

Research Article

Defect Contour Detection of Complex Structural Chips

Bin Lin ¹, Jie Wang,¹ Xia Yang,¹ Zhangdong Tang,² Xuan Li,² Cenlin Duan,²
and Xiaohu Zhang ¹

¹Sun Yat-sen University, Guangzhou 510725, China

²China Aerospace Components Engineering Center, Beijing, China

Correspondence should be addressed to Xiaohu Zhang; zhangxiaohu@mail.sysu.edu.cn

Received 21 January 2021; Revised 13 May 2021; Accepted 19 May 2021; Published 28 May 2021

Academic Editor: Jude Hemanth

Copyright © 2021 Bin Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the manufacture of chips, it is important to detect defects to assess whether the chip is potentially damageable that could cause unnecessary cost. Most assessment rules are set in light of characteristics determined by defect contours, such as area and range. However, conventional image process methods seldom show a satisfactory performance on chips with complex structures because they are difficult to distinguish defect contours from edges of structures. To solve this issue, this study proposes a method based on region segmentation search. The positions of structures in the image are calculated by edge matching to obtain the number of structure layers in each pixel. Regions whose pixels have the same number are divided into subregions which are coded by the two-pass algorithm. The edges in each subregion are then extracted by the Canny operator to construct edge information of the whole image. Interpolation is used to correct incomplete defect edges according to their endpoints. The remaining interference contours are eliminated on the basis of their shapes. A study of a certain kind of chips is presented. Different illumination situations were simulated to verify the robustness of the proposed method. Most bubbles in the images were detected successfully with their contours coded accurately. Because of this, more than 92% of assessment results of chips were identical to the ones in reality engineering, which proves that the method proposed by this study can efficiently detect the defect contours and improve the ability obviously relative to the current approaches.

1. Introduction

With the rapid advancement of chip manufacturing, structures of chips are in the direction of increasing complexity and elaboration. Therefore, automated visual inspection (AVI) has received great attention in chip detection due to its nondestructiveness and reliability [1]. Among all the characteristics of defects, edge detection is one of the most important aspects because it is the basis of many quality assessments. However, the edges of defects are blurred by noise in the image. Moreover, there is no definite criterion to distinguish the edges of defects from those of structures in the image, which results in false detection.

In recent decades, many traditional approaches such as projection and filtering have been applied in the chip inspection field. Teoh et al. [2] developed five separate algorithms to deal with, namely, missing components, misalignment, wrong orientation of IC chips, wrong parts, and poor solder joints on surface mount PCBs. Gallegos et al.

[3] detected the presence or absence of surface-mounted devices (SMDs) in the inspection of printed circuit boards based on linear projection and other processing. Lin [4] used wavelet characteristic to inspect ripple defects in the surface barrier layer (SBL) chips of ceramic capacitors and used discrete cosine transform (DCT) to restore image for inspection of tiny pinhole defects in randomly textured surfaces of surface barrier layer (SBL) chips. Wu et al. [5] proposed a two-stage inspection process which analyzed the features of the components under three colors (red, green, and blue) structure light source for surface mounted components after they have been placed in wet solder paste on a printed circuit board (PCB). And the next year (2010), they gave out another way which combined an adaptive segmentation method, the sliding location window algorithm, and the Bayes classifier to solve this problem. Wu et al. [6] calculated the geometrical parameters of the chip components by means of vision-based methods such as the maximum external rectangle method, the Canny operator,

the Zernike moment operator, and the fast Fourier transform (FFT). Huang et al. [7] used an improved Canny operator to inspect the IC chip pin. Zheng et al. [8] also used the Canny operator to inspect the appearance of QFP chip lead. Chen and Perng [9] combined many digital image processing including normalization, shrinking, segmenting, and Fourier-based image restoration and defect identification on the basis of characteristics of statistical textures of the molding surface. Nurhadiyatna et al. [10] proposed a method which consisted of K-means clustering, Gabor filter, and Canny edge filter to detect surface defects on chip pad area. However, most of these research studies focused on detecting the existence of different types of defects rather than confirming the contours of defects. Furthermore, the studies on contour detection could hardly be applied in complex structural chips for two reasons: first, these methods strongly depend on threshold setting, which makes blurred edge detection an open technical challenge. A low threshold may cause false detection while a high one may lead to missed detection of weak edges. Second, the edges abstracted in the whole image belong to not only defects but also structures, which sometimes are almost indistinguishable.

In addition to traditional methods, some studies based on machine learning and hybrid method have been applied in this field. Chen and Liu [11] adopted a neural network architecture named adaptive resonance theory network 1 (ART1) to recognize defect spatial patterns. The neural network and derived methods were also applied in semiconductor wafer postsawing inspection [12] and light-emitting diode (LED) inspection [13–16]. Wang [17] integrated spatial statistics, kernel-based eigendecomposition, and support vector clustering to extract and separate four kinds of composite defect patterns. Li [18] proposed a method based on wavelet transform, rough set theory, and support vector machine to classify copper clad laminate defects. Xie et al. [19] developed a scheme to simulate datasets of these 4 patterns including rings, semicircles, scratches, and clusters for classifiers' training and testing. After that, defects in real optical images were resized and detected based on image processing operations and the trained support vector machine. Lin et al. [20] proposed the application of convolutional neural network- (CNN-) based defect inspector named LEDNet for LED chip defect inspection and achieved a low inaccuracy and localized exact defect regions as well. The central issue of learning-based methods can be distilled to the following: first, traditional networks and SVM cannot realize the defect localization and edge detection but the classification of defects. Second, unfixed shapes of defects increase the cost of time and energy on image annotation excessively. Third, many digital image processes before classification such as image segmentation,

feature extraction, and description artificially will bring the same problems with traditional methods. Fourth, the universality of trained models is so poor that new models should be designed and trained for different types or batches of productions.

In order to address these problems, it is necessary to derive a more effective and efficient way to detect the defect edge of complex structural chips. This paper proposes a method based on multiregion segmentation and edge correction as shown in Figure 1. As all structures of a chip are determined, the contours of them can be regarded as priori information to locate their position, which is the basis of image segmentation. An edge correction technique is used to modify the incomplete edges. The remaining fault edges caused by noise and structure are removed according to their shape feature while the right ones are preserved completely.

The remainder of this paper is organized as follows. Section 2 presents the proposed model that combines region segmentation search and edge correction for defect contour detection. Section 3 utilizes a kind of chip to demonstrate the capability of the proposed method. Section 4 presents the conclusion and directions for future work.

