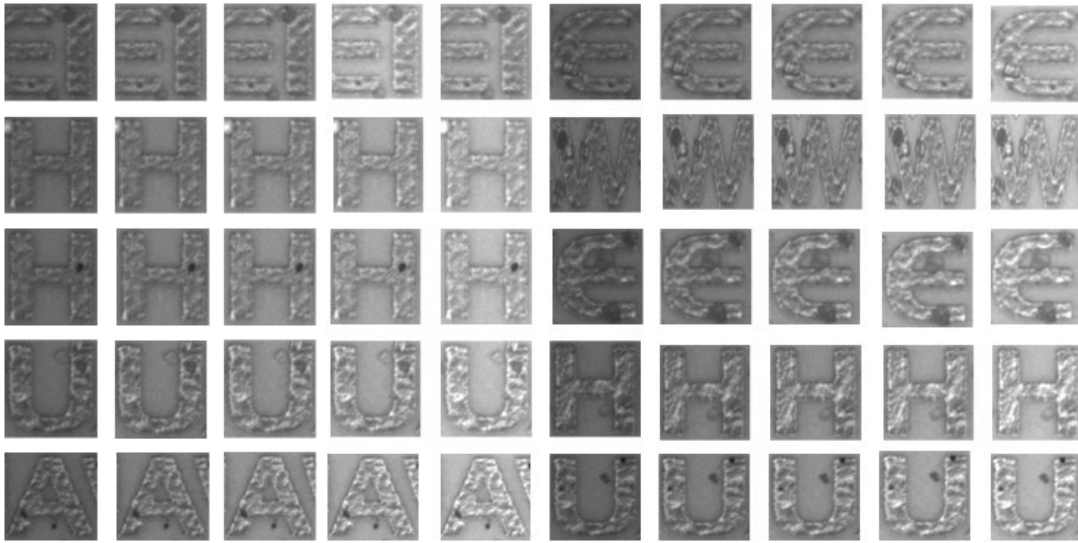


Figure 6.1: Some examples of real defect images

and the defect-free area is represented as black and the defective area as white. We utilize the six sets of real images from the factory, which consist of the six characters ‘H’, ‘U’, ‘A’, ‘W’, ‘E’ and ‘I’, each of which include 160 defect-free images that cover five different illumination conditions: darker, dark, normal, bright and brighter. We chose 60 defect-free images were taken under the five different illuminations to train the MPPC model. Through experiments, we set the two thresholds of the detection stage as $C = 2.0$ and $T = 0.5$, respectively.

As experimental images, we utilized two kinds of defects: real and synthesized or artificial defects shown in Fig. 6.2(a) and Fig. 6.2(b), respectively. Fig. 6.2 (a) shows small dust attached to the logotype ‘H’, which we utilized as a representative one to do the defect detection experiments under different illumination conditions.

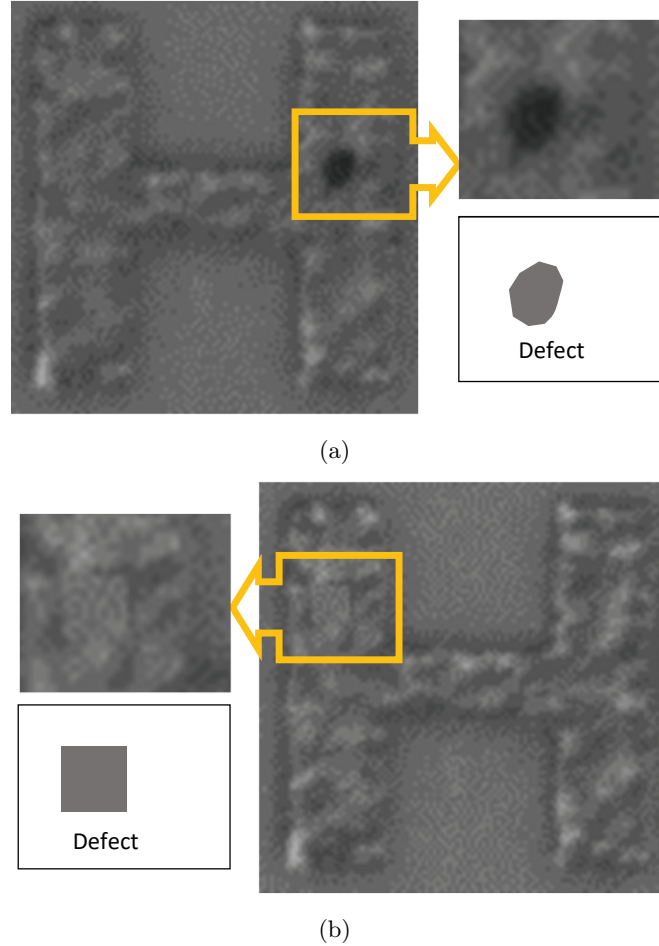


Figure 6.2: Two types of experimental images. (a) real defect and (b) synthesized defect.

Due to somewhat difficulty of collecting real defects from factories, we utilized some synthesized defects in the experiments as shown in Fig. 6.2 (b), where we first extracted a small square area from the background (unprinted portion) and pasted it into the logotype (printed portion). This artificial defect is representative of mis-printing of the logotype. As shown in Fig. 6.2(a), one can understand to define any ground truth for the real defects because of the uncertainty of the boundary even through magnification. Therefore, we use the synthesized defects in order to evaluate the performance of the proposed MPPC-based detection quantitatively too. The image data used in all the experiments in this paper can be downloaded from <https://hce-lab.net/cloud/index.php/s/3FFFFmU8JIL1RTO>.

Basically, since the proposed method does not consider the translation and rotation

changes in target images, any original data image should be aligned in the same position and should not have any individual rotation. In any real factory, it is not so difficult to realize this requirement because of their quality control. However, in this paper, we realized it through three inspection procedures. The first one is the inspection by some operators in the factory where all the data were corrected by using a specialized fixation tool to give them a constant position and orientation relative to the camera. Next, we checked their rotational constancy. Finally, all the elemental logotype images corresponding to each character were corrected in their position through the registration by the normalized cross correlation.

All the experiments in this paper were done by use of MATLAB 2016a implemented on a PC with Intel(R) Xeon(R) E5-1620 CPU with 64 GB RAM and 3.5GHZ clock.

6.2 Analysis measurement

There are several ways to evaluate the performance of defect detection. First pixel-level precision, recall, and F-measure are tried to test the proposed MPPC models and the detection algorithm for the statistical test based on the models, where our problem assumed as a two-class or binary classification problem to classify any pixel into the defective class and the defect-free class. Along with our problem to detect defects in pixel, we utilized three evaluation metrics: *Precision* (also known as positive predictive value), *Recall* (also known as sensitivity) and *F-measure*. These are widely applied in pattern recognition, information retrieval and binary classification. And pixel-level defect detection is a typical binary classification problem, so these three indicators can also be used for the quantitative analysis of defect detection. *Precision* can be considered a measure of accuracy, while *Recall* can be considered a measure of defect integrity. Here, $Prec(i)$ and $Rec(i)$ for test image i are define as

$$Prec(i) = \frac{TP(i)}{TP(i) + FP(i)}, \quad (6.1)$$

$$Rec(i) = \frac{TP(i)}{TP(i) + FN(i)}, \quad (6.2)$$

where

- $TP(i)$ is the number of true positive pixels in image i ;
- $FP(i)$ is the number of false positive pixels in image i ;
- $TN(i)$ is the number of true negative pixels in image i ;
- $FN(i)$ is the number of false negative pixels in image i .

True positive, *false positive*, *true negative* and *false negative* are four possible outcomes in an instance. Sine our target is defect detection, we define defective pixels as positive pixels, and defect-free pixels as negative pixels. If the pixel is defective and it

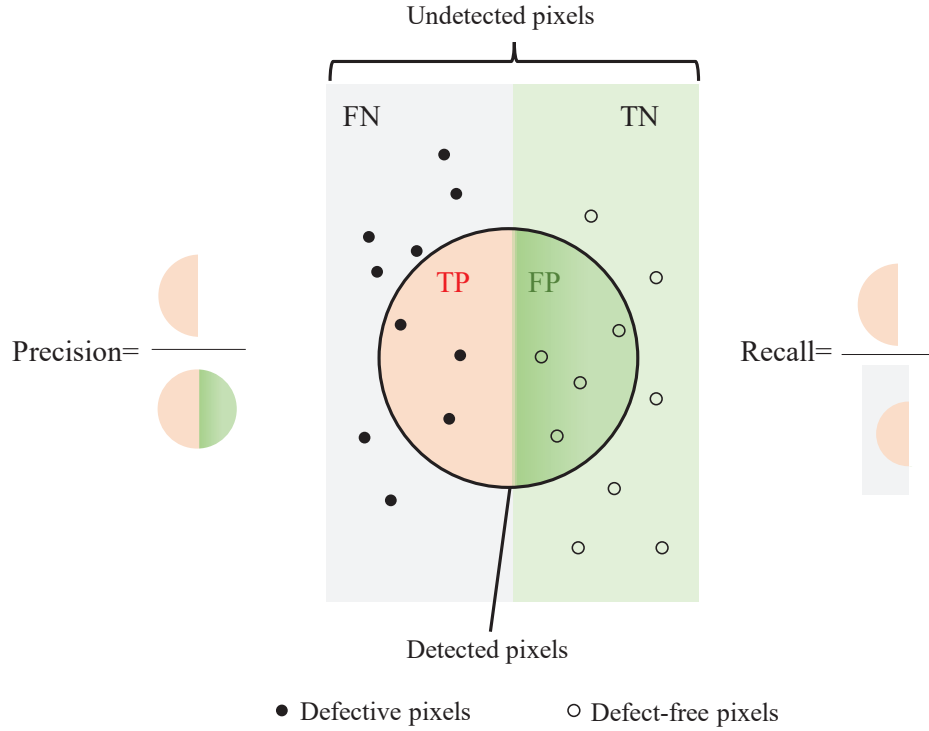


Figure 6.3: Abridged general view of the analysis measurements

will be classified as defective, it is counted as a true positive; if it is classified as defect-free, it is counted as a false negative. If the pixel is defect-free and it will be classified as defect-free, it is counted as a true negative; if it is classified as defective, it is counted as a false positive. Fig. 6.3 shows an abridged general view of the analysis measurements: *Precision* and *Recall*. Through Fig. 6.3, *Precision* can be seen as a measure of exactness for defect detection and *Recall* can be seen as a measure of completeness of detection. The F-measure indicator comprehensively considers recall and precision. Larger F-measure indicates higher performance.

