Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Automatic Die Inspection for Post-sawing LED Wafers

Chuan-Yu Chang[1], Chun-Hsi Li[2], Yung-Chi Chang[1], and MuDer Jeng[2]
[1]Department of Computer Science and Information Engineering,
National Yunlin University of Science & Technology, Douliou, Taiwan
[2]Department of Electrical Engineering, National Taiwan Ocean University, Keelung, Taiwan
chuanyu@yuntech.edu.tw

***Abstract*—Wafer defect inspection is an important process that is performed before die packaging. Conventional wafer inspections are usually performed using human visual judgment. A large number of people visually inspect wafers and hand-mark the defective regions. This requires considerable personnel resources and misjudgment may be introduced due to human fatigue. In order to overcome these shortcomings, this study develops an automatic inspection system that can recognize defective LED dies. An artificial neural network is adopted in the inspection. Actual data obtained from a semiconductor manufacturing company in Taiwan were used in the experiments. The results show that the proposed approach successfully identified the defective dies on LED wafers. Personnel costs and misjudgment due to human fatigue can be reduced using the proposed approach.**

***Keywords*—Post-sawing LED inspection, Automatic inspection**

I. INTRODUCTION

Post-sawing inspection is one of the essential processes to increase the yield before die packaging in the capital intensive manufacturing industries. The inspection contributes manufacturers to prevent unnecessary outcome on unrecoverable defects. Studies have been made on this field in [1-4]. Su *et al.* [1] proposed using neural networks for wafer post-sawing inspections where back-propagation (BP), radial-basis function (RBF), and learning vector quantization (LVQ) neural networks were utilized to inspect the wafers with a mean feature. Before inspection can proceed, locating positions of respective dies is necessary. Hence, we applied one previous developed result of our research team [5] -- CHDDNN to locate the dies in this paper.

An LED wafer contains thousands of dies. Conventional approaches may need operators to inspect defects manually with the aid of SEM. However, manual inspections requiring frequently zooming in and out make this approach ineffective. Contrastively, an automatic approach that needs no zoom in and out in inspection stage and can process on a large amount of tiny dies in a single scale determined at the image-acquiring stage is necessary. Thus, an automatic approach which can locate, inspect, and mark individual dies is proposed to inspect the post-sawing LEDs.

This paper is organized as follows: Section II describes the proposed defect inspection approach, including the die-locating, the extracted features, and the inspecting networks. Section III discusses configurations of the training parameters, and briefly demonstrates the inspection results of the proposed approach. Conclusion and future developments are summarized in the last section.

II. AUTOMATIC DIE INSPECTION

Figure 1 illustrates the automatic approach, where Chang's Contextual HNN [6] is adopted for the die-location; the defect inspection applies neural networks to inspect the located dies; and the inspection results are stored for the die picker to remove the defective dies. The normal dies may be sent to packaging process after the inspection.

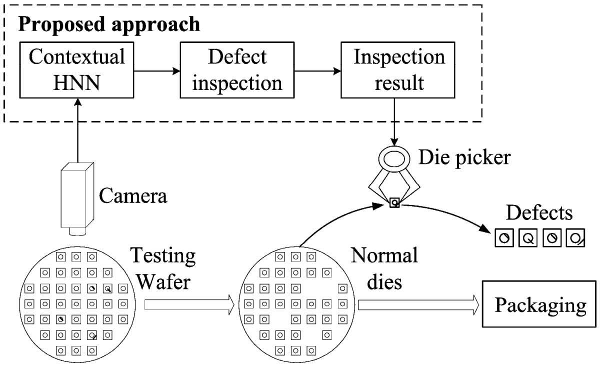


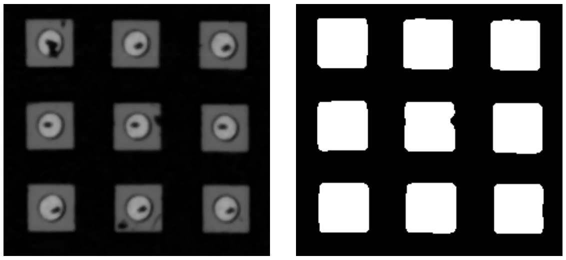
Figure 1. Schematic diagram of the automatic post-sawing wafer inspection approach.

