International Journal of Image and Graphics
Vol. 9, No. 1 (2009) 133–152

REFERENCE-FREE MACHINE VISION INSPECTION OF SEMICONDUCTOR DIE IMAGES

ADA N. Y. NG and EDMUND Y. LAM*

Department of Electrical and Electronic Engineering,
University of Hong Kong,
Pokfulam Road, Hong Kong
**elam@eee.hku.hk*

RONALD CHUNG

Mechanical and Automation Engineering,
Chinese University of Hong Kong,
Shatin, Hong Kong

KENNETH S. M. FUNG and W. H. LEUNG

ASM Assembly Automation Ltd,
Kwai Chung, Hong Kong

Received 30 July 2007
Revised 21 April 2008
Accepted 2 October 2008

Advances in electronic technology have made integrated circuits (ICs) the fundamental components in all electronic devices. To increase their production yield by catching defects as early as possible, we need to perform quality assurance on the semiconductor dies during the assembly and packaging processes. A common approach is to employ machine vision to compare a test die with a "known good die". However, difficulties in ensuring identical imaging conditions (such as illumination) are limitations to this die-to-die comparison approach. Instead, in this work we develop a novel reference-free defect detection algorithm for an IC die by analyzing its image. By identifying intrinsic and extrinsic features of various segments in the image, we implement a classification scheme to identify whether the die is defective or not. We rely on the fact that normal ICs contain regular patterns, and the abnormal and irregular regions are classified as potential areas of defects. Experimental results show that the proposed reference-free defect detection algorithm can detect most of the defects from different types of IC dies, and can also correctly classify normal IC dies as non-defective. These results demonstrate the feasibility of the reference-free defect detection approach.

Keywords: Industrial inspection; pattern recognition; feature extraction; image analysis.

1. Introduction

Integrated circuits (ICs) are the fundamental components of all electronic devices today. In order to enhance the production yield of the ICs, it is vital to perform

quality assurance during various stages of the manufacturing process.[1] For instance, in the IC assembly and packaging, even a small dust particle or dirt can cause malfunction in the ICs at the end, resulting in a decrease of the production yield. Hence, defect detection of the IC dies is a necessary step in semiconductor manufacturing. This is commonly achieved through machine vision techniques.

This research follows on our previous work[2,3] in further developing a novel method to inspect an IC die with only a single capture of its image and a classification based on intrinsic and extrinsic properties of its various segments. The output is a labeling of whether the die is defective, and if it is, where the defects are. We call this a reference-free machine vision inspection, as it is fundamentally different from the "die-to-die" comparison commonly used in practice and widely studied by the researchers.[4–7]

In those methods, two images are needed during the inspection, with one of them being a reference image known to be defect-free. It is also possible to compare two repetitive patterns on the same image.[8] Processing is often applied as follows: The image under inspection is subtracted from the reference image to generate a difference image. Next, the difference image is filtered to attenuate the noise. Then, thresholding is applied to check if there is any defect.[9] For instance, the golden image template reference approach[10] needs to generate a golden template of each patterned wafer image under inspection and then perform a pixel-to-pixel comparison. Zhang *et al.*[11] proposed to adopt the reference-based method through extracting some features from the reference image, and then these features are compared with the inspect images. However, it should be noted that rotation, translation, brightness and alignment adjustments between the reference image and the inspected image are always needed before the comparison is possible.[12] Obviously, all these necessitate some non-trivial computation in pre-processing before the inspection, and in some cases, this method is simply impractical because no reference IC die image exists.

In contrast, our novel reference-free IC die defect detection algorithm only utilizes the image of the IC die under inspection, and performs the defect detection through image understanding and analysis. This is often fast and flexible, as there is no need to identify a suitable reference image and pre-process it to suite the imaging condition of the inspected image.[13] In addition, since the input to the proposed algorithm is only one image of any IC die, the machine vision system is ready to inspect any type of IC die by applying the proposed algorithm. There is also no need to adjust the lighting and other capturing parameters, such as the "die-to-die" comparison method and the automated wafer inspection method by Zoroofi *et al.*,[14] where a set of multi-spectral optical filters are used to capture wafers under different illumination conditions. Our method also has the advantage over a neural-network approach for semiconductor wafer post-sawing inspection by Su *et al.*,[15] where samples of the IC die types need to be collected first for training the neural-network. Our method is also more powerful than a previously proposed non-referential defect detection algorithm for the semiconductor wafers,[16,17]