The *Precision* is defined as:

$$Precision = \frac{1}{I} \sum_{i=1}^I Prec(i). \quad (6.3)$$

The *Recall* is defined as:

$$Recall = \frac{1}{I} \sum_{i=1}^I Rec(i). \quad (6.4)$$

Accuracy, *Precision*, and *Recall* are calculated as average value for the test frames. The *F-measure* is defined as

$$F-measure = \frac{2Precision \cdot Recall}{Precision + Recall}. \quad (6.5)$$

F -measure is calculated as an average value for the test frames, which is a computing score of *Precision* and *Recall* as a weighted harmonic mean value; I is the total number of test images.

For image-based performance evaluation [6], we utilized the detection rate and the false alarm rate as follows:

$$\text{Detection Rate} = \frac{N_T}{N_{TD}} \quad (6.6)$$

$$\text{False Alarm Rate} = \frac{N_F}{N_{TF}} \quad (6.7)$$

where N_T , N_F , N_{TD} and N_{TF} are the numbers of defective logotypes or images correctly detected, defect-free detected as defective, the total number of defective samples, and defect-free logotypes, respectively.

6.3 Experiments result

We believe that the following factors may have impact on defect detection result.

- Type of defect
- Size of defect
- Shape of defect
- Illumination condition
- Noise intensity of image
- Different printed characters

In this study, we deal with the type of defect that occurs when there is some dust attached to the logotype area. This type of defect is the most common and is difficult to detect. We consider two types in the experiment: fine dust and a larger foreign body. Because our method is pixel-based, the shape of the defect has little effect on the test results. Next, we will describe the defect detection performance of our method for different defect sizes, different illumination conditions, different noise intensities, and different printed characters.

6.3.1 Different defect sizes

We want to test how well small defects can be detected by our algorithm. Here, we use four types of image whose defect sizes are different. The size of the defect is of 10×10 , 7×7 , 5×5 , or 3×3 pixels. Because it is difficult to collect small defects. Here, we utilized 100 synthesized defect images for each defect size to do experiments. Fig. 6.4 shows some examples of experimental results for different size defects. From the top row

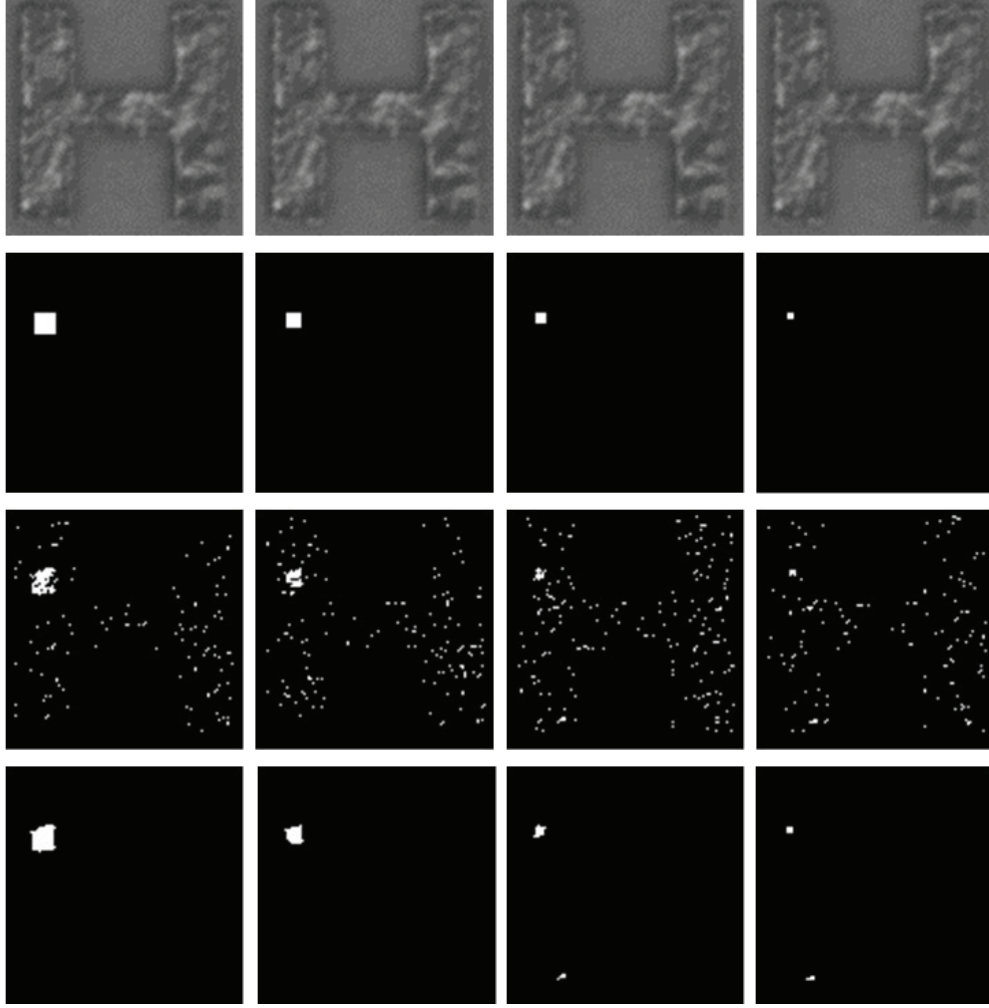


Figure 6.4: Some examples of defect detection with different defect sizes

shows the test images, their ground truth images, the detected results, and the filtered results. From the leftmost column, the defect size of 10×10 , 7×7 , 5×5 , and 3×3 pixels.

Table 6.1 shows the performance with respect to the size change. For almost of the results, the filtering could improve the performance substantially. Table 6.2 represents the image-based performance evaluation with 400 synthesized images including the four sizes of defects. We can see that when the defects size greater than or equal to 5×5 pixels, the detection rate can be maintained at a high level (about 95% on average). However, 3×3 pixels defect are slightly too small, and our algorithm is likely to miss detection. Therefore, we believe that Our algorithm can detect defects greater than or equal to 5×5 pixels, and ensure a high detection rate, while we may need more magnified observation if we want to detect much smaller defects.

Table 6.1: Pixel-level based performance evaluation of defect detection under different size of defect

Size of defect	Filter	Recall	Precision	F-measure
10×10	OFF	0.72	0.57	0.63
	ON	0.89	0.94	0.91
7×7	OFF	0.73	0.54	0.62
	ON	0.87	0.94	0.90
5×5	OFF	0.68	0.51	0.59
	ON	0.85	0.85	0.85
3×3	OFF	0.64	0.51	0.58
	ON	0.86	0.72	0.79

Table 6.2: Image based performance evaluation for different defect sizes

Size of defect	10*10	7*7	5*5	3*3
Detection Rate(%)	97	96	94	61

6.3.2 Different illumination conditions

Realizing precise control of the actual factory illumination conditions is difficult. Illumination conditions are mainly affected by two factors: one is the angle of the light source, and the other is the intensity of the illumination. The angle of the light source in the general factory is fixed at a well-adjusted angle. Therefore, we consider the second factor, i.e. the intensity of the illumination. It is common in factories to have multiple sets or stages of visual inspection, whose condition may change over time. The following factors can cause variations in the illumination intensity.

- Illumination conditions are affected by natural light.
- Life of light sources is limited. When we replace them, we cannot guarantee that the conditions will remain as they were before the change.
- We cannot guarantee that the illumination conditions of each production line are the same.
- In actual lighting, alternating current is used, which causes the lighting to fluctuate. As production lines run very fast, the shutter speed of the camera used for detection is very high. This results in a noticeable change in the illumination of the captured photo.

