IET Image Processing

Research Article

IET Journals
The Institution of Engineering and Technology

Robust defect detection in 2D images printed on 3D micro-textured surfaces by multiple paired pixel consistency in orientation codes

ISSN 1751-9659
Received on 14th June 2019
Revised 25th April 2020
Accepted on 13th July 2020
E-First on 5th October 2020
doi: 10.1049/iet-ipr.2019.0724
www.ietdl.org

Sheng Xiang[1] ✉, *Dong Liang*[2], *Shun'ichi Kaneko*[1], *Hirokazu Asano*[3]

[1]*Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan*
[2]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, People's Republic of China*
[3]*Kamakura Koki Co., Ltd., Warabi 335-0002, Japan*
✉ *E-mail: xiangsheng@hce.ist.hokudai.ac.jp*

Abstract: Defect detection is now an active research area for production quality assurance. Traditional visual inspection systems are developed by human beings, which is a time-consuming, labour-intensive, and highly error-prone process, and are therefore unreliable. To overcome these problems, the authors proposed a new method for detecting defects when printing on a 3D micro-textured surface. They utilise an orientation code as the basis to resist the fluctuations in illumination. Based on the consistency of the pixel pairs, they developed a model called multiple paired pixel consistency to represent the statistical relationship between each pixel pair in defect-free images. Finally, based on this model, they designed a defect detection method. Even with different defect sizes, illumination conditions, noise intensities, and other characteristics, 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%. This demonstrates that the authors' approach can achieve state-of-the-art accuracy in real industrial applications.

1 Introduction

Defect detection plays an important role in the quality control (QC) of the manufacturing industry. In addition, the application of computer vision in an automatic quality inspection has been widely studied because an increasing number of factories have begun using automatic production lines. We are interested in assessing how currently available vision systems perform a variety of QC tasks on printed products. We mainly consider the inspection of printed characters/text or logotypes for defects, such as holes, scratches, dents, and foreign objects.

At present, the printing quality is often assessed by human inspectors, which is a labour-intensive and time-consuming job. However, the results of an inspection might be unreliable because humans may give different results depending on the time, mood, skills, and experience of the inspectors. In addition, human inspection is subjective because there is no quality standard independent of the operator. To solve these problems, human inspection is being replaced by automatic visual inspection systems [1].

Texture is an important feature of defect detection. In fact, the task of detecting defects is largely seen as a texture analysis problem. Xie [2] provided a comprehensive review of defect detection in textures, and classified texture analysis techniques into four categories: statistical [3], structural [4, 5], filter-based [6, 7], and model-based [8] methods. (1) Statistical approaches detect defects by measuring the spatial distribution to evaluate the textures, including histogram properties [9], co-occurrence matrices [10], autocorrelation [11], and local binary patterns [12]. In [12], a new noise-resistant version of LBP was proposed to extract colour and texture features jointly, and then utilise these features for defect detection. These methods are extremely effective for detecting defects on regular textures and are simple to implement. (2) Structural approaches usually consider texture to be characterised by texture primitives, and the spatial arrangement of these primitives [13]. A structural method of fabric image defect recognition was proposed in [14]. First, a texture image is threshold processed through a histogram analysis, and is mapped to a skeleton representation. Finally, histograms of the skeleton position and length are measured to identify and locate defects. Such approaches [15, 16] perform well on extremely regular textures. However, utilising structural methods to detect defects has certain drawbacks, i.e., they are not applicable to structures with a low degree of regularity or small defects. (3) Filter-based approaches can be divided into spatial-domain, frequency-domain, and joint-domain techniques. These methods apply filter banks to an image for computing the energy of the filter responses, which include the Fourier transform (FT) [17], Gabor transform [18], and orthogonal wavelet transform [19]. They are effective and simple to implement, but are ineffective with images with a complex texture. (4) Model-based approaches first extract texture features from an image through modelling and parameter estimation, and then discriminate whether a test image conforms to the normal texture model to realise a defect detection. Such approaches include the Gaussian–Markov random field [20] model and the Gaussian mixture model [21] as representatives. Their results have been satisfactory; however, the computational complexity is high, and they are not good at detecting small defects.

Deep-learning based defect detection approaches have also attracted significant attention. Tao *et al.* [22] proposed a twofold procedure to localise and classify metallic defects. First, a novel cascaded autoencoder (CASAE) architecture is utilised to localise the defect and then classify it using a convolutional neural network (CNN). Li *et al.* [23] utilised a compact CNN architecture that applies microarchitectures with a multilayer perceptron to optimise the detection of fabric defects. Krummenacher *et al.* [24] utilised novel general wavelet features and features learned from deep CNNs to detect wheel defects. However, unlike other application fields where it is easy to collect training samples, it is difficult to obtain defective samples in the field of industrial manufacturing because the probability of defects occurring in industrial manufacturing is extremely low. In addition, manually labelling the training dataset is a time-consuming and laborious task.

Herein, we deal with a printed logotype on a surface embossed with randomly distributed 3D micro-textures, as shown in Fig. 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. Due to they have an attractive appearance, good handling, and excellent slip resistance, such surfaces have been widely used in numerous products. Their three-

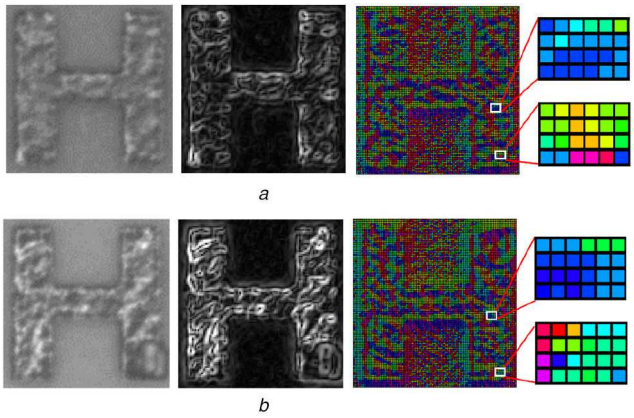

Fig. 1 *Examples of printed characters*
(a) Defect-free and, ***(b)*** Defective. From the leftmost column: raw, gradient, and OC images

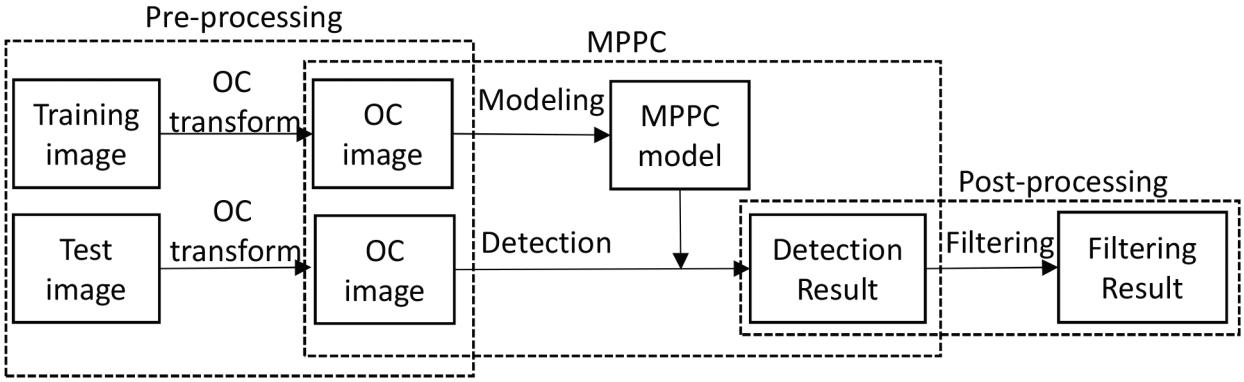

Fig. 2 *Overall block diagram*

dimensional (3D) microstructures, 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. We proposed another method for defect detection in background regions, called an accumulated and aggregated shifting of intensity, in short AASI [25], for the same type of surfaces. In this study, we concentrate on a different problem of defect detection on printed logotypes on the same objects. Although the target surfaces in the products are the same, because their logotypes have different types of defects and the statistical characteristics of the images observed from defect-free objects are very different from their backgrounds, a novel scheme for their detection is needed, as shown herein. From Fig. 1, we can see that the gradient images of typical defect-free and defective images have large differences or fluctuations within the entire area of the logotype, which makes its utilisation for defect detection difficult. Instead of a gradient feature, we propose the use of orientation code matching (OCM) [26] as a robust matching or registration. Comparing the orientation code (OC) images of defect-free and defective parts shown in Fig. 1, we find a certain similarity in the defect-free part, whereas there is rather prominent variation in the defective part, which shows that it may be possible to utilise the OC as the basis of our development.

We need another technique to develop a model of the statistical relationship between pixels in logotypes that require precision and robustness even in noisy situations. Liang *et al.* [27] proposed a co-occurrence probability-based pixel pair (CP3) background model for robust background subtraction, which is suited to our purposes.