A. Contextual HNN

Contextual HNN [6] proposed a pixel-labeling process that assigns pixels to edge points in accordance with their spatial contextual information. We applied this technique to label the wafer as die area or non-die regions. Before performing [6], we applied a 3×3 median filter to remove undesired noise. Figure 2 shows the result of the die detection where dies are detected and colored as white regions in Fig. 2(b). These regions are then numbered using an 8-adjencent connecting component

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

labeling [7] as different dies. According to the labeling results, we are able to inspect individual dies.



(a) (b)

Figure 2. The die detecting result. (a) the wafer image after removing noise with medium fiter. (b) the die detecting results using Contextual HNN [6].

B. Defect Inspection

The proposed method inspects each die on its appearance, light-emitting area and p-electrode. Figure 3 illustrates the inspection approach. If either one of the inspections fails, the die is marked as defective immediately.

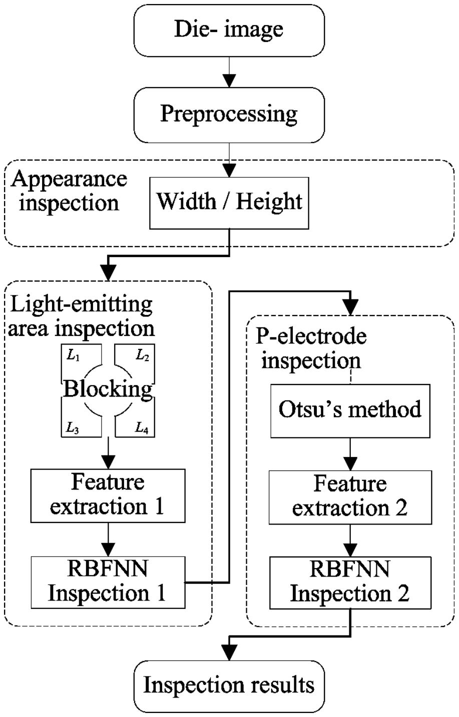
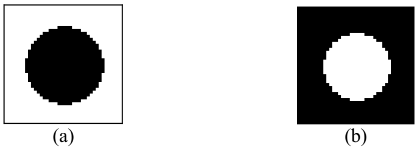


Figure 3. The proposed defect inspection approach. If the die fails in either of the inspections the die is marked as defective in the inspection result.

1) Appearance Inspection: Because the illumination is not ideally identical on the wafer, thus, before the appearance inspection begins, a lighting compensation technique [8] is applied to compensate the die images. The compensating bias is defined as the difference of the average of the intensity of top-ten-percent of the input image, and that of the top-ten-percent of the user preferred die image that located in central region of a wafer image. Appearance of the die is inspected with the width and height of the die, that are two parameters defining the shape (rectangular) of regular dies. The tolerances (T_{width} and T_{height}) are defined as ten percent of those of a regular die according to the manufacturer's specification. Once the die passed the appearance inspection, two masks designed according to the specification are applied to extract the light-emitting area and p-electrode. Fig. 4 shows the two masks for the extraction.



(a) (b)

Figure 4. (a) the mask (inside the border) for light-emitting area. (b) the mask fo the p-electrode.

2) Light-emitting area Inspection: Because we adopted neural networks as the inspecting approach, some statistical features representing this area are extracted for the inspection. These representative features are used in training the radial-basis function neural networks (RBFNN); the features are also the inputs of the trained RBFNN in the inspecting phase.

a) Blocking: Blocking is based on the observation that some of the features change significantly when the statistical area changes. Thus, some of the dies can be easily found to be defective when the features are considered within a small region. This is especially true when contamination exists on the surface of the light-emitting area. Four-region blocking is adopted to separate the square and hollow shape of the light-emitting area (refer to Blocking in Fig. 3).