Therefore, the proposed algorithm should be robust to illumination fluctuations, for which we have introduced the orientation codes to build the MPPC model. Here, we utilize images taken under five different illumination conditions shown in Fig. 6.5. There are 50 real defect images and 100 defect-free images for each character. Fig. 6.5 shows some detection results. From the first row shows the test images, their ground truth

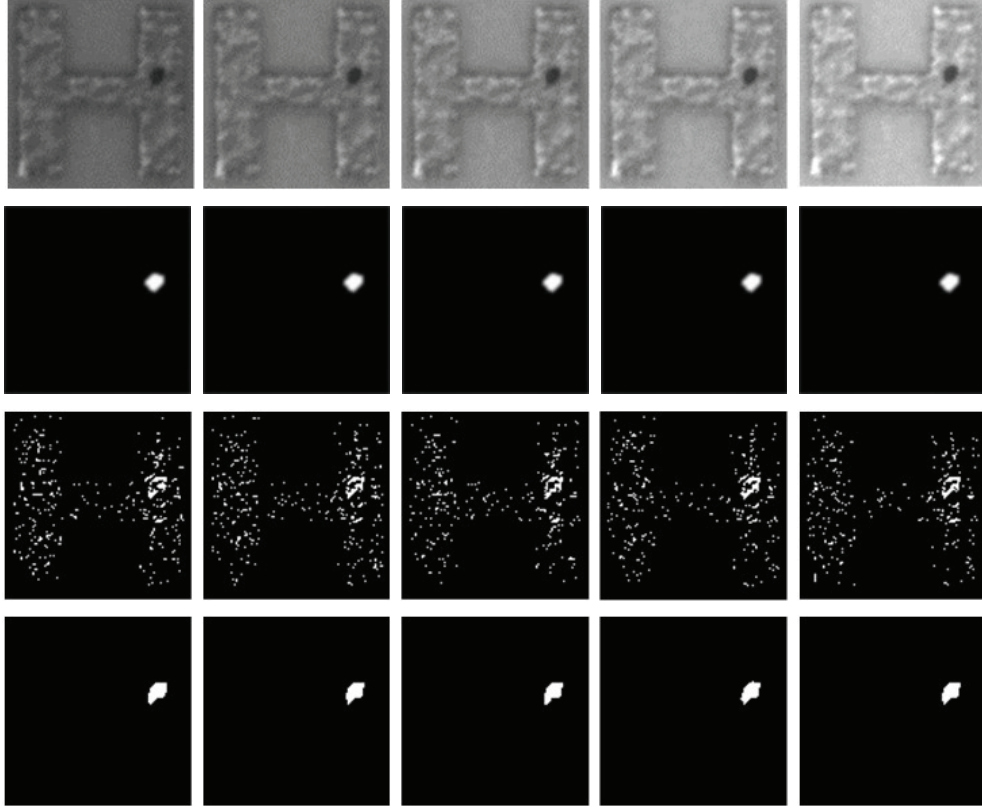


Figure 6.5: Some examples of defect detection under different illumination conditions

images, the detection results, and filtered results. The left columns, from the darker to the brighter illumination condition images. Although there is a kind of severe fluctuation in illumination, we found the proposed method could detect the real defect very similarly in size and shape from this figure, which shows the strong robustness of the orientation codes in the MPPC models.

Table 6.3 shows the pixel-level performance evaluation, while Table 6.4 shows the image-based performance evaluation. From Table 6.3, we can see that under the different illumination conditions, the performance of defect detection is very high and similar. The better performance of these automatic inspection systems in the circumstance of illumination change is really necessary for quality control in many production lines, for instance, distributed to many countries now a day.

6.3.3 Different noise intensities

In this section, we will verify whether our algorithm is robust to a certain level of noise. Here, one can understand it is not so easy to gather various noisy images, so we utilize four types of synthesized images with different amounts of additional noise, where the type of noise we selected is Gaussian. Since we have many observed images from

Table 6.3: Pixel-level performance of defect detection under different illumination conditions

Illumination condition	Filter	Recall	Precision	F-measure
Darker	OFF	0.57	0.69	0.63
	ON	0.80	0.97	0.88
Dark	OFF	0.59	0.69	0.64
	ON	0.82	0.97	0.90
Normal	OFF	0.64	0.62	0.63
	ON	0.84	0.98	0.91
Bright	OFF	0.61	0.63	0.62
	ON	0.81	0.98	0.89
Brighter	OFF	0.59	0.64	0.61
	ON	0.79	0.97	0.87

Table 6.4: Image-level based performance of defect detection under different illumination conditions

Illumination condition	Detection Rate (%)	False Alarm Rate (%)
Darker	100	0
Dark	100	0
Normal	100	0
Bright	100	0
Brighter	100	0

the real factories, we could estimate a noise level from these data, where we found in the background part the standard deviation of additional noise is around $\sigma = 3$ in the gray-scale of 256 levels. We utilize 100 artificial defect images and 100 defect-free images for each character. Fig. 6.6 shows some examples of defect detection results for different noise intensities. From the first row shows the test images, their ground truth images, the detected results, the filtered one. From the left column, no additional noise, the added version with σ intensity noise, 2σ , 3σ , respectively. One can find the performance was getting gradually worse along with increasing noise intensity, however, up to 3σ the proposed method could detect the true position although additional noise.

From Tables 6.5 and 6.6, we could find that when the intensity of the additional noise in the image is within around 2σ , the performance of defect detection was not deteriorated. However, in the case of 3σ we found some performance degradation in these experiments.

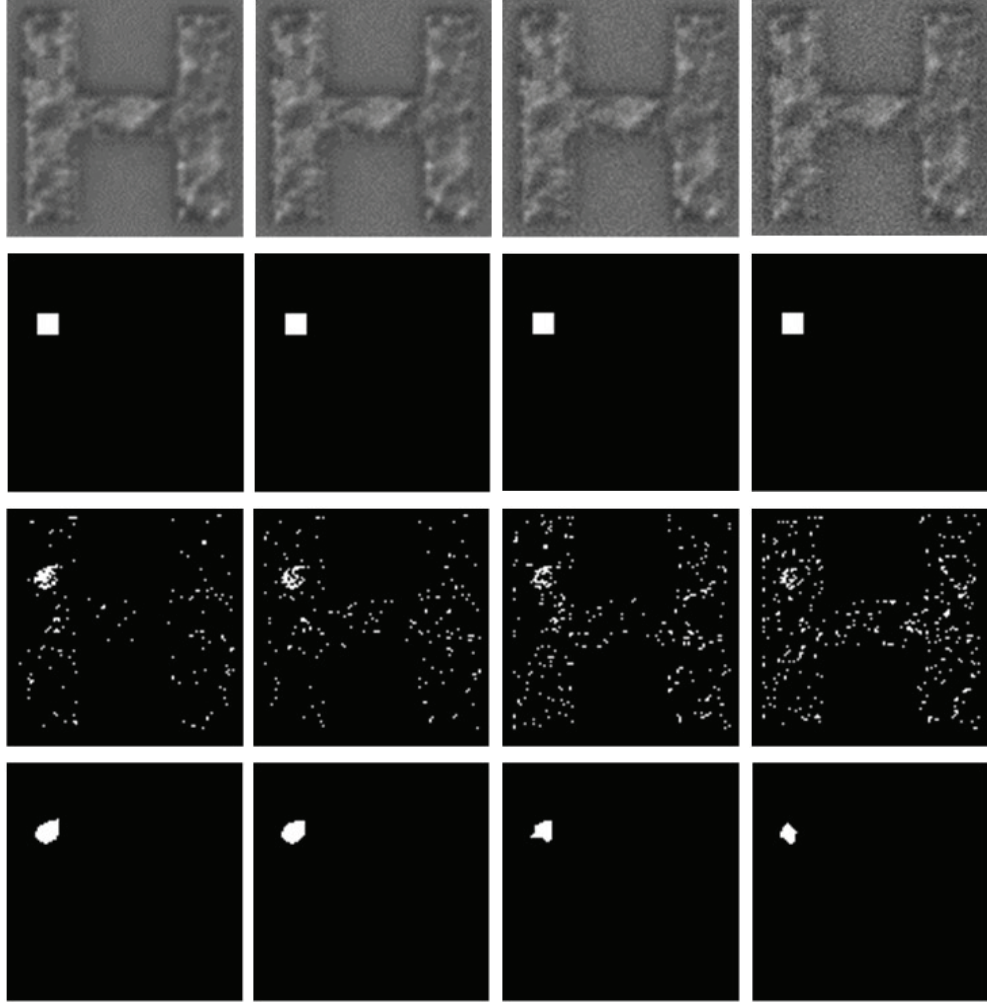


Figure 6.6: Some examples of defect detection with different amount of additional noise

Table 6.5: Pixel-level based performance of defect detection for different noise intensities

Noise intensity	Filter	Recall	Precision	F-measure
0	OFF	0.72	0.57	0.63
	ON	0.89	0.94	0.91
σ	OFF	0.68	0.56	0.61
	ON	0.84	0.93	0.88
2σ	OFF	0.65	0.54	0.59
	ON	0.79	0.90	0.84
3σ	OFF	0.64	0.53	0.58
	ON	0.78	0.90	0.84

6.3.4 Different characters

In this section, we want to consider the performance of defect detection for different printed characters. Here, six sets images which consist of 20 images for character “H”, 15

Table 6.6: Image-level based performance of defect detection for different noise intensities

Noise intensity	Detection Rate (%)	False Alarm Rate (%)
0	97	0
σ	96	0
2σ	96	1
3σ	89	7

for “U”, 20 for “A”, 5 for “W”, 20 for “E” and 15 for “I” were used, where they contain real defects. We used four cases of real defects in different positions for “H,” three for “U,” four for “A,” one for “W,” four for “E,” and then three for “I,” respectively. For the logotype “I”, we need some additive explanation that because of the size is relatively small, we could not find a sufficient number of stable support pixels in their MPPC model. Therefore, we extend the area to include a part of the neighboring “E” as shown in Fig. 6.7. This figure shows some examples of defect detection results for