Here, we propose a new approach for robust defect detection that utilises two principles: OCs and their statistical relationship between a pixel and its supporting pixels. The OCs are used for extracting the texture features of the image at a location with contrast, not only at an edge, and an analysis of the statistical relationship in the OCs of different pairs of pixels is conducted to obtain a defect-free model for each pixel in the logotype. We show the effectiveness of our approach experimentally for printed regions, such as logotypes, characters, and symbols, on embossed surfaces. Furthermore, based on this modelling, we propose an algorithm for defect detection mainly in the logotype regions, which any customer maybe sensitive to check the quality of their printing.

The proposed model of defect-free images, called multiple paired pixel consistency (MPPC), was briefly introduced in [28]. In this paper, we introduce MPPC with new contents and give more detailed explanations for our work with new experimental results. First, to obtain more precise detection results, we propose a precise spatial differentiation for achieving better spatial sensitivity. In addition, we utilise kurtosis to determine the potential distribution and analyse the statistical relationship of any pair of pixels. We then use an original filter. Finally, we confirm the performance of the proposed method through some comparative experiments. Our contributions of this paper are as follows: (i) this work proposes the MPPC defect-free model to realise the robustness of illumination fluctuations and noise for detecting the defect in logotypes printed on 3D micro-textured surfaces; (ii) the MPPC model is sensitive to small and vague defects. Owing our model is based on a single Gaussian model which is more sensitive, robust, and efficient, for example, than the Gaussian mixture model; (iii) only a small number of positive samples are utilised to define or train the pixel-wise elemental models, whereas in deep learning approaches, one may need two balanced training sets of defect-free and defective images that are not easy to achieve in general owing to the high-quality control required. In addition, our method works in pixel-wise rather than region. This can detect defects more precisely. Fig. 2 shows the overall block diagram of the proposed method. It contains three main parts: pre-processing, the proposed MPPC method, and post-processing. First, we converted the training image from the grey-scale image to the OC image, and then used the obtained OC image to generate the MPPC model. Third, we used the obtained MPPC to carry out a defect detection on the test image, and finally, output the final detection result with the proposed filtering algorithm. The remaining of this paper is organised as follows. Section 2 describes an improvement in the

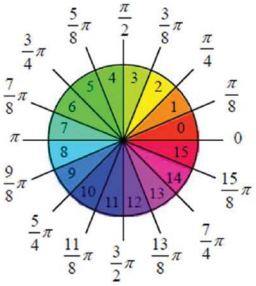

Fig. 3 *Sixteen-OC*

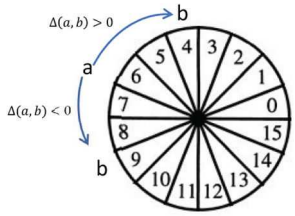

Fig. 4 *Scheme of signed difference in OC*

OCs and statistical modelling. Section 3 introduces the working mechanism of the MPPC defect-free model. Section 4 presents the defect detection using the MPPC model. Section 4.1 introduces an original filtering for improving the performance of the proposed method, and to show the effectiveness of the proposed method some experimental results are given in Section 5. Finally, some concluding remarks are given in Section 6.

2 Improvement of OCs

In this section, we first introduce the original version of the 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 the statistical relationship in a pair of any pixels on the logotypes.

2.1 Orientation codes

The OCs were proposed for applying a filtering to extract the robust features based on only the orientation information involved in gradient vectors from any images. This approach achieves strong robustness to an image representation and can resist the fluctuations in the illumination in the real world because it utilises only a gradient orientation as a discriminative feature, which represents the orientation angle rather than the strength of the gradient.

Let $I(i, j)$ be the brightness of pixel (i, j). Its partial derivatives in the horizontal and vertical directions are written as follows:

$$\nabla I_x = \frac{\partial I}{\partial x} \tag{1}$$

and

$$\nabla I_y = \frac{\partial I}{\partial y} \tag{2}$$

The orientation angle θ can be computed using

$$\theta = \tan^{-1}\left(\frac{\nabla I_y}{\nabla I_x}\right) \tag{3}$$

of which the actual orientation is determined after checking the signs of the derivatives, thus making the range of θ be $[0, 2\pi)$. The 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:

$$C_{i,j} = \begin{cases} \left\lfloor \dfrac{\theta_{i,j}}{\Delta\theta} \right\rfloor & |\nabla I_x| + |\nabla I_y| \geq \Gamma \\ N & \text{otherwise} \end{cases} \tag{4}$$

where Γ is a threshold level for ignoring pixels with low-contrast neighbourhoods. The pixels with neighbourhoods of sufficient contrast are assigned an OC of $\{0, 1, \ldots, N-1\}$, whereas the ignored pixels are assigned the code N. Fig. 3 shows the code system used in this paper, where $\Gamma = 10$ and $N = 16$.

2.2 Signed difference in OC

Herein, we propose a somewhat new definition in an OC space, which is better suited for the more detailed statistical design than the original. In comparison with the previous definition [26], the definition used here has both positive and negative differences. We expect it to give a more complete and precise distribution of the differences in OC that facilitates a statistical handling. This is expressed as follows:

$$\Delta(a, b) = \begin{cases} (a-b) - N & \text{if } b \leq \left(\dfrac{N}{2} - 1\right) \text{ and } (a-b) \geq \dfrac{N}{2} \\ (a-b) + N & \text{if } b > \left(\dfrac{N}{2} - 1\right) \text{ and } (a-b) < -\dfrac{N}{2} \\ a - b & \text{otherwise} \end{cases} \tag{5}$$

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

Fig. 4 schematically shows the above-mentioned definition of the difference, which can distinguish between clockwise and anti-clockwise orientations from a to b along the circle defining any angles by considering the cyclicity.

A similar scheme, the histogram of gradient (HOG) [29], was proposed and successfully applied in numerous researches. Compared to this scheme, OC is much simpler and has certain other merits. In the OC approach, we do not require any gradient strength information but only the orientation angles, and through the fundamental experiments conducted, we found a similar performance in terms of the robustness to fluctuations in illumination. In the calculation of the angles, whereas HOG has $[0, \pi]$, OC has a range of $[0, 2\pi]$, wherein we aim to detect any defects that are rather fine through the use of this wider resolution in combination with a more precise differentiation operator, as described in the next section.

2.3 Precise differentiation

In general, the Sobel operators of larger domains, for instance 3×3 or 5×5, give a stable estimation of the direction angle. However, in this study, we aim to detect extremely small defects during the real factory production, the manufacturing processes of which continue to advance, and the defects occurring in actual production lines continue to decrease. Therefore, to realise a better spatial sensitivity, rather than greater stability, we proposed a new gradient or differentiation operator for obtaining a higher resolution or definition.

In (6), we define two convolution matrices for the gradient operator in a 2×2 region.

$$\boldsymbol{G}_x = \begin{bmatrix} -1 & 1 \\ -2 & 2 \end{bmatrix}, \quad \boldsymbol{G}_y = \begin{bmatrix} 2 & 1 \\ -2 & -1 \end{bmatrix} \tag{6}$$

A pixel of interest in a convolution matrix is defined at the lower-left position in this 2×2 region and may have a weight of -2; the neighbouring pixels along the x and y axes, then have a weight of $+2$, whereas the neighbouring pairs of pixels have a weight of -1 or $+1$, allowing a smoothing effect for de-noising.

Fig. 5 shows three typical examples used to evaluate the sensitivity by comparing the proposed operator with a Roberts

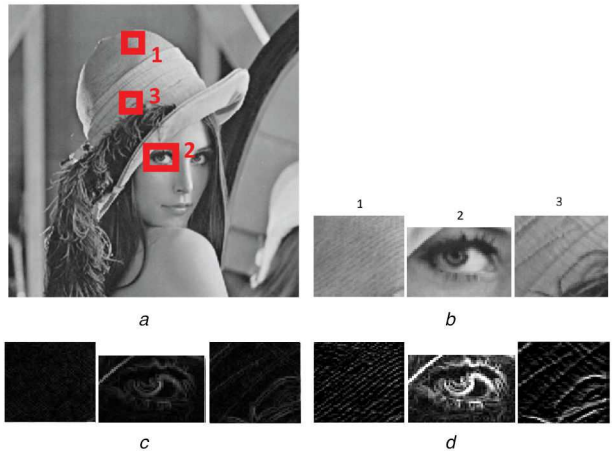

Fig. 5 *Examples of applying the proposed and Roberts operators*
(a) Lena, ***(b)*** Magnified parts, ***(c)*** Roberts operator, and, ***(d)*** Proposed operator

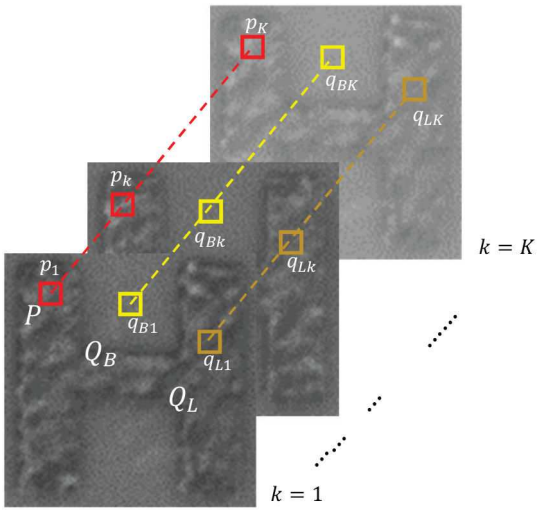

Fig. 6 *Observing and comparing OC difference histograms*

operator of 2×2 domains as a representative small operator through the use of a standard Lena image.