2. Research Method

As the chip contains many various structures inside, there are many overlap regions with different gray levels in the image, which are the basis to divide a chip image into different regions which change related to the position of structures. We propose a novel region segmentation method using a contour matching method and a two-pass algorithm.

2.1. Region Segmentation Search Methodology

2.1.1. Structural Location Based on Contour Matching. Although all the structures certainly appear in the chip image, the image brightness is irregular and structures may occlude mutually. Therefore, this paper used a contour matching method based on the Canny operator to solve these two problems [21].

(1) *Template Edge Model Building.* Each structure has a respective image as a template whose edges are used to build its model. The models are built by following the below-mentioned steps:

Step 1. Image graying: each Canny is only used to process the gray image. Thus, the template image should be grayed if it is a color image. Taking an RGB color image as an example, the formula used for graying is as follows:

$$I_{\text{gray}}(x, y) = 0.299 * I_{\text{red}}(x, y) + 0.587 * I_{\text{green}}(x, y) + 0.114 * I_{\text{blue}}(x, y), \quad (1)$$

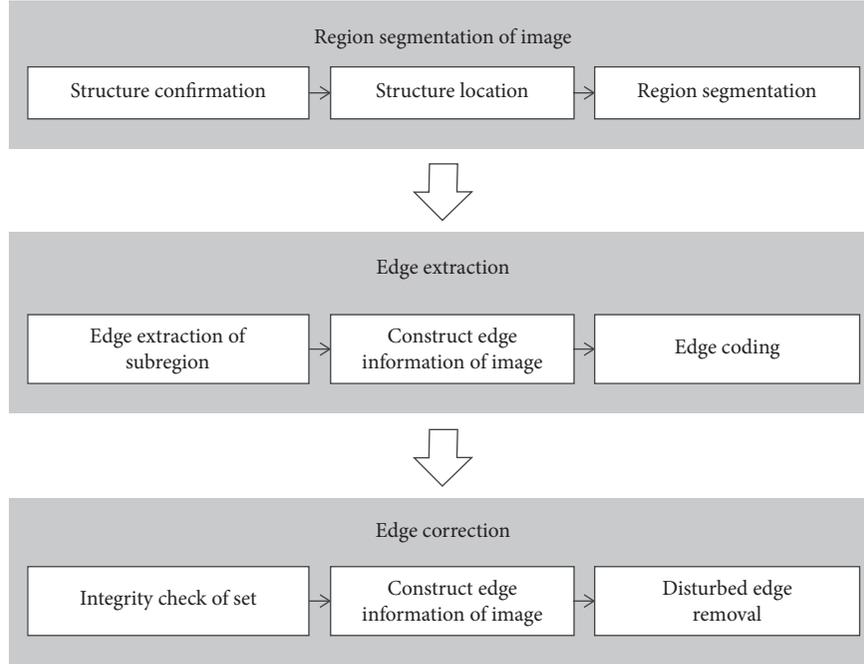


FIGURE 1: Flowchat of the proposed method.

where $I_{\text{gray}}(x, y)$ is the gray image value, $I_{\text{red}}(x, y)$ is the red channel value, $I_{\text{green}}(x, y)$ is the green channel value, and $I_{\text{blue}}(x, y)$ is the blue channel value.

Step 2. Image smoothing: each image is convoluted by Gauss filter to eliminate noise.

$$I(x, y) = G(x, y) * f(x, y),$$

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (2)$$

where $I(x, y)$ is the filtered image, $f(x, y)$ is the original image, and $G(x, y)$ is the two-dimensional Gauss function.

Step 3. Gradient calculation: gradient magnitudes and directions of an image are determined by Sobel operator:

$$P_x[x, y] = f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1) - f(x-1, y-1) - 2f(x-1, y) - f(x-1, y+1),$$

$$P_y[x, y] = f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1) - f(x-1, y+1) - 2f(x, y+1) - f(x+1, y+1),$$

$$M[x, y] = \sqrt{P_x[x, y]^2 + P_y[x, y]^2}, \quad (3)$$

$$\theta[x, y] = \arctan\left(\frac{P_y[x, y]}{P_x[x, y]}\right).$$

Step 4. Nonmaximum suppression: in order to ensure the accuracy of edge location, the gradient magnitude at each edge pixel should be compared with those at its two neighbors in the gradient direction. If it is larger, the pixel will be marked as a maximum edge point; otherwise, it will be marked as the background.

Step 5. Double threshold determination and edge connection: two thresholds T_{high} and T_{low} are required to classify the gradients of the remaining pixels that

have not been suppressed. T_{high} is usually set twice as high as T_{low} . The gradients higher than T_{high} are definitely strong edge points, while those lower than T_{low} are definitely nonedge points. The weak edges which are higher than T_{low} but lower than T_{high} are determined depending on whether there is more than one strong edge point in its neighborhood.

Step 6. Model building: because the model image may have some parts without any edge information, the

coordinates of the confirmed points need to be rearranged to reflect the start point as the center of gravity:

$$\begin{aligned}x_{p_i} &= x_{p_i} - x_{\text{gravity}}, \\y_{p_i} &= y_{p_i} - y_{\text{gravity}}, \\x_{\text{gravity}} &= \frac{\sum_{i=1}^N x_{p_i}}{n}, \\y_{\text{gravity}} &= \frac{\sum_{i=1}^N y_{p_i}}{n},\end{aligned}\quad (4)$$

where x_{p_i} is the x -coordinate of the i th point, y_{p_i} is the y -coordinate of the i th point, x_{Gravity} is the x -coordinate of gravity, y_{Gravity} is the y -coordinate of gravity, and n is the total number of edge points.

Step 7. Model storage: the coordinates along with their gradient magnitude and direction information are saved as the template models.

(2) *Model Matching.* The template models built are applied to search the structures in the chip image. Each model consists of a set of points $P_i^T = (x_i^T, y_i^T)$, gradients in X and Y directions $G_i^T = (Gx_i^T, Gy_i^T)$, and directions $D_i^T(x_i, y_i)$, where $i = 1, 2, \dots, n$, in which n is the number of elements in the Template (T) dataset. The gradient magnitudes need to be normalized for reducing the influence of illumination.

The gradient magnitudes and the direction of the search image (i.e., the chip image) are also calculated by the Canny operator, which are denoted by $G_{u,v}^S = (Gx_{u,v}^S, Gy_{u,v}^S)$ and $D_{u,v}^S(u, v)$, where (u, v) is the coordinate in the image, $u = 1, 2, \dots, p$, in which p is the number of rows in the search image, and $v = 1, 2, \dots, q$, in which q is the number of columns in the search image.