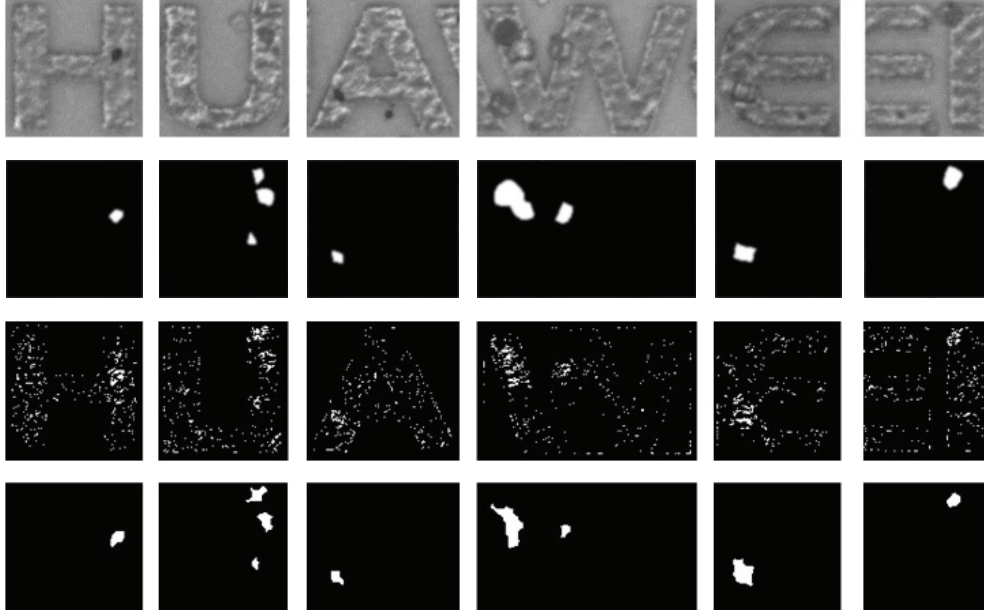


Figure 6.7: Some examples of defect detection results for different characters

each character, where in the background parts in “U”, “A” and “W” for instance, we had additive noise but they were not handled in this paper. From the first row, they are test images, their ground truth images, the detected results, the filtered results. We can see that the proposed method can locate most part of the defect area. After filtering, the over-detection noise is almost removed, and some holes are filled up.

In this paper, we have concentrated on the logotype area and excluded the background area (unprinted area). However, in Fig. 6.7, the third, fourth and last columns contain some defects in the background area. To solving these problems, we will extend our algorithm to do defect detection in the background area.

Table 6.7 shows the pixel-level based performance evaluation, while Table 6.8 shows

Table 6.7: Pixel-level based performance of defect detection for different characters

Character	Filter	Recall	Precision	F-measure
H	OFF	0.62	0.65	0.63
	ON	0.83	0.97	0.90
U	OFF	0.61	0.60	0.60
	ON	0.80	0.92	0.86
A	OFF	0.65	0.62	0.64
	ON	0.88	0.96	0.92
W	OFF	0.64	0.64	0.64
	ON	0.78	0.89	0.83
E	OFF	0.66	0.62	0.64
	ON	0.87	0.89	0.88
I	OFF	0.58	0.6	0.59
	ON	0.73	0.95	0.83

Table 6.8: Image-level based performance of defect detection for different characters

Character	Detection Rate (%)	False Alarm Rate (%)
H	100	0
U	100	0
A	100	0
W	100	0
E	100	0
I	100	0

the image-based performance evaluation. From these two tables, we could see that the performance for each character is very high and similar. After filtering, the recall and precision improved substantially, and the F-measure reached around 0.85. This shows our method has achieved good results. For the image-based evaluation, all images with defects were successfully identified. This shows the stability of our method.

6.3.5 Comparison with other methods

With the reference to several paper on defect detection, we have found that many researchers did the defect detection for surface inspection, and the materials commonly considered in the defect detection field are steel [86, 87], textile [88, 89], and wood [90, 91]. And for printing inspections, the researchers are generally carried out on paper materials [92] and capsules [93]. However, we have not found any research on the detection of defects in the materials considered in the present study. For verification of the effectiveness of proposed method, however, we selected three methods to compare. They are

the prior knowledge guided least squares regression (PG-LSR) for fabric defect inspection [36], the phase only transform (PHOT) for surface defect detection [94], and the modified RPCA with noise term and defect prior (PN-RPCA) for fabric inspection [95]. All of these three methods are unsupervised and contain no learning stage or information on the texture being inspected. And they are very effective for the detection of defects in fabrics. In PG-LSR, the authors combined the global structure of texture surface feature space and a prior from local similarity seamlessly, which makes it more robust and locate the defect more precisely. PN-RPCA is based on joint low-rank and sparse matrix recovery, and then introduces a noisy term that makes it more robust.

Here, we utilize 50 synthesis defect images and 20 real defect images for comparison experiments. Their implementation was conducted by use of the source code disclosed, and the selection of parameters were the recommended ones by the authors, respectively.

Fig. 6.8 and Fig. 6.9 show some representative results for the synthesized and the real defects, respectively. From the first row, they are test images, their ground truth images, the detected results by PG-LSR, PHOT, PN-RPCA, the proposed method and the proposed method with modification, respectively.

As we can see from Fig. 6.8 and Fig. 6.9, around a half of defects were detected by all the four methods are roughly consistent with the ground truth. For high-contrast defects, they could get rather good detection performance. But for low-contrast ones, PG-LSR, PHOT, AND PN-RPCA did not have good performances, where the materials used in the experiments were somewhat different from fabrics in texture. Fabric material has some regular textures, but the texture of our materials are almost random and no regular pattern. Table 6.9 shows the pixel-level based performance evaluation, while

Table 6.9: Pixel-level based performance of defect detection for comparing

	Defect type	Recall	Precision	F-measure
PG-LSR	Synthesized	0.35	0.34	0.35
	Real	0.46	0.44	0.45
PHOT	Synthesized	0.49	0.49	0.49
	Real	0.53	0.87	0.66
PN-RPCA	Synthesized	0.74	0.51	0.61
	Real	0.51	0.51	0.51
MPPC	synthesized	0.89	0.96	0.91
	Real	0.83	0.97	0.90
MPPC+	synthesized	0.92	0.93	0.93
	Real	0.88	0.98	0.93

Table 6.10 shows the image-based performance evaluation. From them, we could see that PG-LSR could successfully detect about a half of defects, while PHOT and PN-RPCA could successfully detect about 70% defects.

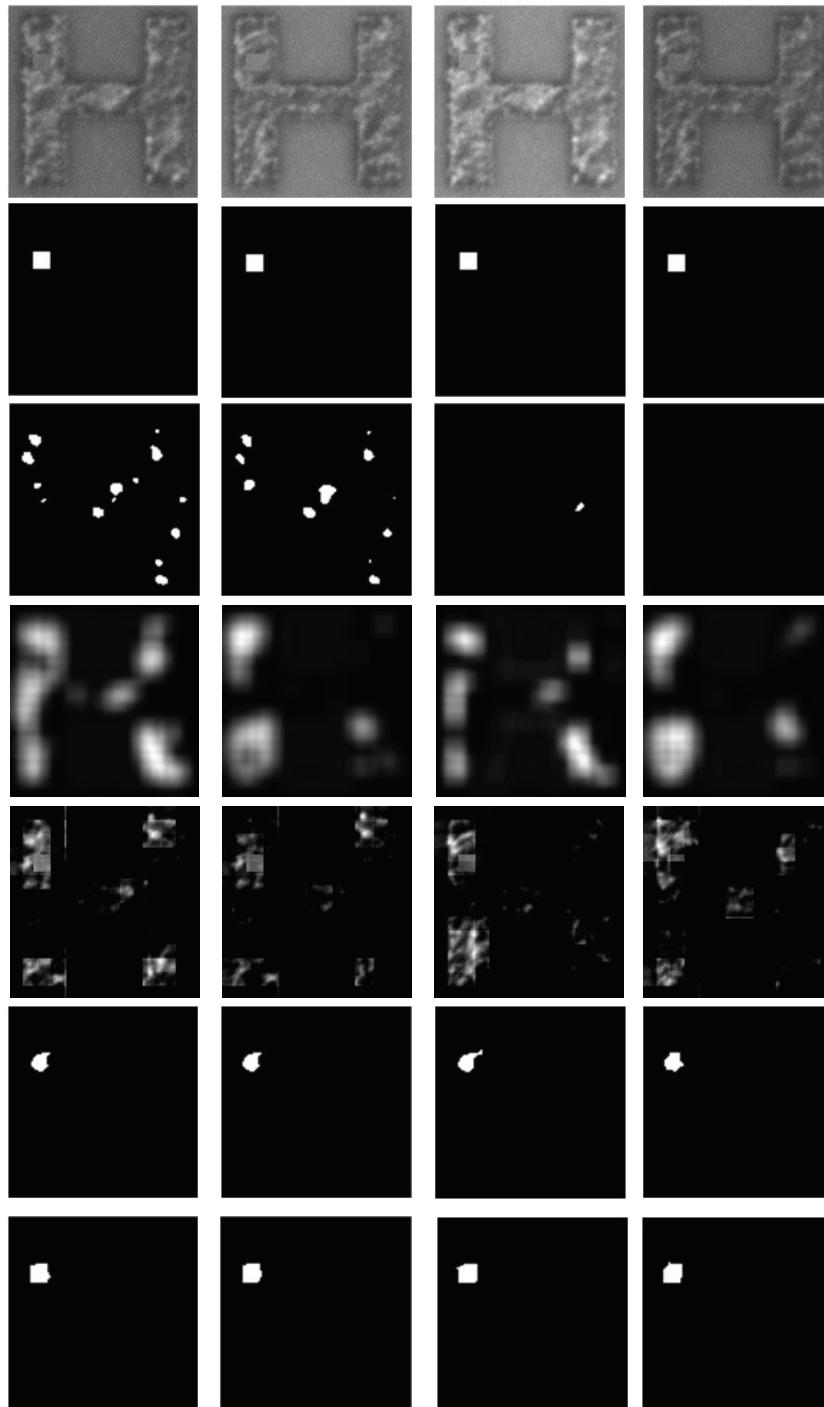


Figure 6.8: Some examples of defect detection results for synthesized defect

6.4 Computation cost

Because our algorithm is designed to apply to actual production lines, its computational efficiency should meet certain requirements. In this section, we consider our