b) Feature extraction 1: Considering structural unity of a regular die, statistical properties, such as mean (*m*), mean square (*ms*), standard deviation (*sd*), mean deviation (*md*), and entropy (*e*) statistics [9] are adopted as the features of the light-emitting area. Let N_{Li} denote the area of the *i*-th (i = 1 to 4) blocked light-emitting area L_i. The statistical features are obtained using the following equations:

$$m: \qquad F_m^L = \sum_{\forall s_i(x,y) \in L_i} s_i(x,y) \Big/ N_{Li} \tag{1}$$

$$ms: \qquad F_{ms}^L = \sum_{\forall s_i(x,y) \in L_i} \left(s_i(x,y)\right)^2 \Big/ N_{Li} \tag{2}$$

$$sd: \qquad F_{sd}^L = \sqrt{\sum_{\forall s_i(x,y) \in L_i} \left(s_i(x,y) - F_m^L\right)^2 \Big/ N_{Li}} \tag{3}$$

$$md: \qquad F_{md}^L = \sum_{\forall s_i(x,y) \in L_i} \left|s_i(x,y) - F_m^L\right| \Big/ N_{Li} \tag{4}$$

$$e: \qquad F_e^L = -\sum_{\forall s_i(x,y) \in L_i} s_i(x,y) \log_2\left(s_i(x,y) + 1\right) \tag{5}$$

where $s_i(x, y)$ is an intensity value of L_i at coordinate (x, y). The *m* feature is a measure on an integral average of the intensity in L_i. The *ms* gains intensity of the regions and obtains a value of greater magnitude than that of the *m* feature does. The *sd* and *md* measure the spread of intensity in L_i. The square root is performed to use the original units and the outliers increase the value of the standard deviation. Both *sd* and *md* change when there are defects or scars in the region. The last feature *e* measures how much information is carried; the greater the value, the greater the uncertainty. Defects or scars also affect the measured entropy. The feature vector F^L of the light-emitting area is organized as $(F^L_m, F^L_{ms}, F^L_{sd}, F^L_{md}, F^L_e)^T$.

c) RBFNN Inspection 1: In this study, the inspections of the light emitting area and the p-electrode are conducted using radial-basis function neural networks (RBFNN). RBFNN has the characteristics of simplicity, robustness, and optimal approximation. The RBFNN viewed as a curve-fitting network performs an input-to-output mapping with a supervised training strategy [10]. The network is presented with training pairs, each consisting of a vector from an input space and a desire network response. In the training stage, the network updates its weights to fit the desired output. Once the network is convergent and the training complete, we can use the network to inspect the region with the extracted features. Since we use the same inspecting networks in the light-emitting area and the p-electrode, we will described the networks later in the p-electrode inspection.

3) P-electrode Inspection: P-electrode usually contains a probing scar after the probing test. The unprobing p-electrode and the overlarge probing scar are both abnormal phenomenon. Thus, they are both considered as defective in the inspection. Some features of the defects are enhanced while considering in pure electrode region. Thus, the inspection inspects the features extracted in the separated region with another inspecting network.

a) Otsu's method: With an automatic approach, we apply Otsu's method to separate the p-electrode and the probing scar using the optimal threshold. The automatic algorithm [11] sequentially searches for an optimal threshold T_{opt} that maximizes the class separability and analyses the discriminant for maximum between-class variance that is used to measure class separability in gray levels.

b) Feature extraction 2: According the optimal threshold, statistical features of mean (*m*) and standard deviation (*sd*) statistics are extracted. Besides, the area of pure electrode (*a*) and the deviation that is smaller than standard deviation (*wd*) are also extracted. The features are extracted using the following equations:

a: $$N_O = F^O_{area} = \sum_{s(x,y)\geq T_{opt}} 1 \tag{6}$$

m: $$F^O_m = \sum_{s(x,y)\geq T_{opt}} s(x,y) \Big/ N_O \tag{7}$$

sd: $$F^O_{sd} = \sqrt{\sum_{s(x,y)\geq T_{opt}} \left(s(x,y) - F^O_m\right)^2 \Big/ N_O} \tag{8}$$