In the matching process, the template model is shifted in the search image to compare with the subimage using a similarity evaluation:

$$\begin{aligned}S_{u,v} &= \frac{1}{2} (S_{\text{dir}_{u,v}} + S_{\text{mag}_{u,v}}), \\S_{\text{dir}_{u,v}} &= \frac{\sum_{i=1}^n D_T(x_i, y_i) \cdot D_s(u + x_i, v + y_i)}{\sqrt{\sum_{i=1}^n D_T(x_i, y_i)^2 \cdot \sum_{i=1}^n D_s(u + x_i, v + y_i)^2}}, \\S_{\text{mag}_{u,v}} &= \frac{1}{n} \sum_{i=1}^n \frac{(Gx_i^T \cdot Gx_{(u+x_i, v+y_i)}^S) + (Gy_i^T \cdot Gy_{(u+x_i, v+y_i)}^S)}{\sqrt{(Gx_i^T)^2 + (Gy_i^T)^2} \cdot \sqrt{(Gx_{(u+x_i, v+y_i)}^S)^2 + (G_{(u+x_i, v+y_i)}^S)^2}},\end{aligned}\quad (5)$$

where $S_{u,v}$ is the score of similarity at point (u, v) in the search image, $S_{\text{dir}_{u,v}}$ is the score of direction similarity, and $S_{\text{mag}_{u,v}}$ is the score of magnitude similarity. The score is related to the portion of the structure visible in the search image. If the template model and the search image match perfectly, the score is 1; if the structure is not present in the search image, the score is 0.

In the process of calculating $S_{u,v}$ at a point, the partial score $S_{p_{u,v}}$ (i.e., equation (6)) is introduced to improve the algorithm efficiency. If $S_{p_{u,v}}$ is lower than a minimum score (S_{min}), the calculation of $S_{u,v}$ will be stopped and the matching evaluation of the next point will be started.

$$S_{p_{u,v}} = \frac{1}{m} \sum_{i=1}^m \frac{(Gx_i^T \cdot Gx_{(u+x_i, v+y_i)}^S) + (Gy_i^T \cdot Gy_{(u+x_i, v+y_i)}^S)}{\sqrt{(Gx_i^T)^2 + (Gy_i^T)^2} \cdot \sqrt{(Gx_{(u+x_i, v+y_i)}^S)^2 + (G_{(u+x_i, v+y_i)}^S)^2}},\quad (6)$$

where m is the number of points used for match. It is evident that $S_{p_{u,v}}$ is smaller or equal to 1, so the first restriction is $Sm > S^{\text{min}} - 1 + (m/n)$.

If there is a perfect match between the template and the model, the partial score at any point should be greater than the minimum score S^{min} , i.e., $(m/n)S_{p_{u,v}} \geq S^{\text{min}}$. However,

there are still some situations where the missing part of the object is checked first, the partial sum will be low that the structure in the chip image may not be considered as a match. So a new criterion which takes into account both restrictions is set that the evaluation of the partial score will be stopped at

$$S_m < \min \left(\left(S^{\text{min}} - 1 + \frac{1-g}{1-g} \cdot \frac{m}{n} \right), \left(S^{\text{min}} \cdot \frac{m}{n} \right) \right),\quad (7)$$

where g is a greediness parameter which determines the fraction of the second restriction.

This similarity measure combines several insights: the similarity measure is invariant to nonlinear illumination changes because all gradient vectors are normalized. More importantly, this similarity measure is robust when the object is partially visible or mixed with other objects.

2.1.2. Automatic Region Segmentation. After determining the position of each structure in the image, it is necessary to segment the image into several subregions according to the locations and the overlap relationships of the structures. In this paper, the two-pass algorithm is used to encode each overlap region class, for automatic segmentation of chip image.

(1) *Two-Pass Algorithms*. A connected region consists of adjacent pixels with the same pixel value. Therefore, the connected region can be found in the image by these two conditions. For each connected region found, a unique label is given to distinguish the other connected regions. In this paper, a two-pass connected domain labelling algorithm for 8 neighborhoods is used [22]. As shown in Figure 2, the specific steps are as follows:

Step 1. First pass: the image is scanned from left to right and from top to bottom. If the point scanned is the effective pixel (nonzero), the situation of the leftmost, upper left, top-right points in the eight neighborhoods of this point will be judged. If none of them are effective points, the point is the beginning of a new label.

Step 2. Second pass: in the first pass, the points in the same connected region may have different labels. Therefore, the image should be scanned in the same way to merge these labels as the following rules:

- (1) If the leftmost point and the upper right point of a labelled point are effective points, the label of all three points will be set to the smaller label of them
- (2) If the upper left point and the upper right point of a labelled point are effective points, the label of all three points will be set to the smaller label of them
- (3) Otherwise, the label of point will be set to one of four in the order of leftmost, upper left, topmost, and upper right

The points with the same label form a cluster.

(2) *Region Segmentation*. According to the positions of the structures in the image, the number of layers of structures on each pixel is determined. The pixels with the same amount of layers are regarded as an overlap region class. A chip image can be separated into target (class) regions and nontarget regions. At the same time, the target region contains several connected regions which are regarded as subregion of this class. A unique label is given to each subregion. The steps are as follows:

Step 1. Each pixel is determined which class it belongs to

Step 2. For each class, the pixels of this class are set to 1, and the others are set to 0

Step 3. The Two-pass algorithm is used to encode the binary image, so as to segment the image into different sub-region of different class

An illustration is shown in Figure 3. There are 3 classes in the image, in which the class 3 region is marked in deep gray. After image binarization and two-pass coding, the target region is divided into two subregions whose edges are marked in red and yellow, respectively.

2.2. Edge Correction

2.2.1. Incomplete Edge Correction. The edges of the whole image are extracted from each subregion and encoded to form the defect contour information. However, it is difficult to set a proper threshold to extract all the defect contours

perfectly. There are two cases in the incomplete defect contours as shown in Figure 4: (1) a contour is divided into several edges and stored in different labelled point sets and (2) some parts of a contour are undetected.

The beginning and end points of these incomplete contours are defined as endpoints, as shown in Figure 5. There are two cases of endpoints: (1) only one edge point is in the 8-neighborhood of endpoint and (2) only two adjacent edge points are in the 8-neighborhood of endpoint, as shown in Figure 6.