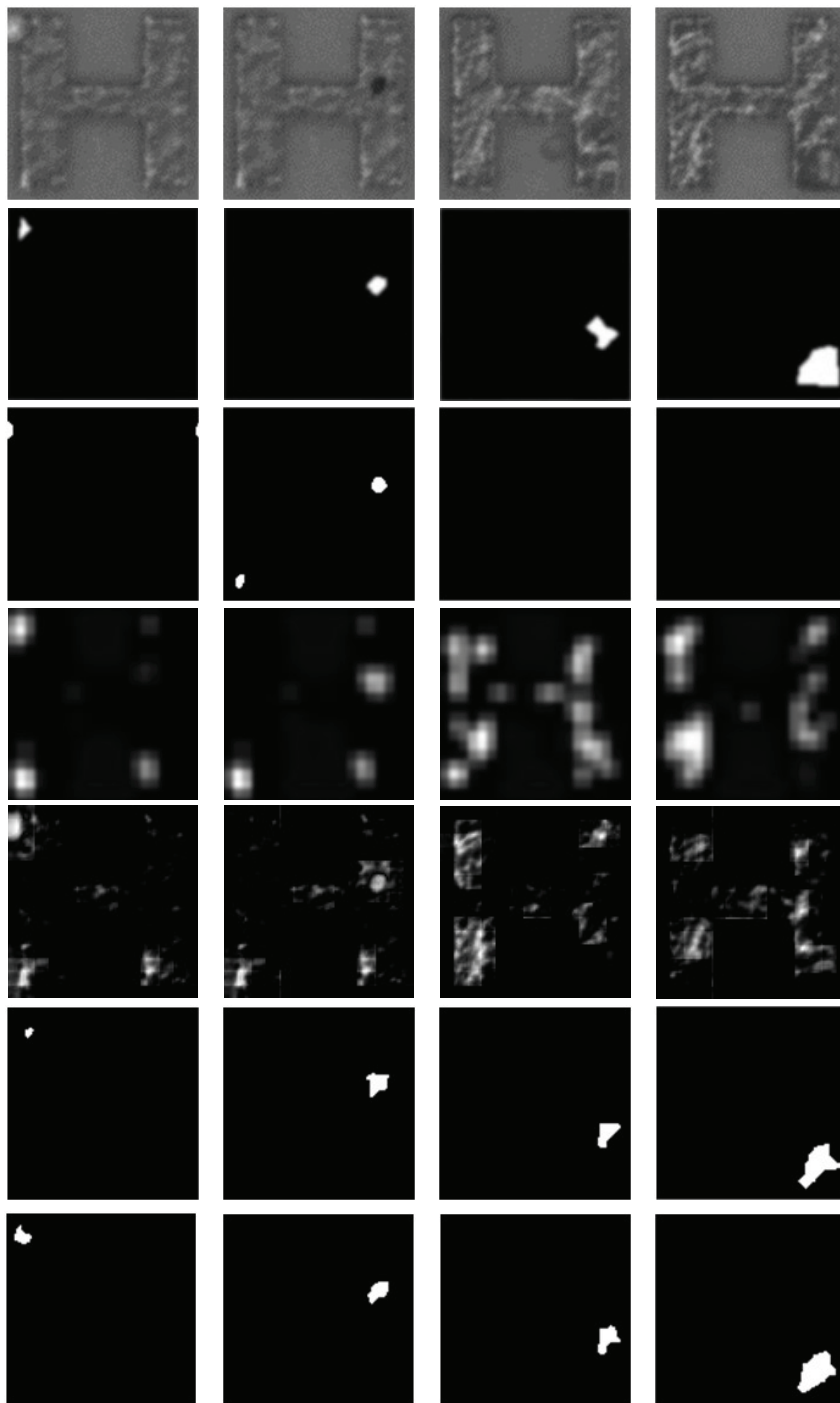


Figure 6.9: Some detected defects in the real defect images

proposed method's computation costs. We consider two aspects: one is based on images, and the other is based on pixels. We used a Windows 10 PC with an Intel Xeon

Table 6.10: Image-level based performance of defect detection for comparing

	Defect type	PG-LSR	PHOT	PN-RPCA	MPPC	MPPC+
Detection Rate (%)	Synthesized	48	66	66	98	98
	Real	65	50	80	100	100

3.5 GHz CPU and 64 GB of RAM.

The proposed algorithm proposed consists of two parts: training stage and detection stage. First, when we consider the computation time in the training stage, 100 defect-free 110×111 images were used to create the defect-free MPPC models. We used two versions of the programs: MATLAB and C++. The development environments were MATLAB 2016a and Visual Studio 2013 together with the Opencv2.4.11 library.

For the MATLAB version, we used the built-in timing function. For the C++ version, we used the clock timing function. In the MATLAB version, the entire training process required 9548.69 s for original MPPC and 10694.53s for MPPC with PDI, while in C++, 111.06 s were required for the original MPPC, and 119.94s for MPPC with PDI. For each pixel, the training stage took 0.78 s and 0.88 s in MATLAB, and 9.09 ms and 9.82 ms in C++.

Next, the computation costs were considered in the detection stage for the same image size. In the detection stage, the computation costs of the original MPPC and MPPC with PDI were the same. For the MATLAB version, the computation time for each image was 19.58 ms.

Because the computing time is short, the timing is susceptible to some operations, for example, swapping. Table. 6.11 shows the computation time in the C++ version. We

Table 6.11: Computation time in the detecting stage

NO	computation cost <i>ms</i>
1	0.989
2	0.989
3	0.849
4	0.989
5	0.971
6	0.861
7	0.928
8	0.983
Average	0.945

measured the average computation time 0.945 ms from eight trials by use of the same image, therefore measured the average computation time as 0.945 ms from ten trials using the same image, which resulted in $1.6 \mu s$ per pixel using MATLAB, an $0.0773 \mu s$ per pixel using C++.

6.5 Discussion

Compared to the defect detection methods based on deep neural networks, MPPC is a simple algorithm in structure and training data preparation. MPPC is a statistical modeling-based method used to extract the defect-free information to detect any outliers representing defects in an image. Unlike deep neural network algorithms, our method does not require any negative (or defective) samples, and requires only positive (or defect-free) example for training. Therefore, the cost of MPPC regarding in training data preparation is low. Furthermore, collecting defective samples is extremely difficult in actual production. With the improvement of production technology, defects tend to be small and difficult to detect, and the probability of their occurrence is very low. This increases the difficulty of collecting defective examples required to train deep neural networks.

To upgrade the performance of MPPC, we proposed the position-dependent data inhibition (PDI) modification and the experimental results demonstrate the effectiveness of PDI. If the training data set contains a small amount of outlier data or the training data contains defects, the outliers can be inhibited by using PDI. This can improve the effectiveness of the trained MPPC model. Therefore, after introducing PDI, the cost of preparing the training data set is further reduced.

However, MPPC has some disadvantages. MPPC focus on the logotype area and excludes the background area (unprinted area). However, in Fig. 5.9, the first, second, third, and last columns do contain some defects in the background area. Fig. 6.10 shows three typical types of defects: only in the logo, not only in the logo but also in

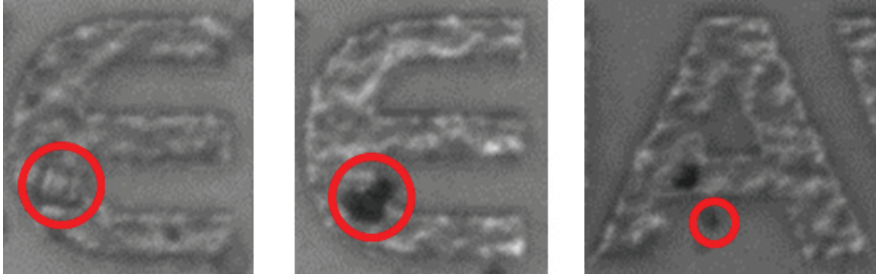


Figure 6.10: Three types of real defects

the background, and only in background. In this study, we concentrated on the first type. From left to right shows the case of defect is only in the logo part, defect is in the logo and in the background, defect is only in the background, respectively. Because our method uses OC as basis, the contrast between adjacent pixels in the background is relatively low, so we cannot obtain effective information. To solve this issue, we will extend our algorithm to perform defect detection in the background area by introducing existing background defect detection method [96].