wd: $$F^O_{wsd} = \sum_{\left|s(x,y)-F^O_m\right|<F^O_{sd}} 1/N_O \tag{9}$$

where $s(x, y)$ is an intensity value at coordinate (x, y) in the p-electrode. Because the area of the p-electrode (*a*) (that is larger than T_{opt}.) is small and may change seriously while defect occurs, thus, it is considered as one of the features. The within standard deviation feature (*wd*) estimates the degree of "normal." Regular p-electrode has a *wd* value approximately equal to the standard deviation. Scars may produce a low mean value, larger standard deviation, small area, and small within standard deviation in the same instance. The feature vector F^O of the p-electrode is organized as $(F^O_m, F^O_{sd}, F^O_{area}, F^O_{wsd})^T$.

c) RBFNN Inspection 2: Figure 5 shows the architecture of RBFNN which consists of three layers: an input layer, a hidden layer of nonlinear processing neurons, and an output layer.

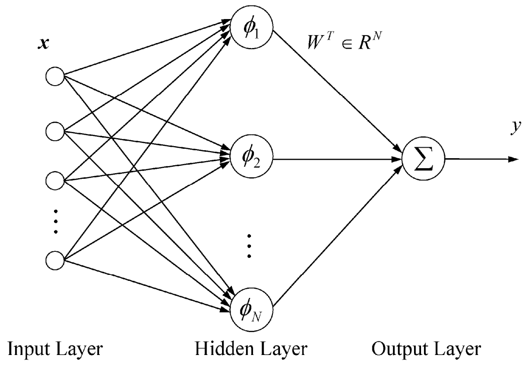


Figure 5. Architecture of the inspecting RBFNN.

The feature vector F^L (in light-emitting area inspection) or F^O (in p-electrode inspection) organizes the input $\boldsymbol{x}$ to the input layer. The radial-basis function gives the hidden layer the nonlinear transform ability [10]. The output layer produces the inspection results summing up the weighted results of the nonlinear transformation of the hidden layer. The output of RBFNN is:

$$\begin{aligned} y = f(\boldsymbol{x}) &= \sum_{k=1}^{N} w_k \phi_k(\boldsymbol{x}, \boldsymbol{c}_k) \\ &= \sum_{k=1}^{N} w_k \phi_k(\|\boldsymbol{x} - \boldsymbol{c}_k\|_2) \end{aligned} \tag{10}$$

where $\mathbf{x} \in R^{n\times1}$ (vector of $n\times1$ real numbers, organized by F^L or F^O in this paper) is an input vector; $\phi_k(\cdot)$ is the basis function; w_k are the weight vectors between the output neuron and k-th hidden neuron; $\|\cdot\|_2$ denotes the Euclidean norm; N is the number of neurons in the hidden layer; and $\mathbf{c}_k \in R^{n\times1}$ are centers of the radial-basis function (RBF) in the input vector space. In this paper, the Gaussian function is chosen as the basis function of the RBF network; it is defined as:

$$\phi_k(x) = \exp\left(-x^2 / \sigma^2\right) \tag{11}$$

where σ is the spread parameter, which is initially as :

$$\sigma = \frac{d_{\max}}{\sqrt{K}} \tag{12}$$

where $d_{\max}$ is the maximum Euclidean distance between the selected center and K is the number of centers [10].

In a simple training algorithm, the weights are the only parameter. This requires selecting a large number of centers to perform adequate sampling of the input. In the proposed method, the stochastic gradient-based supervised learning algorithm [10], in which the weights, position of the RBF centers, and the width of the RBFs are updated according to results of the supervised learning in each iteration, is adopted. The instantaneous error cost function *J*(*n*) is defined as:

$$\begin{aligned} J(n) &= \frac{1}{2}|e(n)|^2 \\ &= \frac{1}{2}\left[y_d(n) - \sum_{k=1}^{N} w_k(n)\phi\{\boldsymbol{x}(n), \boldsymbol{c}_k(n)\} \right]^2 \\ &= \frac{1}{2}\left[y_d(n) - \sum_{k=1}^{N} w_k(n)\exp\left(-\frac{\|\boldsymbol{x}(n) - \boldsymbol{c}_k(n)\|_2^2}{\sigma_k^2(n)} \right) \right]^2 \end{aligned} \tag{13}$$