The steps for endpoint connections are as follows:

Step 1. If the distance between two endpoints of a set is less than the threshold T_1 , the set will be considered as case 1 with its endpoints connected using interpolation. The interpolated points are marked as the same label. The endpoint information of this edge will be updated.

Step 2. The minimum endpoint distance of every two set is calculated. If the distance is less than the threshold T_2 , the two sets will be considered as case 2 with their endpoints connected using interpolation. The interpolated points and the set with a bigger label are merged to the other set. The endpoint information of the two sets will be updated. This step will be repeated until the set will not change. The flowchart is shown in Figure 7.

2.2.2. Interference Contour Elimination. It is noteworthy that some of the edges extracted are not defect contours. These interference contours are usually the edges of some structures and appear as line segments. Thereby, linear fitting is applied on all the contours corrected.

Assuming that the polynomial of the line in which a contour lies is $p(x) = a_0 + a_1x$, all points in this contour are substituted into the polynomial:

$$\begin{cases} a_0 + a_1x_1 = y_1, \\ a_0 + a_1x_2 = y_2, \\ \vdots \\ a_0 + a_1x_n = y_n, \end{cases} \quad (8)$$

where n is the number of points. Equation (8) can be converted to obtain the least square solution of a_0 and a_1 through equation (9). If the residual norm of the fitted polynomial is less than the threshold value, the edge is considered as a line segment that needs to be eliminated.

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix}. \quad (9)$$

3. Result and Discussion

3.1. Defect Contour Detection. In this study, a certain kind of chip is presented to demonstrate the feasibility of the proposed method. The X-ray photos of chips selected randomly were taken for the purpose of bubble (defect) contour

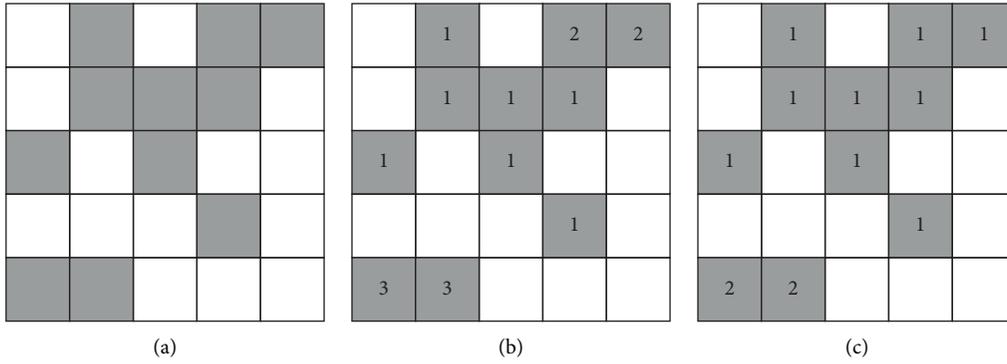


FIGURE 2: Two-pass algorithm illustration: (a) original image; (b) first pass; (c) second pass.

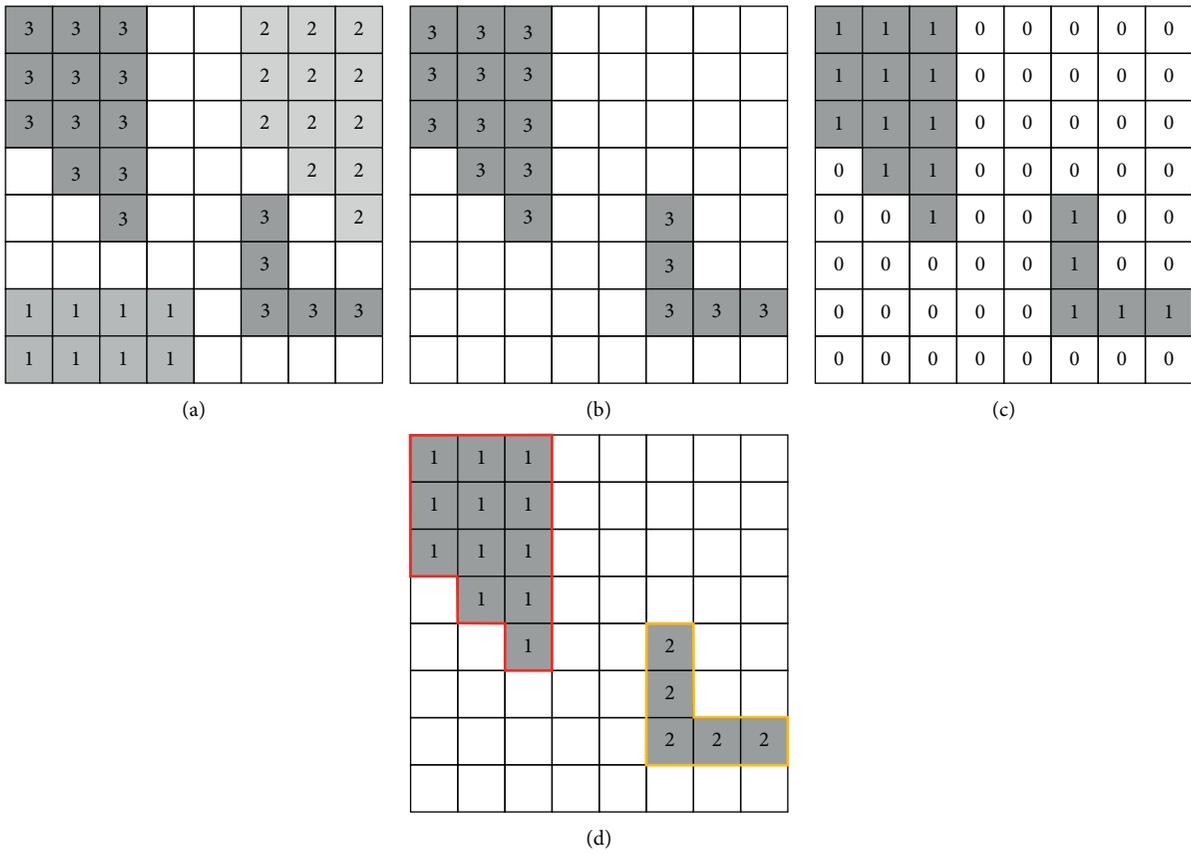


FIGURE 3: Region segmentation illustration: (a) class determination of image; (b) target region (class 3); (c) image binarization; (d) subregion coding.