6.6 Summary

In the chapter, we first introduced the design of experiments in detail, and then we proved the performance of MPPC and MPPC+PDI. we did follow experiments to verify the performance of proposed method.

- Using different sizes of defects tests the limitation of proposed method for defect detection.
- Using images difference illumination conditions test the robustness of the proposed algorithm to illumination fluctuations.
- Using images difference noise conditions test the robustness of the proposed algorithm to noise.
- Using different characteristics and defect types test the performance of proposed method.

We found the performance of the proposed algorithm is extremely stable and highly accurate, and the recall, precision, and F-measure in most of the results can reach 0.85, 0.93, and 0.9, respectively. In addition, the defect detection rate can reach almost 100%. These results demonstrate that the proposed approach can achieve state-of-the-art accuracy in real industrial applications. Moreover, we also discussed the time consumption, it can meet real applications' requirements.

Chapter 7. Conclusions and Future Works

7.1 Conclusions

With the objective of detecting defects on printed logotypes on micro 3D textured surfaces, an MMPC model has been proposed in this study. The defects are regarded as abnormalities on printed logotypes and can be measured at the pixel level. First, to obtain precise detection results, we propose a precise spatial differentiation for realizing better spatial sensitivity. In addition, we use kurtosis to determine the potential distribution and analyze the statistical relationship of any pair of pixels. Then, based on the statistical relationship obtained after the analysis, we establish the MPPC model for the defect-free samples. Finally, we perform defect detection based on the obtained MPPC model. The MPPC model consists of both local and global features for each pixel and realizes accurate detection, even under extreme environments, such as illumination fluctuations or noisy conditions.

Furthermore, a scheme called PDI was proposed to improve the performance of the MPPC model. The PDI was designed to handle the problem of the training data containing noisy data. The PDI improved the effectiveness of robustness of the MPPC model. Furthermore, the PDI also helped the MPPC model to resist interference under conditions of noisy data. This scheme is expected to serve as a general type of data inhibition in other statistical systems. The experimental results have demonstrated the potential of the proposed modification method.

To validate the performance of the proposed MPPC and MPPC+PDI model, data obtained from real-world industrial products was collected to obtain a dataset. This dataset consists of two types of defects: real and synthesized defects. For the synthesized defects, we used 100 images for each character, and for real defects, we used six image sets, consisting of 20 images for the character ‘H,’ 15 for ‘U,’ 20 for ‘A,’ 5 for ‘W,’ 20 for ‘E,’ and 15 for ‘I’, which are captured under different illumination conditions. Even with various defect sizes, illumination conditions, noise intensities, and other characteristics, the performance of the proposed algorithm was extremely stable and highly accurate; the recall, precision, and F-measure in majority of the results reached value of 0.85, 0.93, and 0.9, respectively. In addition, the defect detection rate reached almost 100%. This demonstrates that the proposed approach can achieve state-of-the-art accuracy in real industrial applications.

7.2 Future works

The PDI provided a new scheme for improving the effectiveness and robustness of the MPPC model. In future works, PDI can serve as a general type of data inhibition in other statistical systems. In the present study, the proposed MPPC model solves the problems of defect detection in printed logotypes on micro 3D textured surfaces. In future works, the MPPC model will be a novel feature of machine vision classification; it could be used as a feature in deep learning methods.

Moreover, in the future work following this research, we intend to classify inspection defects. In addition, the MPPC model can be integrated into other fields, such as image matching and object detection.

Defect-type classification

With the development of industrial automation and intelligent control, defect detection systems have been widely used in manufacturing industries. It can provide a product surface analysis report that can be used to control and improve the quality of the product. Defect classification plays the most important role in this system. There are many types of defects that can occur on the surface of the product, such as missed printing and scratches. Various defects have different adverse effects on the surface quality of the product. In the market, products have different tolerance standards for various defects. Therefore, it is necessary to identify the defect type using pattern recognition technology. The defect classification results provide effective information for the quality classification of the product surface.

Image matching

Image matching has laid the foundation for the robot vision system in various fields such as navigation, object positioning and recognition, and computer-aided diagnosis and treatment. It comprises a search for pictures based on a reasonable quantitative similarity measure and locating their position at the best similarity value. Moreover, the majority of practical applications suffer from uncontrollable disturbances, such as light fluctuations, low light, noisy environments, occlusions, and highlights, which often result in high false matching rates in the case of some algorithms. The MPPC model we propose is robust, and we intend to apply it in the field of image matching.

Object detection

Object detection plays an important role in various tasks such as video surveillance, traffic monitoring, scene background initialization, and object tracking. The MPPC model provides a new object detection strategy. The object to be detected can be

considered as a defective part, and the background can be designated as a defect-free area, which is similar to the defect detection problem. The MPPC model can resist illumination changes and noise and does not require a large number of samples; an MPCC model can be built using a small number of samples and without object information. These advantages can be adapted to actual applications. In the future, we intend to apply the MPPC model to foreground-detection applications.

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Takayuki Tanaka. Thank you for give me many valuable suggestions of my research when I presented my research in GM. They have helped me a lot with my research and writing the papers. And I also thank you for your daily help. Especially in the last year when Prof. Kaneko retired, you became my supervisor. You helped me a lot so that I could finish my PhD thesis successfully. Thank you very much for your help.

Thank you very much, Prof. Masahiko Onosato and Prof. Satoshi Kanai, for reviewing my thesis and providing so many insightful advices and valuable comments to complete the thesis and prepare the presentation.

I would like to thanks my supervisor Prof. Shun'ichi Kaneko, for his guidance and support all the time. I came to Japan in 2015 to start my academic journey. I started out as a novice in the field of image processing, and it was Prof. Kaneko who brought me into the world of image processing and pattern recognition. He was very patient with me. He taught me how to find the core of the problem, then how to analyze the problem, and finally how to solve it. In these five years, I learned from Prof. Kaneko how to think philosophically and scientifically and how to face challenges and difficulties. These things will be an invaluable asset for my future life and research. Even when he retired last year, he was still able to maintain his guidance of me. Thank you so much for all your help and guidance during these five years, which will be the important five years of my life.

Thanks to Mr. Hirokazu Asano at Kamakura Koki Co. Ltd. for his suggestions and discussions given me a deeper understanding of practical engineering issues. Thanks Dr. Dong Liang from Nanjing University of Aeronautics and Astronautics, China for his discussions and comments to improve my papers. Thank you very much for Ms. Miki Kikkawa and Ms. Masae Michinaka for their endless support on both of my research and my daily life.

Thanks to all the members in HCE laboratory, and it is an honor to work with them. Thanks to Dr. Wenjun Zhou, my senior and good friend, for helping me in research and daily life. Thanks to Dr. Takashi Kusaka, Dr. Yoshio Tsuchiya, Dr. Asahi Matsuda, Dr. Kotaro Hashimoto and Dr. Saori Miyajima. I learned a lot things from them. Thanks for my friends, Dr. Yaping Yan, Ms. Ying Tian, and Hongyuan Ren, for their daily helps. Thank you all for these wonderful and unforgettable days with them.

Finally, I would thanks my parents and my sister's family. Thank you for their unconditional support and understanding during the past days so that I could successfully

complete my PhD program. Thank you for their encouragement and cheering when I encountered challenges and was depressed. Thanks to them, I love them.

References

- [1] “Product introduction page of shopping website,” <https://www.yodobashi.com/product/100000001003433942/>.
- [2] “Product introduction page of shopping website,” <https://www.yodobashi.com/product/100000001004689001/>.
- [3] “An example of manual inspection,” https://www.working-images.co.uk/ngg_tag/quality-inspection-line/nggallery/slideshow.
- [4] “An example of automatic inspection,” <https://www.cims.com/zh-hans/product-avi-2/>.
- [5] X. Xie, “A review of recent advances in surface defect detection using texture analysis techniques,” *ELCVIA: electronic letters on computer vision and image analysis*, pp. 1–22, 2008.
- [6] H. Y. Ngan, G. K. Pang, and N. H. Yung, “Automated fabric defect detection 襁蚤 review,” *Image and vision computing*, vol. 29, no. 7, pp. 442–458, 2011.
- [7] I.-S. Tsai, C.-H. Lin, and J.-J. Lin, “Applying an artificial neural network to pattern recognition in fabric defects,” *Textile Research Journal*, vol. 65, no. 3, pp. 123–130, 1995.
- [8] H.-F. Ng, “Automatic thresholding for defect detection,” *Pattern recognition letters*, vol. 27, no. 14, pp. 1644–1649, 2006.
- [9] M. Pietikäinen, T. Mäenpää, and J. Viertola, “Color texture classification with color histograms and local binary patterns,” in *Workshop on Texture Analysis in Machine Vision*, vol. 1. Citeseer, 2002.
- [10] B. Thai and G. Healey, “Modeling and classifying symmetries using a multiscale opponent color representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1224–1235, 1998.
- [11] X. Xie, M. Mirmehdi, and B. Thomas, “Colour tonality inspection using eigenspace features,” *Machine Vision and Applications*, vol. 16, no. 6, pp. 364–373, 2006.
- [12] X.-c. Yuan, L.-s. Wu, and Q. Peng, “An improved otsu method using the weighted object variance for defect detection,” *Applied surface science*, vol. 349, pp. 472–484, 2015.