Comparing the results in Sections 2 and 3 in particular, we can see that the proposed operator can represent much finer structures in their corners, lids, and lashes than the Roberts operator, whereas we do not have a significant increase in the noise levels.

3 Multiple paired pixel consistency

3.1 Kurtosis-based statistical analysis for defect-free images of logotypes

Recall that our aim is to detect defects in logotypes on embossed surfaces that may behave as random textured logotypes. Due to logotypes may be printed using the same pattern, based on a fundamental assumption regarding the design of a statistical model, and despite the randomness of the surfaces, it is expected that they will have a similar texture at the same position.

As the first step towards modelling the statistical behaviour of pixels on a logotype on an embossed surface, we consider the relationship between any pair of two pixels on the logotype. In this study, the objects do not have strong colours, and thus we handle their monochrome versions here. By use of a precise operator of differentiation and the OCs mentioned above, we can expect to extract stable statistical characteristics even under fluctuating illumination conditions. Fig. 6 shows two pairs of pixels. In one pair, both pixels are on the logotype, whereas in the other pair one pixel is on the logotype and the other is on the background. Here, we aim to investigate what type of relation any pair of pixels in different locations has in the OC difference. We randomly select P on a logotype as a target pixel, an arbitrary pixel Q_L on the

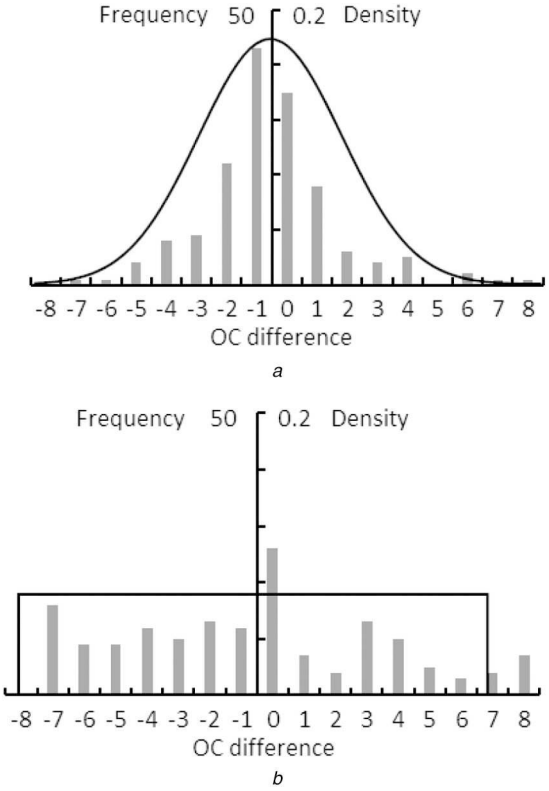

Fig. 7 *Two representative types of OC difference in a pair of pixels*
(a) Logo-logo pair (PQ_L), ***(b)*** Logo-background pair (PQ_B)

logotype part, and an arbitrary pixel Q_B in the unprinted or background part. Fig. 7 shows two typical examples of OC difference histograms made from a dataset of 160 logotype images. As an example of a typical logotype defect, such as a lack of printing, we can see that one of the logo-background pairs (P, Q_B), shown in Fig. 7*b*, has a wide variance compared with the logo-logo pairs (P, Q_L) shown in Fig. 7*a*, where the profile has a clear peak as another prominent feature in comparison with the latter; we can also see a symmetric property around the mean or centre value. From this observation and the rareness of the defects occurring in actual factories, we assume that they can basically be fitted by a single Gaussian distribution.

Their kurtoses are calculated to find their potential distributions. We found that the kurtosis of the histogram in Fig. 7*a* was 0.0065, which is close to 0, making it reasonable to fit or model using a single Gaussian distribution. The histogram can be modelled by the single Gaussian distribution

$$\frac{1}{2.3 * \sqrt{2\pi}} \exp\left(-\frac{(x - (-0.56))^2}{2 * 2.3^2}\right) \tag{7}$$

with a mean of -0.56 and a standard deviation of 2.3. By contrast, for a histogram of logo-background pairs (P, Q_B) in Fig. 7*b*, the calculated kurtosis was -1.2, which shows that it can be modelled using a uniform distribution. Fig. 7*b* can be fitted with a uniform distribution $1/(6.275 - (-8.275))$ because their mean and variance are -1 and 17.6, respectively.

To verify the universality or reproducibility of the above mentioned phenomena, for ten randomly selected target pixels P we constructed ten sets of 20 pixels Q, each of which included ten suitable pixels and another ten unsuitable pixels based on the covariance-based measure introduced, i.e. γ, defined in (13).

Fig. 8 shows the distribution of their kurtoses with respect to the γ values, resulting in two clear clusters of suitable or highly correlated pairs and unsuitable pairs of P and Q, each of which includes 100 kurtoses. The former cluster located in the upper-right area at approximately 0 shows that Gaussian distributions in the statistical models, whereas the latter cluster in the lower-left, at approximately -1.2, shows a uniform distribution.

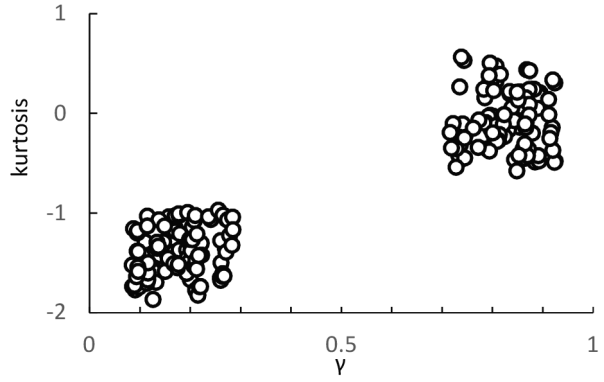

Fig. 8 *Kurtosis distribution for two types of pixel pairs*

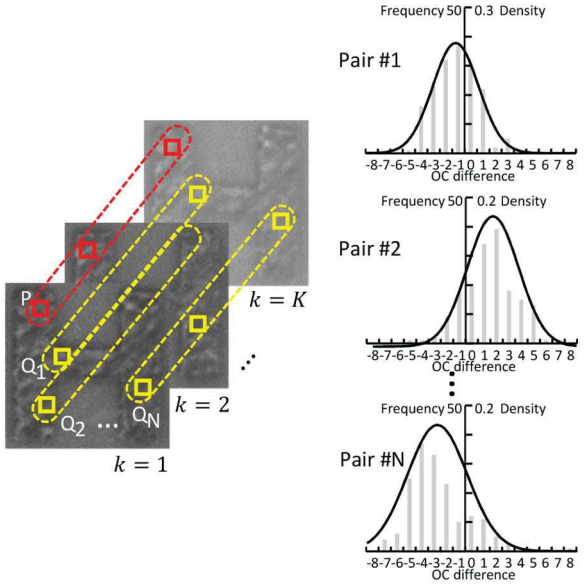

Fig. 9 *Scheme of MPPC modelling*

In these statistical analyses of the histograms, we utilise the signed difference introduced in Section 2, and in the next section demonstrate the extremely important roles of these models. Based on these observations, we ascertain the following significant differences in the behaviours of these pairs: logo-logo pixel pairs may generally have an extremely high correlation owing to the logotype printing, whereas logo-background pairs have no clear correlative relation. We expect the latter type of pair to possibly be used to represent typical defects, such as a lack of printing.

3.2 MPPC defect-free model

Fig. 9 shows a schematic structure of the proposed MPPC model, which can represent one statistical characteristic in the OC difference between two elemental pixels among the pairs. The main idea of this statistical modelling of images is called 'CP3', which was previously proposed by Liang *et al.* [27] for robust background subtraction. We propose an extension of this scheme by introducing the cohesive relationship of OCs in logo-logo pairs defined between each target pixel P on a logotype and the set of supporting pixels, also on the same logotype, which should be selected to achieve a higher consistency or correlation with the target pixel. In other words, similar trends of change as the target pixel are shown, for which we can make a statistical model by fitting a single Gaussian distribution to the OC difference histogram of these pairs with high consistency.