where $y_d(n)$ is the desired network output for iteration *n*. The update equations for the network parameters are given by:

$$\begin{aligned} \boldsymbol{w}(n+1) &= \boldsymbol{w}(n) - \mu_w \frac{\partial}{\partial \boldsymbol{w}} J(n)\Big|_{\boldsymbol{w}=\boldsymbol{w}(n)} \\ &= \boldsymbol{w}(n) + \mu_w e(n)\psi(n) \end{aligned} \tag{14}$$

$$\begin{aligned} \boldsymbol{c}_k(n+1) &= \boldsymbol{c}_k(n) - \mu_c \frac{\partial}{\partial \boldsymbol{c}_k} J(n)\Big|_{\boldsymbol{c}_k=\boldsymbol{c}_k(n)} \\ &= \boldsymbol{c}_k(n) + \mu_c \frac{e(n)w_k(n)}{\sigma_k^2(n)} \cdot \phi\{\boldsymbol{x}(n), \boldsymbol{c}_k(n), \sigma_k\}[\boldsymbol{x}(n) - \boldsymbol{c}_k(n)] \end{aligned} \tag{15}$$

$$\begin{aligned} \sigma_k(n+1) &= \sigma_k(n) - \mu_\sigma \frac{\partial}{\partial \sigma_k} J(n)\Big|_{\sigma_k=\sigma_k(n)} \\ &= \sigma_k(n) + \mu_\sigma \frac{e(n)w_k(n)}{\sigma_k^3(n)} \cdot \phi\{\boldsymbol{x}(n), \boldsymbol{c}_k(n), \sigma_k\}\|\boldsymbol{x}(n) - \boldsymbol{c}_k(n)\|^2 \end{aligned} \tag{16}$$

where $\psi(n) = [\phi\{\boldsymbol{x}(n), \boldsymbol{c}_1(n), \sigma_1\}, \ldots, \phi\{\boldsymbol{x}(n), \boldsymbol{c}_N(n), \sigma_N\}]^T$, $e(n) = y_d(n) - y(n)$, $y_d(n)$ is the desired network output, and μ_w, μ_c, and μ_σ are the appropriate learning rate parameters.

The stochastic gradient-based supervised learning algorithm can be summarized as:

Input:	The feature vectors (F^L or F^O), the number of neurons in the hidden layer, the desired output $y_d(n)$, the learning rate parameters μ_w, μ_c, and μ_σ, and the maximum iteration *M*.
Output:	The convergent RBFNNs for defect inspection.
Task:	
Step 1.	Choose the centers for the RBF functions from the set of inputs.
Step 2.	Compute the initial spread parameter using (12).
Step 3.	Initialize the weights to small random values.
Step 4.	For one input vector, compute the network output using (10).
Step 5.	Update the parameter equations using (14), (15), and (16).
Step 6.	End the learning task if the network has converged or has reached the maximum iteration; otherwise, go to Step 4.

The convergent results of the parameters are used to setup the RBFNN for inspections.

III. Parameter Configuration

Several parameters affect performance of RBFNN, including the selection of initial centers, the number of the neurons in the hidden layer, the learning rate parameters μ_w, μ_c, and μ_σ in the stochastic gradient-based learning algorithm, and the cease criteria. We test for a proper parameter set and higher accuracy of the networks by fixing a single feature as the input of both networks. We randomly chose twenty normal and ten defective dies from 100 normal and 50 defective samples, respectively, as the training samples, and adopt the mean of an integral die as the fixed feature. One of five testing images containing 100 dies (including normal and defective dies) is used.

A. Initial centers

In equation (10), the weight w_k in the output layer and the centers $\boldsymbol{c}_k$ of the radial-basis function dominate the properties of the output. We first focus on the selection of the initial centers and compare several center selecting methods including random, k-means, self-organizing map and SOM with LVQ. Random is the easiest approach. However, to ensure adequately sampling the input space, a large number of centers selecting from the input data are necessary [10]. k-means is a well-known clustering approach that relies on proper initial center. Improper initial centers may easily reduce a local solution. The self-organizing map (Kohonen SOM), an alternative clustering approach, can be improved by combining with a supervised network, learning vector quantization (LVQ). We tested these center selecting methods for better performance. The tested results are given as TABLE I. It can be seen that the combing method of SOM with LVQ performs the best, even compared with the additional test of k-means with LVQ. Thus, we adopted the method of combining SOM with LVQ for initial center selection.