- [13] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.
- [14] R. W. Connors, C. W. Mcmillin, K. Lin, and R. E. Vasquez-Espinosa, "Identifying and locating surface defects in wood: Part of an automated lumber processing system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 573–583, 1983.
- [15] J. L. Raheja, B. Ajay, and A. Chaudhary, "Real time fabric defect detection system on an embedded dsp platform," *Optik*, vol. 124, no. 21, pp. 5280–5284, 2013.
- [16] J. L. Raheja, S. Kumar, and A. Chaudhary, "Fabric defect detection based on glcm and gabor filter: A comparison," *Optik*, vol. 124, no. 23, pp. 6469–6474, 2013.
- [17] A. Monadjemi, "Towards efficient texture classification and abnormality detection," Ph.D. dissertation, University of Bristol, 2004.
- [18] J. Iivarinen, J. Rauhamaa, and A. Visa, "Unsupervised segmentation of surface defects," in *Proceedings of 13th international conference on pattern recognition*, vol. 4. IEEE, 1996, pp. 356–360.
- [19] S. Odemir, A. Baykut, R. Meylani, A. Erçil, and A. Ertuzun, "Comparative evaluation of texture analysis algorithms for defect inspection of textile products," in *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, vol. 2. IEEE, 1998, pp. 1738–1740.
- [20] G. Capizzi, G. L. Sciuto, C. Napoli, E. Tramontana, and M. Woźniak, "Automatic classification of fruit defects based on co-occurrence matrix and neural networks," in *2015 Federated Conference on Computer Science and Information Systems (Fed-CSIS)*. IEEE, 2015, pp. 861–867.
- [21] M. P. Samy, S. Foong, G. S. Soh, and K. S. Yeo, "Automatic optical & laser-based defect detection and classification in brick masonry walls," in *2016 IEEE Region 10 Conference (TENCON)*. IEEE, 2016, pp. 3521–3524.
- [22] Y. Huang and K. L. Chan, "Texture decomposition by harmonics extraction from higher order statistics," *IEEE Transactions on Image Processing*, vol. 13, no. 1, pp. 1–14, 2004.
- [23] C. Coroyer, D. Declercq, and P. Duvaut, "Texture classification using third order correlation tools," in *Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics*. IEEE, 1997, pp. 171–175.
- [24] D. Zhu, R. Pan, W. Gao, and J. Zhang, "Yarn-dyed fabric defect detection based on autocorrelation function and glcm," *Autex research journal*, vol. 15, no. 3, pp. 226–232, 2015.

-
- [25] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [26] M. Niskanen, H. Kauppinen, and O. Silven, "Real-time aspects of som-based visual surface inspection," in *Machine Vision Applications in Industrial Inspection X*, vol. 4664. International Society for Optics and Photonics, 2002, pp. 123–134.
- [27] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [28] T. Mäenpää, M. Turtinen, and M. Pietikäinen, "Real-time surface inspection by texture," *Real-Time Imaging*, vol. 9, no. 5, pp. 289–296, 2003.
- [29] L. Zhang, J. Jing, and H. Zhang, "Fabric defect classification based on lbp and glcm," *Journal of Fiber Bioengineering and Informatics*, vol. 8, no. 1, pp. 81–89, 2015.
- [30] V. A. Sindagi and S. Srivastava, "Oled panel defect detection using local inlier-outlier ratios and modified lbp," in *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*. IEEE, 2015, pp. 214–217.
- [31] J. Chen and A. K. Jain, "A structural approach to identify defects in textured images," in *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1. IEEE, 1988, pp. 29–32.
- [32] M. Bennamoun and A. Bodnarova, "Automatic visual inspection and flaw detection in textile materials: Past, present and future," in *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*, vol. 5. IEEE, 1998, pp. 4340–4343.
- [33] K. Y. Song, J. Kittler, and M. Petrou, "Defect detection in random colour textures," *Image and vision computing*, vol. 14, no. 9, pp. 667–683, 1996.
- [34] A. Tolba and H. M. Raafat, "Multiscale image quality measures for defect detection in thin films," *The International Journal of Advanced Manufacturing Technology*, vol. 79, no. 1-4, pp. 113–122, 2015.
- [35] J. P. Yun, S. J. Lee, G. Koo, C. Shin, and C. Park, "Automatic defect inspection system for steel products with exhaustive dynamic encoding algorithm for searches," *Optical Engineering*, vol. 58, no. 2, p. 023107, 2019.
- [36] J. Cao, J. Zhang, Z. Wen, N. Wang, and X. Liu, "Fabric defect inspection using prior knowledge guided least squares regression," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4141–4157, 2017.

- [37] B. Mallik-Goswami and A. K. Datta, "Detecting defects in fabric with laser-based morphological image processing," *Textile Research Journal*, vol. 70, no. 9, pp. 758–762, 2000.
- [38] K.-L. Mak, P. Peng, and K. F. C. Yiu, "Fabric defect detection using morphological filters," *Image and Vision Computing*, vol. 27, no. 10, pp. 1585–1592, 2009.
- [39] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.
- [40] M. Unser and F. Ade, "Feature extraction and decision procedure for automated inspection of textured materials," *Pattern Recognition Letters*, vol. 2, no. 3, pp. 185–191, 1984.
- [41] A. Monadjemi, M. Mirmehdi, and B. Thomas, "Restructured eigenfilter matching for novelty detection in random textures," *learning*, vol. 5, p. 13, 2004.
- [42] D.-M. Tsai and C.-Y. Hsieh, "Automated surface inspection for directional textures," *Image and Vision computing*, vol. 18, no. 1, pp. 49–62, 1999.
- [43] D.-M. Tsai and T.-Y. Huang, "Automated surface inspection for statistical textures," *Image and Vision computing*, vol. 21, no. 4, pp. 307–323, 2003.
- [44] C.-h. Chan and G. K. Pang, "Fabric defect detection by fourier analysis," *IEEE transactions on Industry Applications*, vol. 36, no. 5, pp. 1267–1276, 2000.
- [45] L. M. Hoffer, F. Francini, B. Tiribilli, and G. Longobardi, "Neural networks for the optical recognition of defects in cloth," *Optical Engineering*, vol. 35, no. 11, pp. 3183–3190, 1996.
- [46] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision research*, vol. 20, no. 10, pp. 847–856, 1980.
- [47] K. L. Mak and P. Peng, "An automated inspection system for textile fabrics based on gabor filters," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 3, pp. 359–369, 2008.
- [48] A. Bodnarova, M. Bennamoun, and S. Latham, "Optimal gabor filters for textile flaw detection," *Pattern recognition*, vol. 35, no. 12, pp. 2973–2991, 2002.
- [49] M. R. Turner, "Texture discrimination by gabor functions," *Biological cybernetics*, vol. 55, no. 2-3, pp. 71–82, 1986.
- [50] D.-M. Tsai and B. Hsiao, "Automatic surface inspection using wavelet reconstruction," *Pattern Recognition*, vol. 34, no. 6, pp. 1285–1305, 2001.

-
- [51] M. Ralló, M. Millán, J. Escofet, and R. Navarro, “Wavelet based techniques for textile inspection,” *Opt. Eng.*, vol. 26, no. 2, pp. 838–844, 2003.
 - [52] M. Mufti and G. Vachtsevanos, “Automated fault detection and identification using a fuzzy-wavelet analysis technique,” in *Conference Record AUTOTEST-CON’95. Systems Readiness: Test Technology for the 21st Century*. IEEE, 1995, pp. 169–175.
 - [53] G. Lambert and F. Bock, “Wavelet methods for texture defect detection,” in *Proceedings of International Conference on Image Processing*, vol. 3. IEEE, 1997, pp. 201–204.
 - [54] Y. Han and P. Shi, “An adaptive level-selecting wavelet transform for texture defect detection,” *Image and Vision Computing*, vol. 25, no. 8, pp. 1239–1248, 2007.
 - [55] X. Z. Yang, G. K. Pang, and N. H. C. Yung, “Discriminative fabric defect detection using adaptive wavelets,” *Optical Engineering*, vol. 41, no. 12, pp. 3116–3126, 2002.
 - [56] X. Yang, G. Pang, and N. Yung, “Discriminative training approaches to fabric defect classification based on wavelet transform,” *Pattern Recognition*, vol. 37, no. 5, pp. 889–899, 2004.
 - [57] B. Mandelbrodt, “The fractal geometry of nature,” 1983.
 - [58] P. P. Ohanian and R. C. Dubes, “Performance evaluation for four classes of textural features,” *Pattern recognition*, vol. 25, no. 8, pp. 819–833, 1992.
 - [59] R. M. Haralick, “Statistical and structural approaches to texture,” *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.
 - [60] A. L. Serafim, “Segmentation of natural images based on multiresolution pyramids linking of the parameters of an autoregressive rotation invariant model. application to leather defects detection,” in *11th IAPR International Conference on Pattern Recognition. Vol. III. Conference C: Image, Speech and Signal Analysis*, vol. 1. IEEE Computer Society, 1992, pp. 41–44.
 - [61] Y. Zhang, Y. Xing, Y. Gong, D. Jin, H. Li, and F. Liu, “A variable-level automated defect identification model based on machine learning,” *Soft Computing*, vol. 24, no. 2, pp. 1045–1061, 2020.
 - [62] G. R. Cross and A. K. Jain, “Markov random field texture models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 25–39, 1983.
 - [63] F. Gayubo, J. L. Gonzalez, E. de la Fuente, F. Miguel, and J. R. Peran, “On-line machine vision system for detect split defects in sheet-metal forming processes,” in *18th International Conference on Pattern Recognition (ICPR’06)*, vol. 1. IEEE, 2006, pp. 723–726.