We now consider how to select the supporting pixels from all candidate pixels for a target pixel. For an arbitrary logo-logo pixel pair (P, Q), we have two sets of OC sequences at the same positions in all K training images as follows:

$$\boldsymbol{P} = \{p_1, p_2, \ldots, p_K\} \tag{8}$$

and

$$\boldsymbol{Q} = \{q_1, q_2, \ldots, q_K\}, \tag{9}$$

where K is the total number of training sample images, as shown in Fig. 9.

For formalisation, in this paper, we use capital letters in bold face, such as $\boldsymbol{Q}$, to represent any set, capital letters in normal font to represent any pixel, and lowercase letters to show any OCs of the pixels.

The expected values and variances over $\boldsymbol{P}$ and $\boldsymbol{Q}$ are defined as follows:

$$\bar{p} = 1/K \sum_{k=1}^{K} p_k, \quad \bar{q} = 1/K \sum_{k=1}^{K} q_k, \tag{10}$$

$$\sigma_{\boldsymbol{P}}^2 = \frac{1}{K}\sum_{k=1}^{K}(p_k - \bar{p})^2, \quad \sigma_{\boldsymbol{Q}}^2 = \frac{1}{K}\sum_{k=1}^{K}(q_k - \bar{q})^2. \tag{11}$$

The covariance between $\boldsymbol{P}$ and $\boldsymbol{Q}$ is defined as follows:

$$C_{\boldsymbol{P},\boldsymbol{Q}} = \frac{1}{K}\sum_{k=1}^{K}(p_k - \bar{p})(q_k - \bar{q}) \tag{12}$$

If $C_{\boldsymbol{P},\boldsymbol{Q}} > 0$, a consistency or co-occurrence probability occurs, and to measure the consistency quantitatively, we use the following Pearson's product-moment correlation coefficient

$$\gamma_{\boldsymbol{P},\boldsymbol{Q}} = \frac{C_{\boldsymbol{P},\boldsymbol{Q}}}{\sigma_{\boldsymbol{P}} \cdot \sigma_{\boldsymbol{Q}}} \tag{13}$$

where $\sigma_{\boldsymbol{P}}$ and $\sigma_{\boldsymbol{Q}}$ are the standard deviations of $\boldsymbol{P}$ and $\boldsymbol{Q}$, respectively.

For all pixels $P(u, v)$ at the position (u, v) over the full regions covering the logotype, we may have $M - 1$ candidate pixels in the same logotype, where M properly defines the total size of the logotype in pixels. The position of P does not vary for every edge pixel, but for all pixels in the logotype. From these candidates, we can select $N(< M)$ supporting pixels in descending order of the value of $\gamma_{\boldsymbol{P},\boldsymbol{Q}}$. The set of N supporting pixels is as follows:

$$\mathbb{Q} = \left\{ Q_i(u_i, v_i) \middle| \gamma_{\boldsymbol{P},\boldsymbol{Q}_i} \geq \gamma_{\boldsymbol{P},\boldsymbol{Q}_{i+1}} \right\}_{i=1,2,\ldots,N} \tag{14}$$

We assume that each supporting pixel Q_i maintains a bivariate OC difference with the target pixel P as follows:

$$\Delta(p, q_i) \sim N(\mu_i, \sigma_i) \tag{15}$$

where $N(\mu_i, \sigma_i)$ is the Gaussian distribution with a mean μ_i and variance σ_i^2, which are calculated from the corresponding pixel sets $\boldsymbol{P}$ and $\boldsymbol{Q}$.

After modelling for one target pixel P, the above set of N pairs of four parameters for the position u_i, v_i and the two statistical parameters μ_i and σ_i are recorded in a row of the look-up table (LUT). Through repetitive modelling, the LUT is filled in to include the total set of MPPC models for all pixels in an elemental logotype.

4 Defect detection by MPPC

We now discuss how to utilise the proposed MPPC model of the relationship between pixel pairs in a defect-free logotype for detecting many types of logotype defects. Due to the MPPC model can represent the statistical behaviour of the relation of an individual target pixel to the supporting pixels around it, we utilise it to statistically test whether a target pixel is recognised as a reasonable sample from the distribution registered in the LUT. The scheme for the defect detection algorithm applied is shown in Fig. 10.

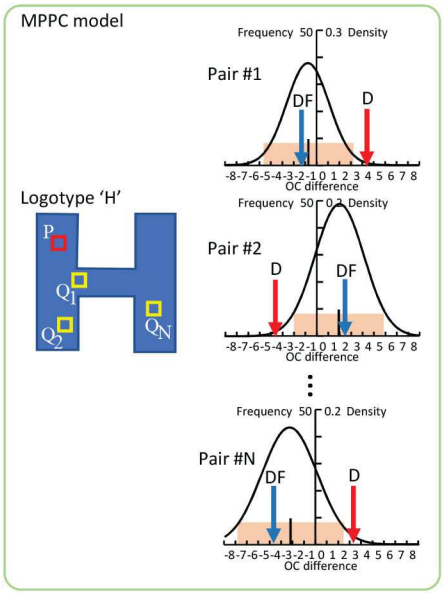

Fig. 10 *Scheme of defect detection when using the MPPC model*

Input: Test image J, set thresholds C and T
Output: Defective/Defect-free
for Each logotype pixel P in J **do**
  Initialization:
  Load LUT $[u_i^{'}, v_i^{'}, \hat{\sigma}_i, \hat{\mu}_i]$ of $\left\{Q_i^P\right\}_{i=1,2,\ldots,N}$
  Pixel pair identification:
  for i = 1,2,...,N **do**
    if $|\Delta(p, q_i) - \hat{\mu}_i| > C * \hat{\sigma}_i$ **then**
      $\beta\left(Q_i^P\right) = 1$
    else
      $\beta\left(Q_i^P\right) = 0$
  Defect decision:
  Compute the mismatched ratio.
  $\xi(P) = \frac{1}{N}\sum_{i=1}^{N}\beta\left(Q_i^P\right)$
  if $\xi(P) > T, (0 < T < 1)$ **then**
    $P \rightarrow$ *defective*
  else
    $P \rightarrow$ *defect-free*

Fig. 11 *Algorithm 1: Defect detection*

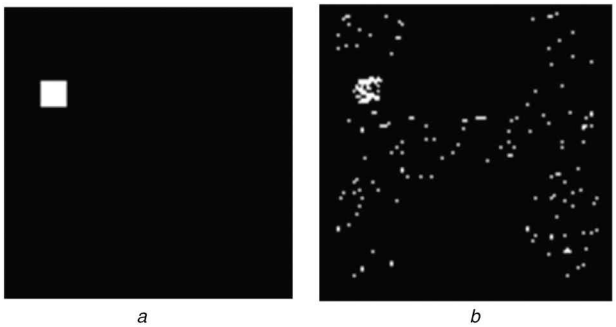

Fig. 12 *Example of detection result*
(a) Ground truth image and, ***(b)*** Detection result

There are several types of defects, including dust or particles, scratches, misprints, hairs, and spit, the characteristics of which include a random alignment, texture, shape, or size. Due to the high performance quality control in production lines, these defects may be extremely small, and furthermore, we should also keep in mind their very low probability of occurrence. In addition, in this study, we also need to deal with the randomness in embossed surfaces. Thus, these fundamental characteristics lead us to approaches based on a pixel-wise evaluation for detection. Afterwards, we may proceed with some following steps to recognise aggregated regions that reveal some common characteristics. From these considerations and the investigation described in Section 3, the two features of our MPPC model, spatial sparseness and high consistency in a correlative relation, can be utilised to handle such defects. The former may prevent any defects from occupying P and some supporting pixels simultaneously, and through the use of the latter feature we then expect to have an evaluation scheme to recognise whether P is occupied with any defects.

The next task must be to design a measure for judging defective pixels or defect-free pixels using the MPPC model. We first test each OC difference between the target pixel P and a supporting pixel Q in the MPPC model or the LUT by applying $\mathcal{Q}$ for P in the target image. We define the following measure:

$$\beta_i = \begin{cases} 1 & \| \Delta(p, q_i) - \mu_i \| \geq C \cdot \sigma_i \\ 0 & \text{otherwise} \end{cases}, \tag{16}$$

to identify the normal ($\beta_i = 0$) or abnormal ($\beta_i = 1$) states at the corresponding position defined by the elemental MPPC model, where the constant C is a parameter that can be set from 1 to 3 to define an area for an acceptance probability of 68–99.7%. Finally, we use the total sum

$$\xi = \frac{1}{N}\sum_{i=1}^{N}\beta_i, \tag{17}$$

to construct a decision rule for the occupation of P by any particular defect, $\xi \geq T$, where $T = (1/N)(\text{floor}(N/2) + 1)$ is a threshold for the general majority rule, and N is the total number of supporting pixels. Pseudo-codes for defect detection are shown in Algorithm 1 (see Fig. 11).