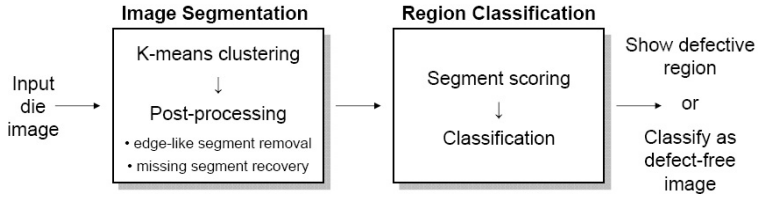


Fig. 1. The processing steps of the reference-free detection of IC die defect algorithm.

where instead of distinguishing between the non-defective and the defective IC die images, it is only targeted at the re-detection of defects within the historical defect imagery. There are also methods targeted at particular types of defects, such as the automated wafer defect detection proposed in Ref. 18 that focuses mainly on micropipes and the IC package inspection for extrusion defects that result from incorrect mounting of the die on the leadframe in Ref. 19.

Our reference-free machine vision inspection system consists of mainly two components, namely the "Image Segmentation" and "Region Classification", together with detailed steps as shown in Fig. 1. This paper is organized accordingly: Sec. 2 describes the image segmentation algorithm. Section 3 explains the region classification scheme. Section 4 illustrates and discusses the detection results. Finally, Sec. 5 concludes the paper and suggests future extensions. Before we proceed, we would like to point out that although our method is developed for the inspection of semiconductor die images during the assembly process, the concept of reference-free inspection through image understanding and classification can be applied to other instances, especially when the more traditional reference-based approach is costly or infeasible.

2. Image Segmentation

Image segmentation is often adopted in the analysis of an image and is typically the first step in computer vision. It divides an image into different homogeneous regions according to the pixels' properties. A homogeneous region is defined as a region where the pixels are connected and have similar properties, such as the intensity value.[20] Through the segmentation result, the machine vision system can understand how the image looks like. When applied on visual inspection of semiconductor manufacturing, these homogeneous regions should ideally represent different visual patterns of the IC die. However, due to the increase of component density in the IC, the semiconductor die becomes increasingly complex. The fact that we only obtain two-dimensional information based on this complex three-dimensional object gives further challenge.[21] Therefore it is usually difficult for the traditional image segmentation algorithms to provide reasonable segmentation results.[22,23] In order to segment the complex die effectively, the K-means segmentation algorithm, which is specially designed for segmenting the IC die images, is proposed to be used here.

The K-means segmentation utilizes the K-means clustering method[24–26] to cluster the pixels and then perform image segmentation. Basically, pixels in adjacent location and with similar intensity form an image feature, which is the circuit pattern or any possible defect of IC die image. Thus, classification of pixels into different clusters can help to segment the image feature. Practically, there are two parts in the proposed K-means segmentation algorithm: The first part roughly segments an image by using K-means clustering on each pixel, and the second part, called a K-means clustering post-processing, checks the homogeneity of the segmented region and determines the final segmentation result.

2.1. *K-means clustering of image pixels*

The K-means clustering of IC die image pixels can cluster the pixels into K groups according to the pixels' properties. However, pixels in the same cluster do not necessarily belong to the same segment of the final segmented image, for two reasons. First, the segmentation result is also affected by the corresponding location of the pixels. A segment is formed only from the pixels sharing similar properties and are connected to each other. If they are disjoint, they will be considered as separate segments, even if their properties are similar according to the clustering. Second, as described below, post-processing steps are needed to adjust the initial segmentation results to avoid issues like over-segmentation, so that defects in the semiconductor images are not merged with other patterns.

In ensuring a reasonable clustering result for the pixels, we need to look into four important issues. First is about the number of clusters, i.e. how we set the value of K. Second is about the formation of the feature vector for K-means clustering, which relates to the amount of information extracted from a pixel for the clustering step. Third is how we determine the initial centroid of each cluster. Finally, the fourth is about reducing the noise in the images so that the clustering result is not severely affected by the presence of noise.