- [64] K. Xu, M. Song, C. Yang, P. Zhou *et al.*, “Application of hidden markov tree model to on-line detection of surface defects for steel strips,” 2013.
- [65] F. López, J. M. Valiente, R. Baldrich, and M. Vanrell, “Fast surface grading using color statistics in the cie lab space,” in *Iberian conference on pattern recognition and image analysis*. Springer, 2005, pp. 666–673.
- [66] H.-b. LIU and G.-w. KANG, “Surface defects inspection of cold rolled strips based on neural network,” *Journal of Image and Graphics*, vol. 10, no. 10, p. 1, 2005.
- [67] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, “Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection,” *CIRP Annals*, vol. 65, no. 1, pp. 417–420, 2016.
- [68] J.-K. Park, B.-K. Kwon, J.-H. Park, and D.-J. Kang, “Machine learning-based imaging system for surface defect inspection,” *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 3, no. 3, pp. 303–310, 2016.
- [69] L. Song, W. Lin, Y.-G. Yang, X. Zhu, Q. Guo, and J. Xi, “Weak micro-scratch detection based on deep convolutional neural network,” *IEEE Access*, vol. 7, pp. 27 547–27 554, 2019.
- [70] Y.-J. Cha, W. Choi, and O. Büyüköztürk, “Deep learning-based crack damage detection using convolutional neural networks,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.
- [71] S. Ghorai, A. Mukherjee, M. Gangadaran, and P. K. Dutta, “Automatic defect detection on hot-rolled flat steel products,” *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 3, pp. 612–621, 2012.
- [72] A. Gururajan, E. F. Hequet, and H. Sari-Sarraf, “Objective evaluation of soil release in fabrics,” *Textile Research Journal*, vol. 78, no. 9, pp. 782–795, 2008.
- [73] Q. Zhu, M. Wu, J. Li, and D. Deng, “Fabric defect detection via small scale over-complete basis set,” *Textile Research Journal*, vol. 84, no. 15, pp. 1634–1649, 2014.
- [74] S. Mei, H. Yang, and Z. Yin, “An unsupervised-learning-based approach for automated defect inspection on textured surfaces,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 6, pp. 1266–1277, 2018.
- [75] F. Ullah and S. Kaneko, “Using orientation codes for rotation-invariant template matching,” *Pattern recognition*, vol. 37, no. 2, pp. 201–209, 2004.
- [76] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1. IEEE, 2005, pp. 886–893.

-
- [77] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of solid-state circuits*, vol. 23, no. 2, pp. 358–367, 1988.
- [78] S. Xiang, D. Liang, S. Kaneko, and H. Asano, "Robust defect detection in 2d images printed on 3d micro-textured surfaces by multiple paired pixel consistency in orientation codes," *IET Image Processing*, 2020.
- [79] D. Liang, M. Hashimoto, K. Iwata, X. Zhao *et al.*, "Co-occurrence probability-based pixel pairs background model for robust object detection in dynamic scenes," *Pattern Recognition*, vol. 48, no. 4, pp. 1374–1390, 2015.
- [80] L. He, Y. Chao, K. Suzuki, and K. Wu, "Fast connected-component labeling," *Pattern Recognition*, vol. 42, no. 9, pp. 1977–1987, 2009.
- [81] N. Bouaynaya and D. Schonfeld, "Spatially variant morphological image processing: theory and applications," in *Visual Communications and Image Processing 2006*, vol. 6077. International Society for Optics and Photonics, 2006, p. 60771Y.
- [82] B. Arnold and R. Göb, "Sample method and quality control," *PROBABILITY AND STATISTICS-Volume III: Applied Statistics*, vol. 2, p. 95, 2009.
- [83] W. Zhou, S. Kaneko, M. Hashimoto, Y. Satoh, and D. Liang, "Foreground detection based on co-occurrence background model with hypothesis on degradation modification in dynamic scenes," *Signal Processing*, vol. 160, pp. 66–79, 2019.
- [84] M. L. McHugh, "The chi-square test of independence," *Biochemia medica: Biochemia medica*, vol. 23, no. 2, pp. 143–149, 2013.
- [85] K. V. Mardia, "Measures of multivariate skewness and kurtosis with applications," *Biometrika*, vol. 57, no. 3, pp. 519–530, 1970.
- [86] M. Selvi and D. Jenefa, "Automated defect detection of steel surface using neural network classifier with co-occurrence features," *International Journal*, vol. 4, no. 3, 2014.
- [87] S. Zhou, Y. Chen, D. Zhang, J. Xie, and Y. Zhou, "Classification of surface defects on steel sheet using convolutional neural networks," *Mater. Technol*, vol. 51, no. 1, pp. 123–131, 2017.
- [88] P. Li, J. Liang, X. Shen, M. Zhao, and L. Sui, "Textile fabric defect detection based on low-rank representation," *Multimedia Tools and Applications*, pp. 1–26, 2017.
- [89] F. Arnia and K. Munadi, "Real time textile defect detection using glcm in dct-based compressed images," in *Modeling, Simulation, and Applied Optimization (ICM-SAO), 2015 6th International Conference on*. IEEE, 2015, pp. 1–6.

- [90] Y. Zhang, C. Xu, C. Li, H. Yu, and J. Cao, “Wood defect detection method with pca feature fusion and compressed sensing,” *Journal of forestry research*, vol. 26, no. 3, pp. 745–751, 2015.
- [91] Z.-N. Ke, Q.-J. Zhao, C.-H. Huang, P. Ai, and J.-G. Yi, “Detection of wood surface defects based on particle swarm-genetic hybrid algorithm,” in *Audio, Language and Image Processing (ICALIP), 2016 International Conference on*. IEEE, 2016, pp. 375–379.
- [92] Y. Fucheng, Z. Lifan, and Z. Yongbin, “The research of printing’s image defect inspection based on machine vision,” in *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*. IEEE, 2009.
- [93] A. Mehle, M. Bukovec, B. Likar, and D. Tomažević, “Print registration for automated visual inspection of transparent pharmaceutical capsules,” *Machine Vision and Applications*, vol. 27, no. 7, pp. 1087–1102, 2016.
- [94] D. Aiger and H. Talbot, “The phase only transform for unsupervised surface defect detection,” in *Emerging Topics In Computer Vision And Its Applications*. World Scientific, 2012, pp. 215–232.
- [95] J. Cao, N. Wang, J. Zhang, Z. Wen, B. Li, and X. Liu, “Detection of varied defects in diverse fabric images via modified rpca with noise term and defect prior,” *International Journal of Clothing Science and Technology*, vol. 28, no. 4, pp. 516–529, 2016.
- [96] Y. Yan, S. Kaneko, and H. Asano, “Accumulated and aggregated shifting of intensity for defect detection on micro 3d textured surfaces,” *Pattern Recognition*, vol. 98, p. 107057, 2020.

Appendix A Publications lists

A.1 Journal Papers

1. Xiang, S., Liang, D., Kaneko, S. and Asano, H.. Robust defect detection in 2D images printed on 3D micro-textured surfaces by multiple paired pixel consistency in orientation codes, IET Image Processing. 2020. DOI: 10.1049/iet-ipr.2019.0724.

A.2 International Conferences

1. Xiang, S., Yan, Y., Asano, H. and Kaneko, S., Robust Printing Defect Detection on 3D Textured Surfaces by Multiple Paired Pixel Consistency of Orientation Codes. In 2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics, Mie, Japan, September 10-12, pp.373-378. 2018. DOI: 10.1109/MECATRONICS.2018.8495682.

2. Xiang, S., Kaneko S. and Liang, D., Robust Method for Detecting Defect in Images Printed on 3D Micro-textured Surfaces: Modified Multiple Paired Pixel Consistency. In VISIGRAPP (5: VISAPP), Valletta, Malta, February 27-29, pp.260-267. 2020. DOI: 10.5220/0008910002600267.

A.3 Domestic Conferences

1. XIANG S., YAN Y., ASANO H. and KANEKO S., Robust Printing Quality inspection on SHIBO surfaces by Multiple Paired Pixel Consistency of Orientation Codes, 2018 JSPE Autumn Meeting. 2018

2. XIANG S., KANEKO S. and LIANG D., Robust Printing Quality Inspection on SHIBO Sur- faces by Multiple Paired Pixel Consistency with Modification, 2019 Academic Meeting of Hokkaido brach of the JSPE. 2019

3. XIANG S., KANEKO S. and LIANG D., Robust Defect Detection in Printing on 3D Micro- textured Surfaces by Multiple Paired Pixel Consistency with Modification, Vision Engineering Workshop 2019. 2019

4. XIANG S., KANEKO S. and TANAKA T., Defect Detection in Logotypes on Embossed Sur- face by Mutiple Paired Pixel Consistency with Data Inhibition Modification, The 26th Symposium on Sensing via Image Information. 2020