4.1 Post-filtering

Fig. 12 shows an example of a detection result, where the white and black dots represent detected defective pixels and defect-free pixels, respectively. To check the fundamental performance of the proposed detection algorithm, we used a synthesised image in which the real logotype 'H' was embedded by copies from a background region of the same shape and size as the white square in the ground truth image. Numerous over detections can be seen in the background and numerous under detections can be found in the defect region. An over detection indicates that the pixels have pseudo-defective characteristics but are defect-free in reality, whereas an under detection of pixels means the exact opposite. 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 proposed algorithm described in the previous chapter. In general, any defect must be a closed, continuous, and solid area. However, our method classifies each pixel independently. Therefore, there is a gap between the pixel-based MPPC and the region-based defect. To overcome this problem, we designed the following simple morphological filter.

Fig. 13 outlines the definitions of true positives (TPs), false positive (FPs), true negatives (TNs), and false negatives (FNs) in the defect detection.

Here, we introduce the idea of connected component labelling [30], in which any connected region of the size less of than a threshold T_h is removed as FPs, and T_h is a threshold that can be adjusted to achieve the desired result. We utilised 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 utilise the closing morphological operator [31], which is defined as follows:

$$Y \cdot M = (Y \oplus M) \ominus M, \tag{18}$$

where Y is any binary image from MPPC; $\oplus$ and $\ominus$ are dilation and erosion operators, respectively; and M is a mask matrix of with a size of 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, at the resolution of the training data set, we observed that defects with a size of less than 5×5 are generally not easily noticeable by customers.

5 Experiments

5.1 Specifications

Fig. 14 shows portions of real defect images for the experiments conducted in this study, which were collected in factories. All defect images have the corresponding ground truth images, and the defect-free area is represented as black and the defective area as white. We utilise the six sets of real images from the factory, which consists of the six characters 'H', 'U', 'A', 'W', 'E', and 'I', each of which includes 160 defect-free images that cover five different illumination conditions: darker, dark, normal, bright, and brighter. We chose 60 defect-free images taken under the five different illumination conditions to train the MPPC model. Through the experiments, we set the two thresholds of the detection stage as $C = 2.0$ and $T = 0.5$, respectively.

As the experimental images, we utilised two types of defects: real and synthesised or artificial defects, as shown in Figs. 15 *a* and *b*, respectively. Fig. 15*a* shows small dust attached to the logotype 'H', which we utilised as a representative to conduct the defect detection experiments under different illumination conditions. Here, 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' were used, all of which contain real defects.

Owing to a slight difficulty in collecting real defects from factories, we utilised some synthesised defects during the experiments, as shown in Fig. 15*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 a misprinting of the logotype. As shown in Fig. 15*a*, one can understand to define any ground truth for the real defects owing to the uncertainty of the boundary despite a magnification. Therefore, we use synthesised defects to evaluate the performance of the proposed MPPC-based detection quantitatively. The image data used in all experiments conducted in this paper can be downloaded from [32].

Basically, because the proposed method does not consider the translation or rotational changes in the target images, any original data image should be aligned in the same position and should not have any individual rotation. In a real factory, it is not too difficult to realise this requirement because of the quality control. However, in this study, we realised it through three inspection procedures. The first one is an inspection by operators in the factory where all data were corrected using a specialised fixation tool to give them a constant position and orientation relative to the camera. Next, we checked their rotational constancy. Finally, all elemental logotype images corresponding to each character were corrected regarding their position through a registration with a normalised cross correlation.

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

5.2 Evaluation indicators

There are several ways to evaluate the performance of defect detection. First, pixel-level precision, recall, and F-measure are applied to test the proposed MPPC models and the detection algorithm for the statistical test based on the models, where our problem is 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 of detecting defects at a pixel level, we utilised three evaluation metrics: *Precision* (also known as a positive predictive value), *Recall* (also known as sensitivity), and $F-measure$. These are widely applied in pattern recognition, information retrieval, and binary classification. In addition, pixel-level defect detection is a typical binary classification problem, and thus these three indicators can also be used for a quantitative analysis of the defect detection. Here, *Precision* can be considered

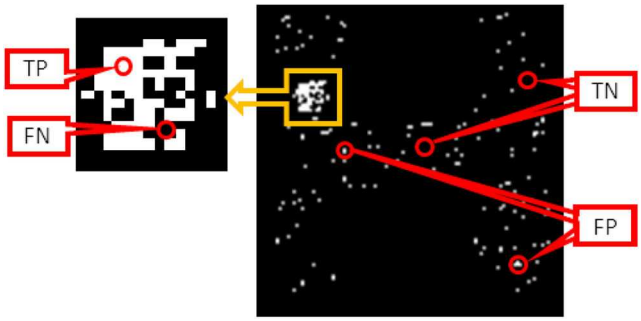

Fig. 13 *Definitions of TPs, FPs, TNs, and FNs in defect detection*

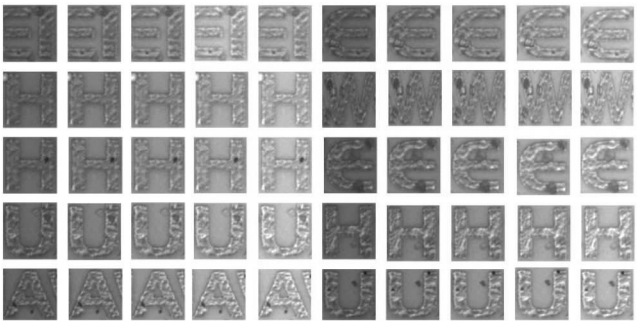

Fig. 14 *Some examples of real defect images*

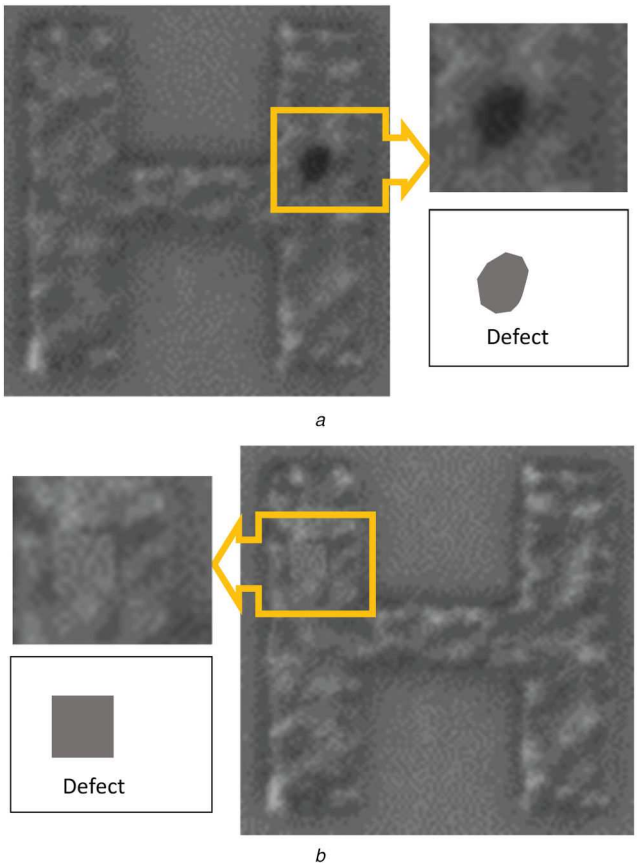

Fig. 15 *Two types of experimental images*
(a) Real and, ***(b)*** Synthesised defects

a measure of the accuracy, whereas *Recall* can be considered a measure of the defect integrity.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{19}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{20}$$

where $F-$measure is a harmonic average of the *Precision* and *Recall*.

$$F - measure = \frac{2\ \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{21}$$

For an image-based performance evaluation [33], we utilised the detection rate and false alarm rate as follows:

$$\text{Detection Rate} = \frac{\text{N}_\text{T}}{\text{N}_\text{TD}} \tag{22}$$

$$\text{False Alarm Rate} = \frac{\text{N}_\text{F}}{\text{N}_\text{TF}} \tag{23}$$

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

5.3 Experiments

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

- 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 some dust is 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. Due to 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.

5.3.1 Different defect sizes: We want to test how well small defects can be detected by our algorithm. The illumination conditions of the images used in this experiment are the same, although the pixel sizes of the synthesised defects added in the four types are changed to 10×10, 7×7, 5×5, and 3×3 pixels. Fig. 16 shows some examples of experimental results for different sized defects.