2.1.1. *Number of clusters*

We need to determine the value of K for the number of clusters *a priori* before performing the K-means clustering. If K is too small, the clustering result will likely be not detailed enough for some IC dies with complicated circuit patterns. On the other hand, if K is too large, it may be an overkill for those dies with simpler patterns, and we have the tendency to over-segment the image afterwards. Therefore, ideally K should be set in accordance with the complexity of the circuit patterns. However, this knowledge is often unknown before image segmentation in the reference-free detection approach, and as a compromise we aim at obtaining a reasonable K that works with most of the IC die images.

In testing our collection of IC images during the assembly process,[a] we find that setting K to two is often insufficient for even moderately complicated circuit patterns. For the analysis point of view, this gives a result similar to thresholding using Otsu's method,[27] which is of limited use. Setting $K = 3$ can handle a larger pool of circuit patterns, yet with bearable processing time. For larger values of K, the computational time is a lot longer, and that is undesirable especially considering the need for near real-time defect detection. A plot on the processing time of 38 IC images with $K = 3, 4, 5$ is shown in Fig. 2. We sort the dies according to their complexities, and normalize the time to unity for the first die with $K = 5$. As can be seen from the Figure, $K = 5$ takes a lot longer to process in most cases, while $K = 4$ also requires more time for a small fraction of the test images, compared with $K = 3$. Hence for most practical purposes, using three or at most four clusters is the most suitable.

2.1.2. *Feature vector for K-means clustering*

The input data to the clustering algorithm is a set of feature vectors, where each feature vector v is associated with a particular pixel. For images, intensity is a fundamental property of the pixel. Another important property is contrast, which can be viewed as the difference in intensity of a pixel with its neighbors. These two properties represent different types of features for the K-means clustering. The former, pixel intensity, is an intrinsic feature, which deals with characteristics

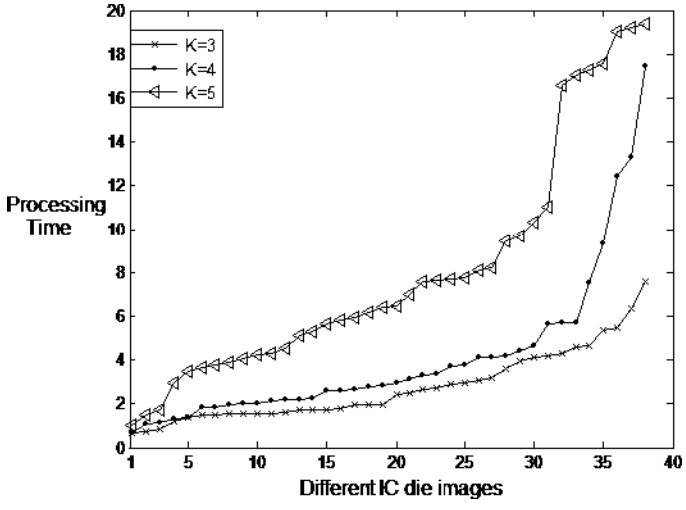


Fig. 2. The time requirement for performing K-means clustering for different K.

[a]The experimental results presented later in this paper are representative of the types of defects and the inspection process, while our database also contains proprietary circuit patterns which we cannot show here.

innate to the pixel. On the other hand, contrast is an extrinsic feature, involving the characteristics of the pixel in relation to its neighbors. Generalizing, we may have P intrinsic features and Q extrinsic features in a feature vector, then $v = [f_1^{(i)} \ \ldots \ f_P^{(i)} \ f_1^{(e)} \ \ldots \ f_Q^{(e)}]^T$, with dimensionality $P + Q$. Higher dimensionality means more information is fed to the K-means clustering, but this also induces more computation and leads to slow speed of the algorithm. When time is of the essence, we suggest using only the intrinsic feature for the clustering, whereas if a better clustering is desired, we see that it is advantageous to involve the difference in pixel intensities among neighbors as the additional extrinsic features.

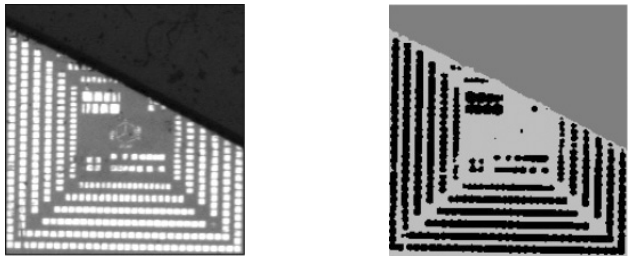
2.1.3. *Initial centroid of each cluster*

The choice of initial centroid values for the K clusters can affect the ultimate clustering result and the time it takes to process. While it is possible to choose the initial centroids from the feature vectors randomly, and this may be desirable in some applications, the processing time is rendered unpredictable. In an extreme case, when the values of all the initial centroids are very close to each other, it can take a long time to complete the K-means clustering. Hence, instead it is preferable to determine the initial centroids based on the distribution of feature vectors. We organize them according to the pixel intensities, which represent the dominant intrinsic feature, and choose the K centroids that have a wide separation in intensities. Due to the complexity of the gray-scale IC die image, the intensity values often spread across the entire intensity range, and therefore these initial centroids are usually well separated.