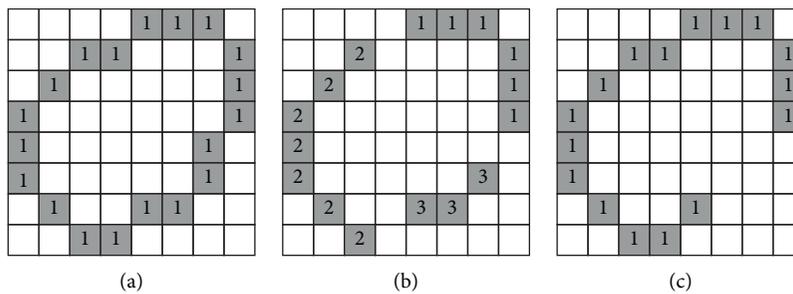


FIGURE 4: Incomplete edge illustration: (a) complete edge; (b) incomplete edge (situation 1); (c) incomplete edge (situation 2).

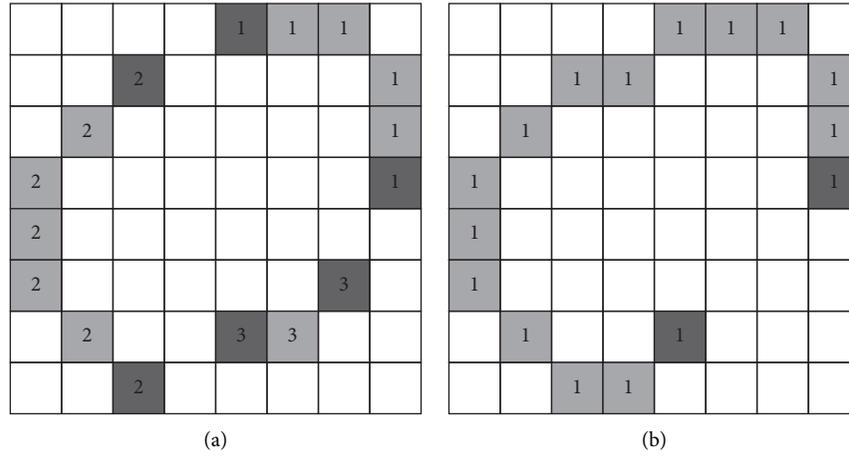


FIGURE 5: Endpoint of incomplete edge: (a) endpoint in situation 1; (b) endpoint in situation 2.

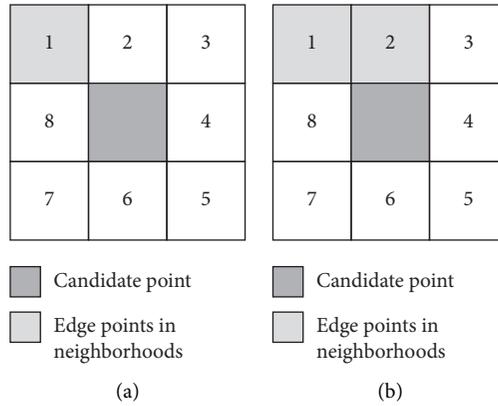


FIGURE 6: Endpoint in different situations: (a) endpoint in situation 1; (b) endpoint in situation 2.

detection. Considering that the chips were placed in the same container, not only the angle between them and the x -axis but also the sizes of them in images were identical. However, the places of structures were slightly different in each chip, and this paper set the structures of a chip as template images to realize multiregion segmentation.

The original images of 110 chips form the normal illumination set. The color channel properties of the above photos were adjusted artificially to simulate different illumination situations. First, the images are converted from the RGB model to HSV (hue, saturation, value) model. Then, the V values of some original images were increased or decreased stochastically and reconverted to the RGB model finally. These processed images were used to simulate the situation under strong or weak illumination. The original images together with the processed ones formed the whole image set. The chip images were subjected to noise; hence, the median filter is used to reduce the noise effect as image preprocessing.

Each structure of the chip was utilized to build a corresponding model which was matched in the chip image with the minimum score S^{\min} ($=0.4$) and the greediness parameter g ($=0.9$). As an illustration, the matching results

of structure 1 and structure 2 in one chip image under different illumination conditions are shown in Figure 8. According to the matching results, the multiregion segmentation was realized as shown in Figure 9.

The edges in each subregion were extracted by the Canny operator, which formed the uncorrected edges in the whole image. After incomplete contour correction and interference contour elimination with the same threshold T_1 ($=10$) and T_2 ($=10$), the contours of bubbles were detected accurately and marked in the right label as shown in Figure 10(b).

3.2. Qualification Assessment. In engineering applications, the qualification of a chip was assessed by humans so that the artificial assessment result of it was used as true value. In order to further prove the reliability of our method, the artificial assessment was adopted to evaluate the qualifications of the chips. The qualification of a chip is judged with the rule whether the range of any bubble is over half the distance between the external cavity where it is in.

Table 1 shows the comparison between the assessment results of our method and the artificial assessment. Another assessment method based on a deep learning method (YOLO