Table 1 shows the performance with respect to the change in size. After applying the filtering algorithm mentioned in Section 4.1, most of the results show that the performance improved substantially. Table 2 represents the image-based performance evaluation with 400 synthesised images including the four sizes of defects. We can see that when the defect size is greater than or equal to a pixel resolution of 5×5, the detection rate can be maintained at a high level (~95% on average). However, defects with a pixel resolution of 3×3 are slightly too small, and our algorithm is likely to miss a detection. Therefore, we believe that our algorithm can detect defects of greater than or equal to pixel resolutions of 5×5, and ensure a high detection rate, whereas we may need a more magnified observation if we want to detect much smaller defects.

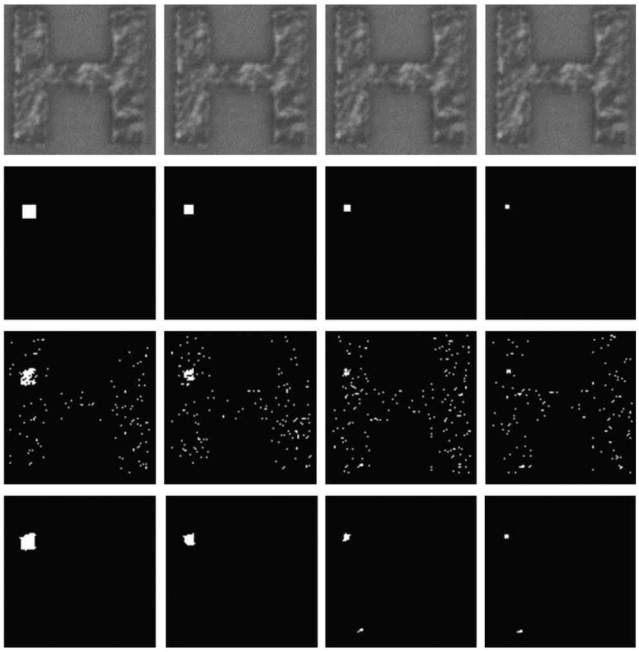

Fig. 16 *Some examples of defect detection with different defect sizes. From the top row, the test images, their ground truth images, the detected results, and the filtered results. From the leftmost column, defects with pixel sizes of* 10×10, 7×7, 5×5, *and* 3×3

Table 1 Pixel-level based performance evaluation of defect detection under different sized defects

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 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

5.3.2 Fluctuations of illumination: Precise control of the actual factory illumination conditions is not easy. Furthermore, it is common in factories to have plural sets or stages of visual inspection, which may undergo changes in the conditions over time. The following factors can cause a variation in the illumination conditions.

- The illumination conditions will be affected by natural light.
- The life of the light sources is limited. When we replace them, we cannot guarantee that 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, an alternating current is used, which causes the lighting to fluctuate. In addition, as the production lines run extremely quickly, the shutter speed of the camera used for detection is very high. This will result in a noticeable change in the illumination of the captured photo.

Therefore, the proposed algorithm should be robust to fluctuations in the illumination, for which we have introduced OCs to build the MPPC model. Here, we utilise the images of the same

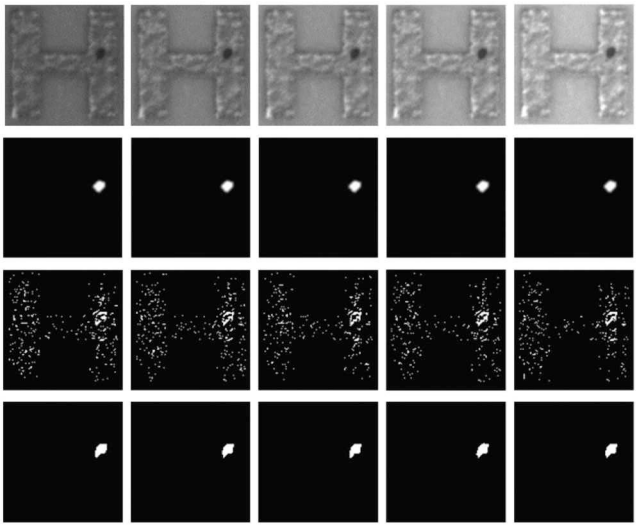

Fig. 17 *Some examples of defect detection under different illumination conditions. From the first row: the test images, their ground truth images, the detection results, and the filtered results. The left columns, images from darker to brighter illumination conditions*

Table 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 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

objects under five different illumination conditions for the experiments, as shown in Fig. 17, which were collected from real factories under the consideration of real illumination conditions.

There are 50 real defect images and 100 defect-free images for each character.

Fig. 17 shows some of the detection results. Despite a severe fluctuation in illumination, we found that the proposed method can detect real defects similarly in size and shape from this figure, which shows the strong robustness of the OCs in the MPPC models.

Table 3 shows the pixel-level performance evaluation, whereas Table 4 shows the image-based performance evaluation. From Table 3, we can see that under different illumination conditions, the performance of the defect detection is extremely high and achieves similar results. A better performance of such automatic inspection systems under changes in illumination is necessary for quality control in numerous production lines, including those with an international distribution.

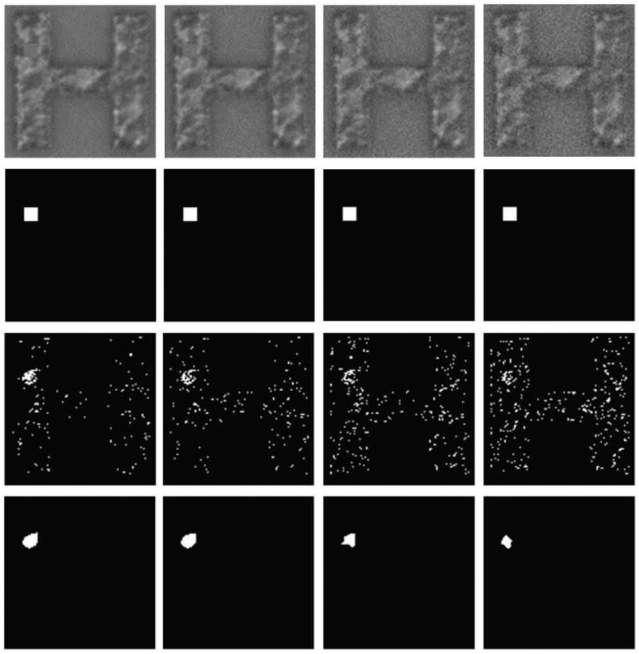

Fig. 18 *Some examples of defect detection with different amounts of additional noise. From the first row: the test images, their ground truth images, the detected results, and the filtered results. From the left column: no additional noise, and versions with σ, 2σ, and 3σ intensity noises added*

Table 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

5.3.3 Different noise intensities: In this section, we verify whether our algorithm is robust to a certain level of noise. Here, it can be seen that it is not easy to gather various noisy images, and thus we utilise four types of synthesised images with different amounts of additional noise, among which Gaussian noise was selected. Due to we have many observed images from real factories, we can estimate the noise level from such data, and we found in the background part that the standard deviation of the additional noise is approximately $\sigma = 3$ at a grey-scale with 256 levels. We utilise 100 synthetic defect images and 100 defect-free images for each character. To create these defect-free images, we utilise the remaining 100 defect-free images from the previous training to conduct defect detection experiments for each character. In addition, the synthetic defect images were generated based on these 100 defect-free images.

Fig. 18 shows some examples of defect detection results for different noise intensities, where it can be seen that the performance gradually worsened along with an increase in the noise intensity; however, up to 3σ, the proposed method could detect the true position despite the additional noise.

From Tables 5 and 6, we can see that when the intensity of the additional noise in the image is within approximately 2σ, the defect detection did not deteriorate. However, in the case of 3σ, we found a performance degradation in these experiments.

5.3.4 Different characters: In this section, we consider the performance of defect detection for different printed characters. We used four cases of real defects in different positions for 'H,' three

Table 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

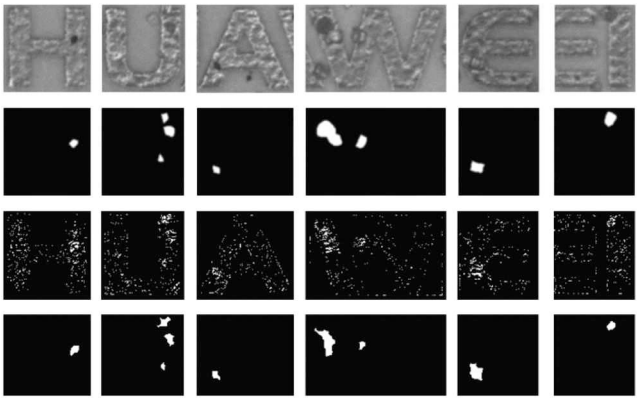

Fig. 19 *Some examples of defect detection results for different characters. From the first row: test images, their ground truth images, the detected results, and the filtered results*

Table 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 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

for 'U,' four for 'A,' one for 'W,' four for 'E,' and three for 'I,' respectively. For the logotype 'I,' we need an additive explanation because the size is relatively small. On the one hand, we cannot find an adequate number of stable supporting pixels, which will impact the effectiveness of our algorithm, which works in a pixel-wise manner independent of each character. On the other hand, the small size may cause the selected supporting pixels to become too clustered, and thus the supporting pixels are easily affected by each other. Therefore, we extend the area to include a part of the neighbouring 'E,' as shown in Fig. 19. This figure shows some examples of defect detection results for each character, where in the background parts in 'U', 'A,' and 'W,' for instance, we have additive noises, which are not handled in this particular paper. We can see that the proposed method can locate most parts of the defect area. After filtering, the over-detection noise is mostly removed and some holes are filled in.