2.1.4. *Image noise reduction*

The raw IC die images from the solid state sensors may sometimes be too noisy for image segmentation. It is desirable to reduce the noise content through a small Gaussian lowpass filtering of the image before performing K-means clustering. As the pixels within the same image feature often have close intensity values, they have a higher chance to be clustered in the same cluster.

Following the discussion above, an example of clustering is shown in Fig. 3, where (a) is the original IC die image and (b) is the result following clustering, where different gray-levels denote different clusters. Note that after applying the K-means clustering into the IC die image, the pixels are grouped into different clusters instead of different segments. But from Fig. 3(b), it can be seen that even within the same cluster, the pixels of different circuit patterns are well separated by their locations. In order to perform image segmentation, the 4-neighbor pixels[28] in the same cluster are connected together to form initial segments. However, due to the limitations of the K-means clustering, the result sometimes contains issues such as missing segment and edge-like segment in some IC die images. We design the following post-processing schemes to mitigate such problems.

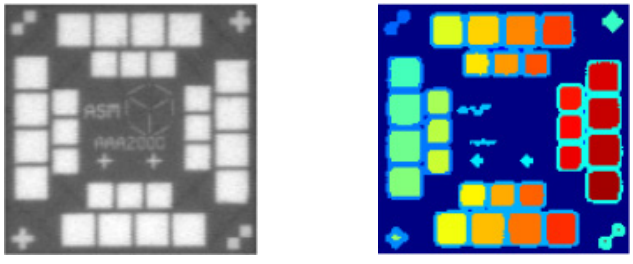


(a) A sample IC die image. (b) Result of the K-means clustering.

Fig. 3. An example of K-means clustering.

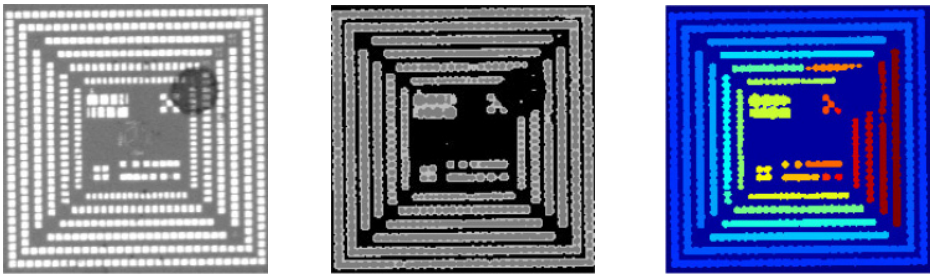
2.2. *K-means clustering post-processing*

The example segmentation results shown above indicate that the K-means clustering method can segment out most of the major circuit patterns and potential defect on the IC dies. But as mentioned in Sec. 2.1.1, clustering the pixels into about three groups may sometimes be insufficient or an overkill. For the latter, the problem of edge-like segment as shown in Fig. 4 occurs. In (b), it can be seen that some edge-like segments emerge around the white square circuit patterns on the four sides of the IC die. This is not desirable because the segments cannot represent any real circuit pattern or defect. On the other hand, if the clustering is insufficient, we have missing segments as shown in Fig. 5. In the initial segmentation result, after removal of edge-like segments shown in (c), the top right defect is still missing. It is because during the K-means clustering, the edge-like pixels dominate to form one cluster instead of the defect as shown in (b). As a result, even after the edge-like segments are removed, the corresponding defect still cannot be segmented out. Hence, in order to perform image segmentation on IC die images, the K-means clustering post-processing is needed to ensure the correctness of the segmentation. The following subsections describe the methods to remove the edge-like segments and recover the missing segments.



(a) A sample IC die image. (b) The initial segmentation result.

Fig. 4. An example of edge-like segment in K-means clustering, using three clusters. Different colors refer to separate segments.