TABLE I. Accuracy Testing Using Different Method for Initial Center Selection.

	Random	k-means	SOM + LVQ	k-means + LVQ
Training sample	100%	100%	100%	100%
Testing sample	80%	87%	91%	88%

The test was set with: 10 neurons in hidden layer; random initial parameter values from 0.01 to 0.000001; learning rate parameters of 0.01s; at most 500 iterations

B. Number of the Neurons in the hidden layer

Deciding the number of the neurons in the hidden layer is also necessary to adequately sample the input space. Using the same testing samples and similar configurations, we re-tested using 5, 10, 15, and 20 neurons in the hidden layer with means of the light-emitting area and the p-electrode individually. The results are given in TABLE II that reveals that 5 or 10 neurons

TABLE II. Testing Accuracy Using Various Numbers Of Neurons In Yhe Hidden Layer.

Number of hidden neurons	5	10	15	20
RBFNN1	85%	91%	86%	81%
RBFNN2	91%	91%	83%	81%

The test was set with: selecting initial centers using SOM with LVQ; random initial parameter value from 0.01 to 0.000001; learning rate parameters of 0.01s; at most 500 iterations in the hidden layer give the best accuracy for the p-electrode; and that 5 neurons in the hidden layer gives the best accuracy for inspecting the light-emitting area. Thus, we used 10 neurons in the hidden layer of the RBFNN.

C. Learning rate parameters and cease criteria

Although it increases complexity, stochastic gradient-based supervised learning greatly improves the performance of the inspection and while being less complex than backpropagation [10]. Besides the common weights in the network, the center positions and spread parameters are adjustable parameters. We simply let the learning rate (μ_w) of the dominant parameter (weight) be larger than the others and obtain trail and error results using (μ_w, μ_c, μ_σ) equal to (0.01, 0.001, 0.001) in the RBFNNs for both the light-emitting area and the p-electrode. We concluded a convergent training process on these training samples where a cease criterion may be set at reasonable 350 in the training process to prevent poor inspections of the well-know problem of over-fitting [10].

D. Feature Combinations

We extracted several features in light-emitting area and p-electrode and tried to improve the inspection using their combinations heuristically. In features of the light-emitting area, we first tested the feature combinations of integral light-emitting area and then compared the feature combinations of blocked light-emitting area. TABLE III shows the testing accuracy of using individual features and their combinations. Though the testing accuracy is not very high, and using individual features is better than using feature combinations, we proposed blocking the light-emitting area to improve the inspection, as described in Section II.B. Accuracies of the features and their combinations of the blocked light-emitting area are shown in TABLE IV. We discovered that the individual features did not perform better than those extracted integrally, but the combinations of features outperform than using individual features extracted from integral or blocked, light-emitting area. Thus, blocking the L-region can improve the inspection accuracy of the combinative features.

TABLE III. ACCURACY TEST USING INDIVIDUAL FEATURES AND THEIR COMBINATIONS OF THE INTERGRAL LIGHT-EMITTING AREA

	F_m^L	F_{ms}^L	F_{sd}^L	F_{md}^L	F_e^L	$F_m^L + F_{ms}^L$	$F_m^L + F_e^L$
Testing accuracy	78%	76%	63%	40%	78%	63%	65%

TABLE IV. ACCURACY OF THE FEATURES AND THEIR COMBINATIONS USING FEATURES OF THE BLOCKED LIGHT-EMITTING AREA

	F_m^L	F_{ms}^L	F_{sd}^L	F_{md}^L	F_e^L	$F_m^L + F_{ms}^L$	$F_m^L + F_e^L$
Testing accuracy	76%	72%	43%	-[a]	70%	90%	91%

[a] Mean deviation was not tested due to the poor accuracy performance in test of integral light-emitting area.