In this paper, we have concentrated on the logotype area and excluded the background area (unprinted area). However, in Fig. 19, the third, fourth, and last columns contain some defects in the background area. To solve these problems, we will extend our algorithm to include defect detection in the background area.

Table 7 shows a pixel-level based performance evaluation, whereas Table 8 shows the image-based performance evaluation.

After filtering, the recall and precision improved substantially, and the F-measure reached ∼0.85. This shows that our method has achieved good results. For the image-based evaluation, all images with defects were successfully identified, demonstrating the stability of our method.

5.3.5 Comparison experiment: Numerous researchers carrying out defect detection for surface inspection have commonly examined steel [34], textile [35, 36], and wood [37, 38]. For printing inspections, examinations have been generally performed using paper materials and pharmaceutical capsules [1]. However, to the best of our knowledge, no studies have investigated defect detection for logotypes on 3D micro-textured surfaces, as examined in this study. To verify the effectiveness of the proposed method, we compared it with a phase only transform (PHOT) for surface defect detection [39], which is based on a particular transform for the removal of any regularities of arbitrary scales from the target images and for preserving only irregular patterns, such as defects. In addition to its high effectiveness in detecting defects on irregularly textured surfaces, the objects they deal with are somewhat comparable to the materials mentioned in this paper. Furthermore, the authors disclosed their source codes and the settings of the parameters.

Herein, we utilise 50 synthesised defect images and 20 real defect images for comparison experiments. The implementation was conducted using the source code disclosed, and the parameters selected were those recommended by the respective authors.

Figs. 20 and 21 show some representative results for the synthesised and real defects, respectively.

As we can see from Figs. 20 and 21, around half of the defects were detected by PHOT, which is roughly consistent with the ground truth, whereas the proposed method could detect all of the defects. For high-contrast defects, PHOT could obtain a rather good detection performance. However, for low-contrast defects, it did not achieve a good performance. In addition, from Fig. 21, we can see that the defect detection performance of our method was not dependent on the positions and was relatively high. Table 9 shows the pixel-level based performance evaluation, whereas Table 10 shows the image-based performance evaluation. From them, we could see that PHOT can successfully detect ∼70% of the defects, whereas the proposed method can successfully detect all defects.

5.4 Computation cost

Owing to our algorithm was designed to be applied in actual production lines, its computational efficiency must meet certain requirements. In this section, we consider the computational cost of our proposed method. We will discuss two aspects, one based on the image level, and the other based on the pixel level. For this, we used a PC with a 3.5 GHz Intel Xeon CPU and 64 GB of RAM running Windows 10.

The proposed algorithm consists of two parts: a training stage and a detection stage. First, when we consider the computational time during the training stage, 100 defect-free images with a pixel size of 110×111 pixels were used to make defect-free MPPC models. We created two versions of the programme, one in MATLAB and the other in C++. The development environments were MATLAB 2016a and Visual Studio 2013 along with the Opencv2.4.11 library.

In the MATLAB version, we use the timing function in MATLAB. In the C++ version, we used the clock timing function. In the MATLAB version, the entire training process took 9548.69 s,

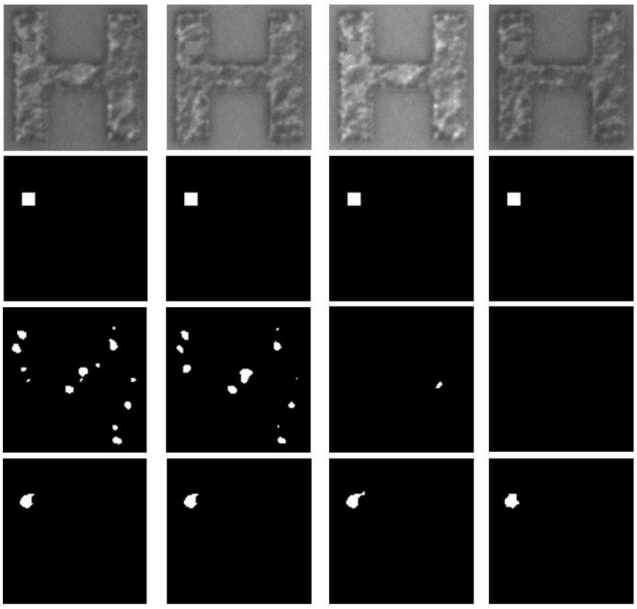

Fig. 20 *Some examples of detection results for synthesised defects. From the first row: test images, their ground truth images, and the detected results by PHOT and the proposed method, respectively*

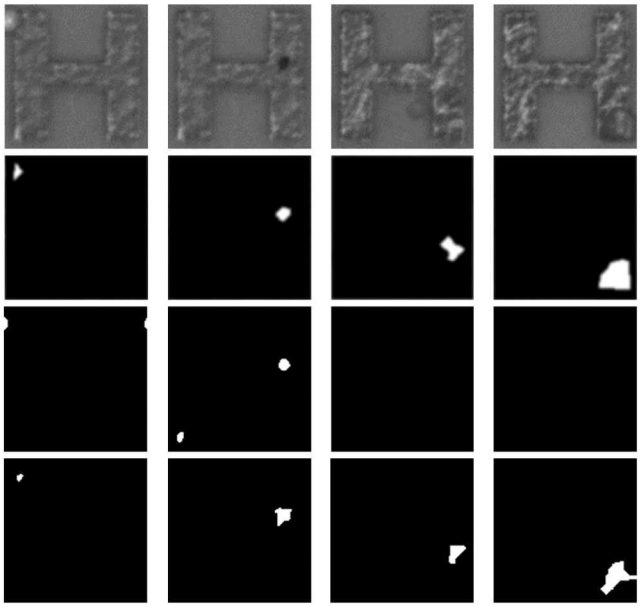

Fig. 21 *Some detected defects in the real defect images*

whereas in C++, it took 111.06 s. For each pixel, the training stage took 0.78 s in MATLAB, and 9.09 ms in C++.

Now, we consider the computational cost in the detection stage for the same sized images. In the MATLAB version, the computational time for each image was 19.58 ms.

Table 11 shows the computational time in the C++ version. Due to the computational time is extremely short, the timing is susceptible to certain aspects, for example swapping. We measured an average computational time of 0.945 ms from eight trials using the same image, i.e. 1.6 μs per pixel in MATLAB and 0.0773 μs per pixel in C++, respectively.

6 Conclusions

After some preparations, such as a precise differentiation scheme and the signed difference in the OCs, we analysed some statistical behaviours of logotypes printed on the embossed surfaces that have their own non-uniformity, where the kurtosis analysis was effective. In addition, we proposed a novel model of statistical similarity called multiple paired pixel consistency, or MPPC, using OCs in defect-free logotypes printed on the embossed surface.

Table 9 Pixel-level based performance of defect detection for comparison

	Defect type	Recall	Precision	*F*-measure
PHOT	synthesised	0.49	0.49	0.49
	real	0.53	0.87	0.66
proposed	synthesised	0.89	0.96	0.91
	real	0.83	0.97	0.90

Table 10 Image-level based performance of defect detection for comparing

	Defect type	Detection rate, %
PHOT	synthesis	66
	real	50
proposed	synthesis	98
	real	100

Table 11 Computation time in the detecting stage

NO	computation cost, ms
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

Based on MPPC models of defect-free images, we proposed a new defect localisation algorithm, which is effective in the detection of both synthesised and real defect images at both the pixel and image levels.

For variations in size, illumination, noise, and characters applied, we confirmed the performance of the proposed approach, and the results of numerous experimental tests demonstrated the effectiveness of the method for a precise defect localisation. Our method achieves better results than some other state-of-the-art defect detection methods. The recall, precision, and *F*-measure for most of the results reached 0.85, 0.93, and 0.9, respectively. In addition, the defect detection rate reached almost 100%. Furthermore, the computational efficiency of our method is high. The computational time for each image with a pixel resolution of 110×111 was 0.945 ms. This shows that our method can meet the needs of actual industrial production in terms of accuracy and computational consumption.