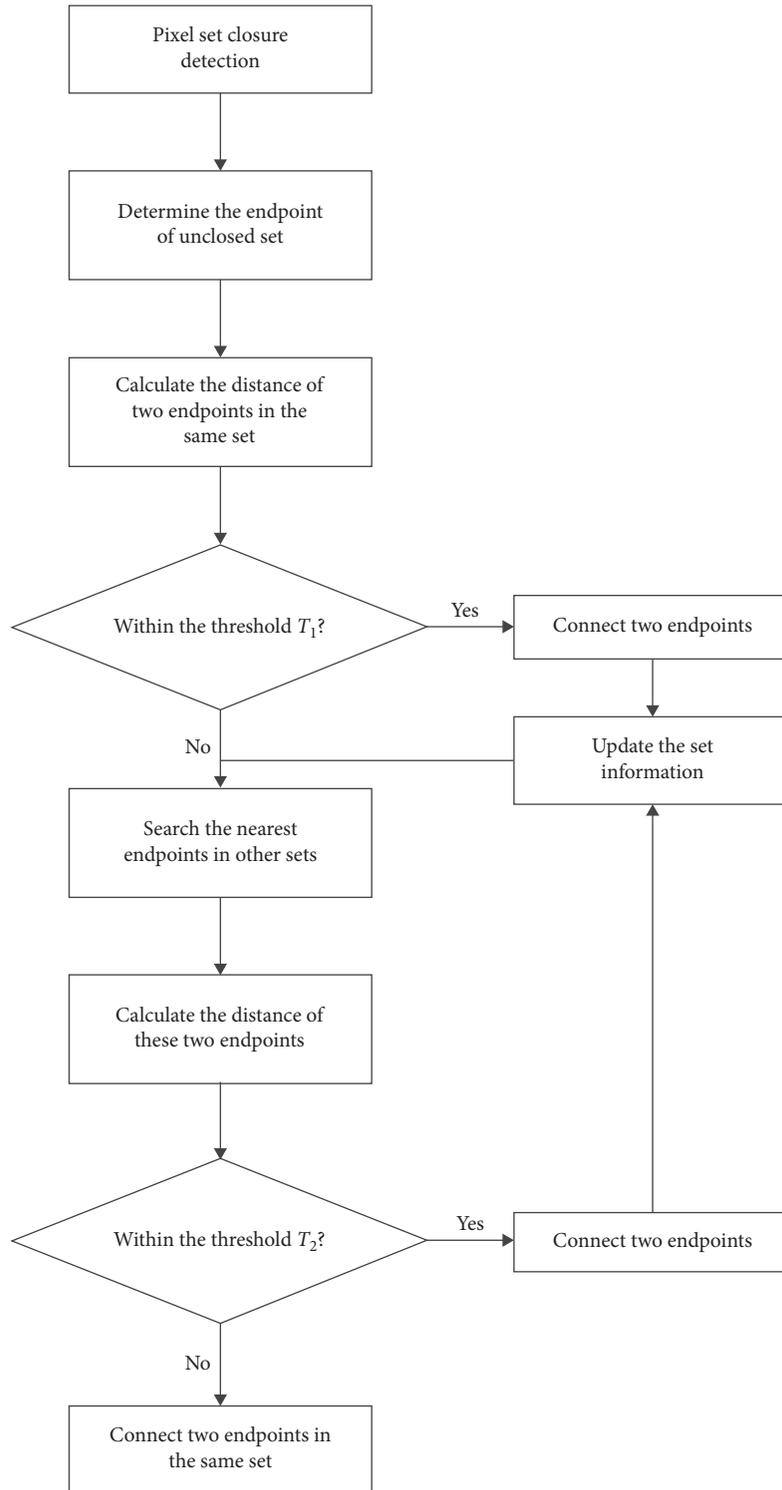


FIGURE 7: Flowchart of incomplete edge correct.

v4) [23] is also used for comparison. The detection result of bubbles is shown in Figure 10(c) and the assessment results of it are compared with the artificial assessment in Table 1. For each illumination situation, 40 images are chosen randomly as a test set while the rest are used as a training set. There were three outcomes: qualified, unqualified, and uncertain. When humans cannot judge whether the defects reach the distance

threshold, the artificial assessment will be uncertain. However, the machine assessment will be uncertain if the distance is between the 95% and 100% threshold. Accuracy rate and precision rate are chosen as indicators to verify the proposed method. Accuracy rate reflects the similarity degree between assessments. Precision rate shows the accuracy when machine assessment is qualified. Once a chip is judged as unqualified or

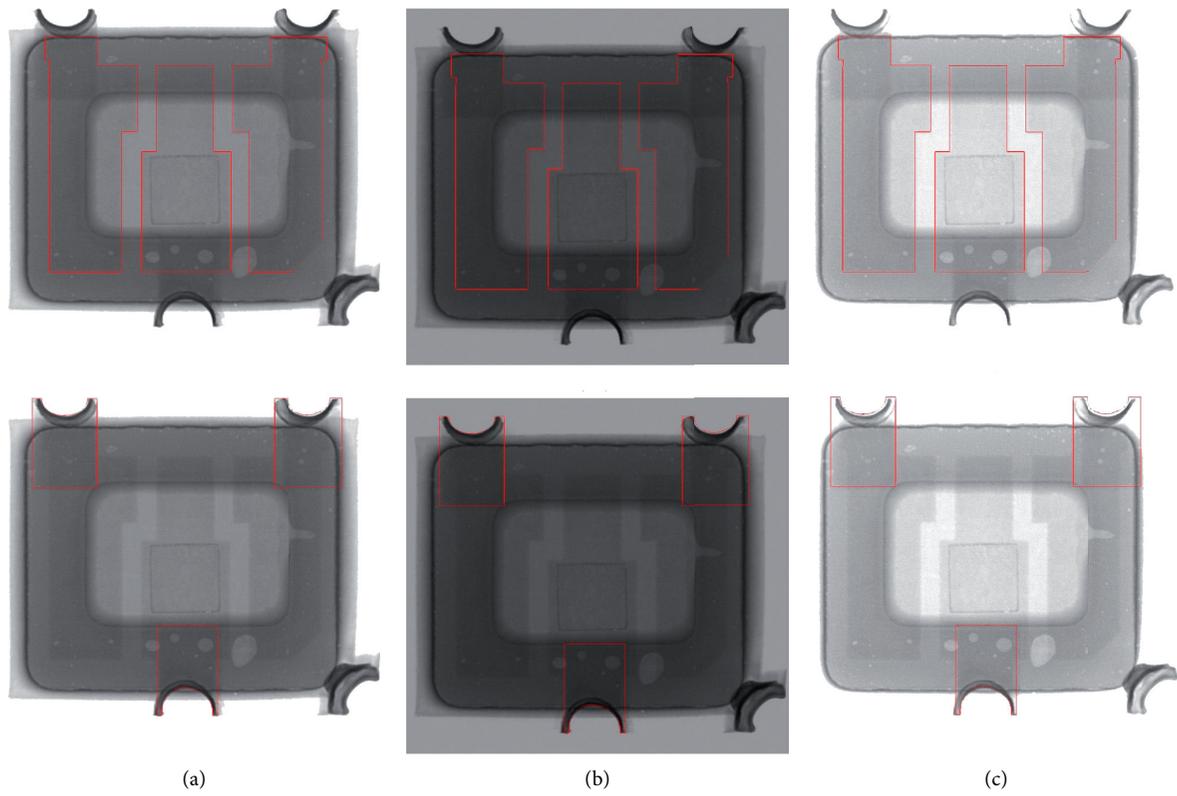


FIGURE 8: Structural location: (a) normal illumination; (b) weak illumination; (c) strong illumination.