In features of the p-electrode, we also tested accuracies of the features and their combinations. TABLE V shows accuracy test of the features and their combinations in p-electrode inspection. In p-electrode, however, we found that using all combination of the features performs best. Thus, all features were used in the p-electrode inspection.

TABLE V. ACCURACY TEST USING INDIVIDUAL FEATURES AND THEIR COMBINATIONS OF P-ELECTRODE

	F_m^O	F_{sd}^O	F_{area}^O	F_{wd}^O	$F_m^O + F_{sd}^O + F_{area}^O + F_{wsd}^O$
Testing accuracy	80%	76%	78%	74%	91%

IV. EXPERIMENTAL RESULTS

The system was implemented with Borland C++ Builder on Microsoft Widows XP operating system with Intel Pentium IV 2.8GHz processor with 2GB RAM. The testing images are provided by the manufacturer in Taiwan. Figure 6 shows the mask operation for the light-emitting area and the p-electrode. According the results, we were able to inspect the light-emitting area and the p-electrode, individually. For example, Figure7 (a) showed the results of the light-emitting area inspec-

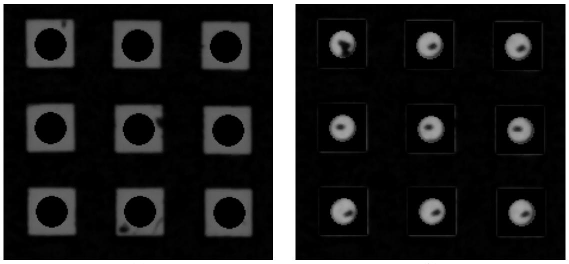

(a) (b)

Figure 6. Mask operations. (a) applying Fig. 4(a) on Fig. 2(a) with results of Fig. 2(b); (b) applying Fig. 4(b) on Fig. 2(a) with results of Fig. 2(b).

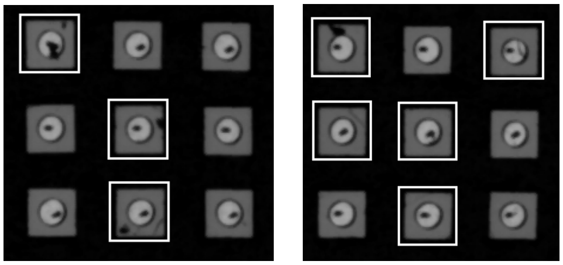

(a) (b)

Figure 7. Ispecting results. (a) the result of the light-emitting area inspection using the extracted region in Fig. 6(a). (b) the result of the p-electrode inspection using the extracted regionin Fig. 6(b). The results are directly shown with the die detecting result on the original wafer image, Fig. 2(a).

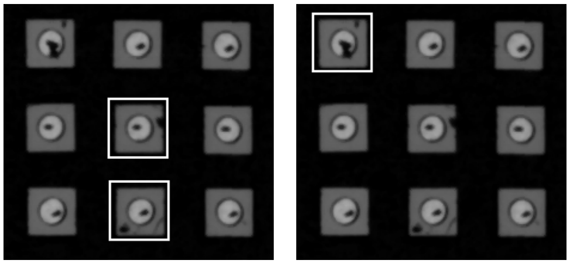

(a) (b)

Figure 8. Result of the integral inspection of the proposed approach. (a) inspecting results of Fig. 2(a). (b) another inspecting results of the post-sawing wafer. The results are obtained with $T_{width} = T_{height} = 5$. The physical size of each die was 200μm×200μm, which is equivalent to a 40×40 pixel-area.