In future studies, if it becomes possible to conduct a detailed analysis of the statistical characteristics in logotypes on the embossed surfaces, we hope to design a schema to identify defect types in several classes, which may contribute to increasing the effectiveness of the quality control of a production line.

7 References

[1] Mehle, A., Bukovec, M., Likar, B., *et al.*: 'Print registration for automated visual inspection of transparent pharmaceutical capsules', *Mach. Vis. Appl.*, 2016, **27**, (7), pp. 1087–1102

[2] Xie, X.: 'A review of recent advances in surface defect detection using texture analysis techniques', *ELCVIA Electron. Lett. Comput. Vis. Image Anal.*, 2008, **7**, (3), pp. 114–117

[3] Hoseini, E., Farhadi, F., Tajeripour, F.: 'Fabric defect detection using auto-correlation function', *Int. J. Comput. Theory Eng.*, 2013, **5**, (1), p. 114

[4] Narendra, V., Hareesha, K.: 'Study and comparison of various image edge detection tehniques used in quality inspection and evaluation of agricultural and food products by computer vision', *Int. J. Agric. Biol. Eng.*, 2011, **4**, (2), pp. 83–90

[5] Taştimur, C., Karaköse, M., Akın, E., *et al.*: 'Rail defect detection with real time image processing technique'. Industrial Informatics (INDIN) 2016 IEEE 14th Int. Conf. on, Poitiers, France 2016, pp. 411–415

[6] Fathi, A., Monadjemi, A.H., Mahmoudi, F.: 'Defect detection of tiles with combined undecimated wavelet transform and glcm features', *Int. J. Soft. Comput. Eng.*, 2012, **2**, (2), pp. 2231–2307

[7] Karlekar, V.V., Biradar, M., Bhangale, K.: 'Fabric defect detection using wavelet filter'. Computing Communication Control and Automation (ICCUBEA) 2015 Int. Conf. on, Maharashtra, India, 2015, pp. 712–715

[8] Yu, L., Qi, D.: 'Applying multifractal spectrum combined with fractal discrete brownian motion model to wood defects recognition', *Wood Sci. Technol.*, 2011, **45**, (3), pp. 511–519

[9] Wakaf, Z., Jalab, H.A.: 'Defect detection based on extreme edge of defective region histogram', *J. King Saud University-Comput. Inf. Sci.*, 2018, **30**, (1), pp. 33–40

[10] Latif-Amet, A., Ertüzün, A., Erçil, A.: 'An efficient method for texture defect detection: sub-band domain co-occurrence matrices', *Image Vis. Comput.*, 2000, **18**, (6-7), pp. 543–553

[11] Zhu, D., Pan, R., Gao, W., *et al.*: 'Yarn-dyed fabric defect detection based on autocorrelation function and glcm', *Autex Res. J.*, 2015, **15**, (3), pp. 226–232

[12] Fekri-Ershad, S., Tajeripour, F.: 'Multi-resolution and noise-resistant surface defect detection approach using new version of local binary patterns', *Appl. Artif. Intell.*, 2017, **31**, (5-6), pp. 395–410

[13] Vilnrotter, F.M., Nevatia, R., Price, K.E.: 'Structural analysis of natural textures', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1986, **8**, (1), pp. 76–89

[14] Chen, J., Jain, A.K.: 'A structural approach to identify defects in textured images'. Systems Man and Cybernetics 1988 Proc. of the 1988 IEEE Int. Conf. on, Beijing, People's Republic of China, 1988, vol. 1, pp. 29–32

[15] Bennamoun, M., Bodnarova, A.: 'Automatic visual inspection and flaw detection in textile materials: past, present and future'. SMC'98 Conf. Proc. 1998 IEEE Int. Conf. on Systems Man and Cybernetics (Cat. No. 98CH36218), San Diego, CA, USA, 1998, vol. 5, pp. 4340–4343

[16] Bodnarova, A., Bennamoun, M., Kubik, K.: 'Defect detection in textile materials based on aspects of the hvs'. SMC'98 Conf. Proc. 1998 IEEE Int. Conf. on Systems Man and Cybernetics (Cat. No. 98CH36218), San Diego, CA, USA, 1998, vol. 5, pp. 4423–4428

[17] Sakhare, K., Kulkarni, A., Kumbhakarn, M., *et al.*: 'Spectral and spatial domain approach for fabric defect detection and classification'. Int. Conf. on Industrial Instrumentation and Control (ICIC) 2015, Pune, India, 2015, pp. 640–644

[18] Jing, J., Chen, S., Li, P.: 'Automatic defect detection of patterned fabric via combining the optimal gabor filter and golden image subtraction', *J. Fiber Bioeng. Inf.*, 2015, **8**, (2), pp. 229–239

[19] Wen, Z., Cao, J., Liu, X., *et al.*: 'Fabric defects detection using adaptive wavelets', *Int. J. Cloth. Sci. Technol.*, 2014, **26**, (3), pp. 202–211

[20] Xiaobo, Y.: 'Fabric defect detection of statistic aberration feature based on gmrf model [j]', *J. Text. Res.*, 2013, **4**, p. 026

[21] Li, M., Cui, S., Xie, Z., *et al.*: 'Application of gaussian mixture model on defect detection of print fabric [j]', *J. Text. Res.*, 2015, **8**, p. 019

[22] Tao, X., Zhang, D., Ma, W., *et al.*: 'Automatic metallic surface defect detection and recognition with convolutional neural networks', *Appl. Sci.*, 2018, **8**, (9), p. 1575

[23] Li, Y., Zhang, D., Lee, D.J.: 'Automatic fabric defect detection with a wide-and-compact network', *Neurocomputing*, 2019, **329**, pp. 329–338

[24] Krummenacher, G., Ong, C.S., Koller, S., *et al.*: 'Wheel defect detection with machine learning', *IEEE Trans. Intell. Transp. Syst.*, 2017, **19**, (4), pp. 1176–1187

[25] Yan, Y., Kaneko, S., Asano, H.: 'Accumulated and aggregated shifting of intensity for defect detection on micro 3d textured surfaces', *Pattern Recognit.*, 2020, **98**, p. 107057

[26] Ullah, F., Kaneko, S.: 'Using orientation codes for rotation-invariant template matching', *Pattern Recognit.*, 2004, **37**, (2), pp. 201–209

[27] Liang, D., Hashimoto, M., Iwata, K., *et al.*: 'Co-occurrence probability-based pixel pairs background model for robust object detection in dynamic scenes', *Pattern Recognit.*, 2015, **48**, (4), pp. 1374–1390

[28] Xiang, S., Yan, Y., Asano, H., *et al.*: 'Robust printing defect detection on 3d textured surfaces by multiple paired pixel consistency of orientation codes'. 2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics, Mie, Japan, 2018, pp. 373–378

[29] Dalal, N., Triggs, B.: 'Histograms of oriented gradients for human detection', 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886–893

[30] He, L., Chao, Y., Suzuki, K., *et al.*: 'Fast connected-component labeling', *Pattern Recognit.*, 2009, **42**, (9), pp. 1977–1987

[31] Bouaynaya, N., Schonfeld, D.: 'Spatially variant morphological image processing: theory and applications'. Visual Communications and Image Processing 2006, San Jose, CA, USA, 2006, vol. 6077, p. 60771Y

[32] Xiang, , 2019, Available at https://hce-lab.net/cloud/index.php/s/3FFFFmU8JIL1RTO

[33] Ngan, H.Y., Pang, G.K., Yung, N.H.: 'Automated fabric defect detection- a review', *Image Vis. Comput.*, 2011, **29**, (7), pp. 442–458

[34] Zhou, S., Chen, Y., Zhang, D., *et al.*: 'Classification of surface defects on steel sheet using convolutional neural networks', *Mater. Technol.*, 2017, **51**, (1), pp. 123–131

[35] Li, P., Liang, J., Shen, X., *et al.*: 'Textile fabric defect detection based on low-rank representation', *Multimedia Tools Appl.*, 2017, **78**, (1), pp. 1–26

[36] Lizarraga-Morales, R.A., Correa-Tome, F.E., Sanchez-Yanez, R.E., *et al.*: 'On the use of binary features in a rule-based approach for defect detection on patterned textiles', *IEEE Access*, 2019, **7**, pp. 18042–18049

[37] Zhang, Y., Xu, C., Li, C., *et al.*: 'Wood defect detection method with pca feature fusion and compressed sensing', *J. Forestry Res.*, 2015, **26**, (3), pp. 745–751

[38] Ke, Z.N., Zhao, Q.J., Huang, C.H., *et al.*: 'Detection of wood surface defects based on particle swarm-genetic hybrid algorithm'. Audio Language and Image Processing (ICALIP) 2016 Int. Conf. on, Shanghai, People's Republic of China, 2016, pp. 375–379

[39] Aiger, D., Talbot, H.: 'The phase only transform for unsupervised surface defect detection'. 2010 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pp. 295–302