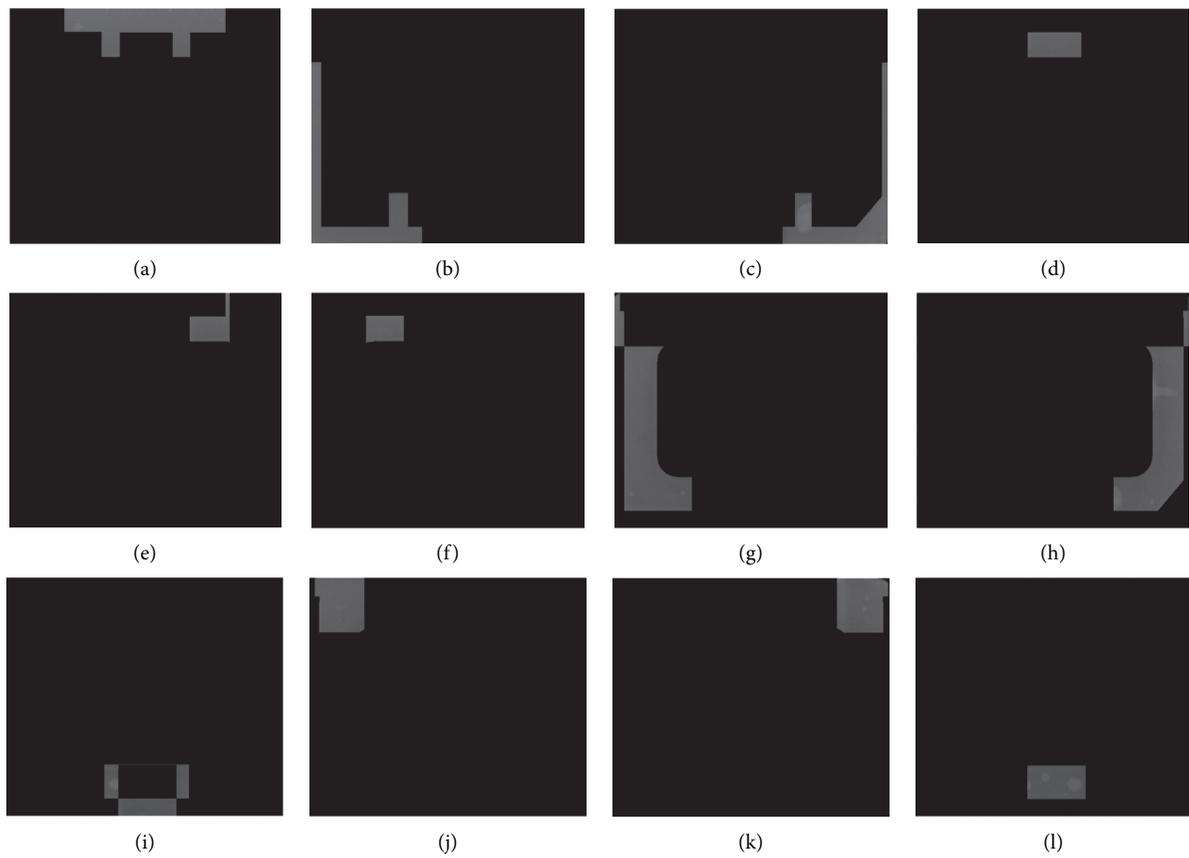


FIGURE 9: Automatic region segmentation.

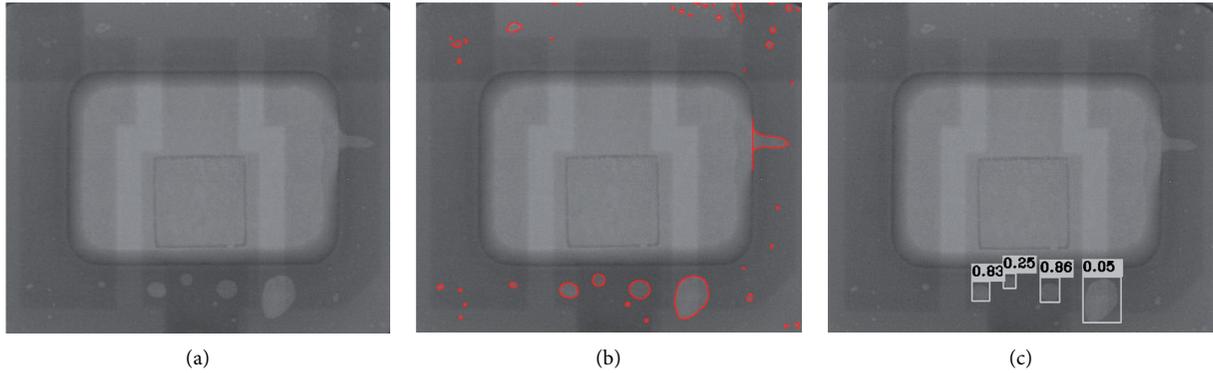


FIGURE 10: Detection result: (a) original image; (b) detected image of the proposed method; (c) detected image of YOLO v4.

TABLE 1: Comparison of assessment results.

Illumination situation	Method	Machine assessment	Artificial assessment			Total	Accuracy rate (%)	Precision rate (%)			
			Qualified	Unqualified	Uncertain						
Normal	Ours	Qualified	19	0	0	110	97.27	100.00			
		Unqualified	0	82	2						
		Uncertain	1	0	6						
	YOLO v4	Qualified	19	9	0						
		Unqualified	0	11	1				40	75.00	67.86
		Uncertain	0	0	0						
Weak	Ours	Qualified	50	3	0	113	92.04	94.34			
		Unqualified	0	49	2						
		Uncertain	0	4	5						
	YOLO v4	Qualified	17	14	0						
		Unqualified	0	7	0				40	60.00	54.84
		Uncertain	0	2	0						
Strong	Ours	Qualified	42	4	0	115	92.17	91.30			
		Unqualified	0	54	0						
		Uncertain	3	2	10						
	YOLO v4	Qualified	20	11	0						
		Unqualified	0	8	0				40	72.50	64.52
		Uncertain	0	0	1						

uncertain, it will be rechecked by humans or instruments; otherwise, no action will be taken on it. So, the latter indicator gets more important than the former one. The proposed method performed the best at normal illumination with the highest rate, while both two indicators had slight declines in other situations. One reason was that the same parameter was difficult to apply to all the changes of image contrast caused by the process of the HSV model. Even so, the precision rate was still higher than 92%. For comparison, because the size of the training set is small and the appearance of bubbles is infinite, the detection capability of YOLO v4 for bubbles is poor. To sum up, the experimental results demonstrated the effectiveness of the proposed method.

Table 2 shows the comparison on cost time of the above assessment methods. Compared to the previous way, the proposed method improves the efficiency by 3-4 times. Even the method based on YOLO v4 has the fastest speed, and it has other time costs for its long-time training and poor portability for other kinds of chips. The cost time of artificial assessment depended strongly on the amount of uncertain chips which took more time to confirm than

TABLE 2: Comparison on cost time.