tion with the extracted light-emitting area in Fig. 6(a), where two defects of die crack in the light-emitting area. Similarly, Fig.7 (b) showed the results of the p-electrode inspection that were inspected with the extracted p-electrode in Fig. 6(b), where the die with a pilling-off electrode were outlined correctly. Figure 8 showed results of the integral inspection. The proposed approach can successfully inspect the defects with correct results. Accuracy of the system was evaluated using the following equation:

$$Accuracy = \left(N_{TP} + N_{TN}\right)/N \qquad (17)$$

where N_{TP} was the number defective (positive) inspection results confirmed as defective by the engineer of the manufacturer; N_{TN} was the number of normal (negative) inspection results confirmed as normal; and N was the total number of dies in the inspection. The average accuracy of the proposed approach was illustrated in Table VI. Since the defective results in appearance inspection were very few, the implementation integrated the appearance inspection together with the successive inspections. The accuracy was 90.7% in the p-electrode inspection. It was 91.5% in the light emitting inspection. The integral accuracy was 92.4%. The inspection system had an average inspection time of 8.78ms per die for the 500 dies in our experiments.

TABLE VI. ACCURACY OF THE DEFECT INSPECTION

Appearance + p-electrode	Appearance + light emitting area	Integral (the proposed approach)
90.7%	91.5%	92.4%

V. CONCLUSION

Conventional LED wafer inspections are inefficient. Automatic inspection avoids human factors, and their corresponding costs. The proposed approach achieves automatic die inspection by locating dies with contextual-Hopfield neural network [7], inspecting dies in different regions, and recording the inspection results in the die map. The die inspection proceeds from the coarse to the fine to reduce unnecessary redundant inspections. Appearance inspection uses the width and height of the die according to the die-locating results. P-electrode inspection uses simple statistic features enhanced by Otsu's optimal threshold. Light-emitting area inspection uses dies with features of regions on the light-emitting area for better light quality.

Experimental results show the effectiveness of the scale-wise inspection with an average time of 8.78ms per die. A short inspection time is necessary for a wafer that contains thousands of dies. However, a short inspection time leads to accuracy loss because simple features are used. Thus, to improve the accuracy of the proposed approach, researchers may consider using alterative simple but powerful features in the future.

ACKNOWLEDGMENT

This work was supported by the National Science Council Taiwan, R.O.C., under grant NSC 94-2212-E-224-008.

REFERENCES

[1] C.-T. Su, T. Yang, C.-M. Ke, "Neural-Network Approach for Semiconductor Wafer Post-Sawing Inspection," *IEEE Trans. Semi. Man.*, vol. 15, No. 2, pp.260-266, 2002.

[2] K. W. Tobin, Jr., T. P. Karnowski, and F. Lakhani, "Integrated Applications of Inspection Data in the Semiconductor Manufacturing Environment," *SPIE, Metrology-based Control for Micro-Manufacturing*, vol. 4275, pp. 31-40, 2001.

[3] D.P. Mital and E.K. Teoh "Computer Based Wafer Inspection System," *Proceeding of International Conference on Industrial Electronics, Control and Instrumentation*, vol. 3, pp. 2497-2503, 1991.

[4] J.-M. Zhang, R.-M. Lin, and M.-J. Wang, "The Development of an Automatic Post-Sawing Inspection System Using Computer Vision Techniques," *Computers in Industry*, vol. 30, pp. 51-60, 1999.

[5] C.-Y. Chang, C.-H. Li, S.-Y. Lin, and M. Jeng, "Application of Two Hopfield Neural Networks for Automatic Four-element LED Inspection," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, in press.

[6] C.-Y. Chang, "Contextual-based Hopfield neural network for medical image edge detection," *Optical Engineering*, vol. 45, no. 3, pp.037006-037014, 2006.

[7] R. M. Haralick and L. G. Shapiro, Computer and Robot Vision, 1st ed., Addison-Wesley, pp. 28-48, 1992.

[8] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, Face Detection in Color Image, *IEEE Trans. Patten Analysis and Machine Intelligence*, vol. 24, 696-706, 2002.

[9] H. F. Weisberg, "Central Tendency and Variability", Basic Statistics -- International Handbooks of Quantitative Applications in the Social Sciences, vol. 1, no. 1, pp.1-88.

[10] Fredric M. Ham and Ivica Kostanic, *Principles of Neurocomputing for Science and Engineering*, Int. ed., McGraw-Hill Book Co., pp. 140-152, 2001.

[11] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems Man Cybernetics*, vol. 9, pp. 62-66, 1979.