Illumination situation	Machine assessment		Artificial assessment (s)
	YOLO v4 (s)	Ours (s)	
Normal illumination	9.8	15.2	57.2
Weak illumination	9.4	15.9	60.7
Strong illumination	9.5	16.1	61.3

qualified and unqualified ones. The efficiency of machine assessment was stable for its unique relevance to the amount of chips. Considering the accuracy and precision, the proposed method can play a better role in actual engineering application.

4. Conclusion Remarks

This paper presented a method to address the interference of structures when detecting defect contours of complex structural chips. Meanwhile, the information of every

contour was coded for further usage. The method consisted of two parts: region segmentation search and edge correction. The Canny operator and two-pass were utilized to segment the image into regions automatically according to the location of structures. The edges inside each region were extracted, respectively, in order to reduce the interference of structures. Among all the edges, the incomplete ones were corrected on the basis of criteria. The remaining interference contours were eliminated based on their shape afterwards. The experimental validation on the case study using the X-ray photos of a set of chips successfully demonstrated the effectiveness of the method. The high performance in detection compared to the traditional method could contribute to improving the accuracy of the subsequent qualification assessment.

Data Availability

The image data used to support the findings of this study have not been made available because they are not public.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Bin Lin and Jie Wang contributed equally to this work.

References

- [1] S.-H. Huang and Y.-C. Pan, "Automated visual inspection in the semiconductor industry: a survey," *Computers in Industry*, vol. 66, pp. 1–10, 2015.
- [2] E. K. Teoh, D. P. Mital, B. W. Lee, and L. K. Wee, "Automated visual inspection of surface mount PCBs," in *Proceedings of the Conference of the IEEE Industrial Electronics Society*, Pacific Grove, CA, USA, August 1990.
- [3] J. M. Gallegos, J. R. Villalobos, G. Carrillo, and S. D. Cabrera, "Reduced-dimension and wavelet processing of SMD images for real-time inspection," in *Proceedings of the IEEE Southwest Symposium on Image Analysis & Interpretation*, San Antonio, TX, USA, April, 1996.
- [4] H.-D. Lin, "Automated visual inspection of ripple defects using wavelet characteristic based multivariate statistical approach," *Image and Vision Computing*, vol. 25, no. 11, pp. 1785–1801, 2007.
- [5] H. H. Wu, X. M. Zhang, and S. L. Hong, "A visual inspection system for surface mounted components based on color features," in *Proceedings of the International Conference on Information & Automation*, Zhuhai, China, June 2009.
- [6] H. W. H. Wu, G. F. G. Feng, H. L. H. Li, and X. Z. X. Zeng, "Automated visual inspection of surface mounted chip components," in *Proceedings of the International Conference on Mechatronics & Automation*, Xi'an, China, August 2010.
- [7] Y. Huang, Q. Pan, Q. Liu, X. P. He, Y. F. Liu, and Y. Q. Yu, "Application of improved canny algorithm on the ic chip pin inspection," *Advanced Materials Research*, vol. 317–319, pp. 854–858, 2011.
- [8] J. Zheng, W. Li, Y. Wang, and M. Luo, "Qfp chip visual inspection system and its inspection method," *China Mechanical Engineering*, vol. 24, no. 3, pp. 290–294 + 301, 2013.
- [9] S.-H. Chen and D.-B. Perng, "Automatic optical inspection system for ic molding surface," *Journal of Intelligent Manufacturing*, vol. 27, no. 5, pp. 915–926, 2016.
- [10] A. Nurhadiyah, S. Loncaric, E. Prakasa et al., "Development of visual inspection system for detecting surface defects on sensor chip," in *Proceedings of the International Conference on Computer*, Amman, Jordan, April 2018.
- [11] F. L. Chen and S. F. Liu, "A neural-network approach to recognize defect spatial pattern in semiconductor fabrication," *IEEE Transactions on Semiconductor Manufacturing*, vol. 13, no. 3, pp. 366–373, 2000.
- [12] C. T. Su, T. Yang, and C. M. Ke, "A neural-network approach for semiconductor wafer post-sawing inspection," *IEEE Transactions on Semiconductor Manufacturing*, vol. 15, no. 2, pp. 260–266, 2002.
- [13] W.-C. Chen and S.-W. Hsu, "A neural-network approach for an automatic led inspection system," *Expert Systems with Applications*, vol. 33, no. 2, pp. 531–537, 2007.
- [14] C. Y. Chang, C. H. Li, S. Y. Lin, and M. D. Jeng, "Application of two hopfield neural networks for automatic four-element led inspection," *IEEE Transactions on Systems Man and Cybernetics Part C*, vol. 39, no. 3, pp. 352–365, 2009.
- [15] C.-F. Kuo, C.-T. M. Hsu, C.-H. Fang, S.-M. Chao, and Y.-D. Lin, "Automatic defect inspection system of colour filters using taguchi-based neural network," *International Journal of Production Research*, vol. 51, no. 5, pp. 1464–1476, 2013.
- [16] C.-F. J. Kuo, C.-T. M. Hsu, Z.-X. Liu, and H.-C. Wu, "Automatic inspection system of led chip using two-stages back-propagation neural network," *Journal of Intelligent Manufacturing*, vol. 25, no. 6, pp. 1235–1243, 2014.
- [17] C.-H. Wang, "Separation of composite defect patterns on wafer bin map using support vector clustering," *Expert Systems with Applications*, vol. 36, pp. 2554–2561, 2009.
- [18] T.-S. Li, "Applying wavelets transform rough set theory and support vector machine for copper clad laminate defects classification," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5822–5829, 2009.
- [19] L. Xie, R. Huang, N. Gu, and Z. Cao, "A novel defect detection and identification method in optical inspection," *Neural Computing & Applications*, vol. 24, no. 7-8, pp. 1953–1962, 2014.
- [20] H. Lin, B. Li, X. Wang, Y. Shu, and S. Niu, "Automated defect inspection of led chip using deep convolutional neural network," *Journal of Intelligent Manufacturing*, vol. 30, pp. 2525–2534, 2018.
- [21] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- [22] L. D. Stefano and A. Bulgarelli, "A simple and efficient connected components labeling algorithm," in *Proceedings of the International Conference on Image Analysis & Processing*, Venice, Italy, September 1999.
- [23] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: optimal speed and accuracy of object detection," 2020, <https://arxiv.org/abs/2004.10934>.