(a) A sample IC die image. (b) The K-means clustering. (c) The initial segmentation.

Fig. 5. An example of missing segment in K-means clustering, using three clusters. Different colors refer to separate segments.

2.2.1. *Removal of edge-like segments*

For some of the IC dies, the edge-like segment forms when the transition pixels between the IC die's background and circuit patterns are clustered together. However, edge-like segments have shapes that are unlikely to represent any real circuit pattern or defect. It is therefore necessary to remove them for better representation of the IC die. To achieve this, we first check whether there are edge-like pixels in the K-means clustering result by comparing each cluster of pixels with the edge image of the IC die image. The latter can be obtained using methods such as Canny's edge detector,[29] as was used in the Figures in this paper, but other detection schemes may also be used especially in the interest of time. If there is no cluster of pixels similar to the edge image, there is no need to act further. Otherwise, if certain clusters of pixels also belong to the edge image, such as in Fig. 6, removal of the corresponding edge-like segments is needed.

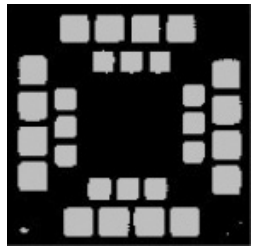
Figure 6 shows the three clusters of pixels of Fig. 4(a) separately. Instead of having circuit patterns or defects on each cluster of pixels, the third cluster of pixels shown in (c) shares similar pixels with the corresponding edge image shown in (d). Thus, the pixels in this third cluster are the edge-like pixels and together they form edge-like segments. Hence all the pixels in this third cluster should be removed.

For completeness of image segmentation, instead of eliminating all the pixels in the edge-like cluster, these edge-like pixels are kept in the segmentation result through merging into the closest remaining cluster in terms of gray-levels. After connecting the 4-neighbor pixels in the same cluster, the new image segmentation result of Fig. 4(a) is formed and it is shown in Fig. 7. As all the major circuit patterns are segmented out in this case, this segmentation result is final.

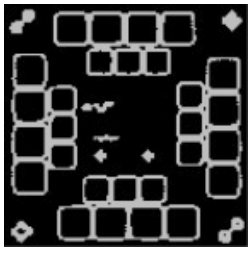
2.2.2. *Recovery of missing segments*

One further step is needed in the image segmentation process, namely the recovery of missing segments. After the initial IC die image segmentation, there are cases

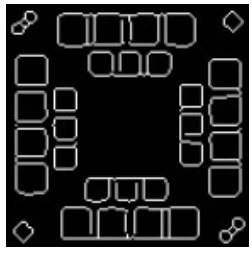
(a) The first cluster of pixels of Fig. 4(a) with the IC die background.



(b) The second cluster of pixels with the circuit patterns.



(c) The third cluster of pixels with the edge-like pixels.



(d) The edge image of Fig. 4(a).

Fig. 6. An example of edge-like segment in K-means clustering.

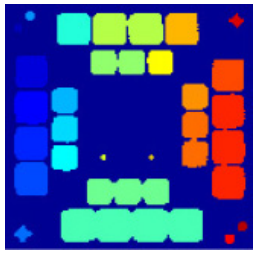


Fig. 7. Segmentation result after merging edge-like pixels.

where we can identify that certain segments are missed, and need to be reconstructed. They include the circuit patterns or defects of the IC dies, as shown in Fig. 5.

Consider in particular the edge-like pixels. This cluster may contain a small number of other circuit patterns (such as the example in Fig. 6(c)), and similarly it is also possible for it to contain defects. Since these small number of circuit patterns or defects are not the edge-like segments, merging them with other clusters of pixels may induce missing segments. Hence, the recovery of missing segments is necessary after the removal of edge segments. In theory, the pixels belonging to the missing segments should be grouped together, but they were incorporated independently into the nearest cluster because the total number of clusters, K, is fixed in advance.

This, however, causes those pixels to deviate substantially from the centroids of the respective cluster. Therefore, by checking the compactness of each cluster, the missing segment can often be identified.

Compactness is one of the most important measures of a cluster, and can be defined in terms of the average within-cluster variance as follows.[30] If x_k represents the data in the kth cluster, and there are n_k of them. $c_k = \sum(x_k/n_k)$ is then the center of the cluster, where the summation is over all pixels within the cluster, and

$$v_k = \frac{1}{n_k}\sum(x_k - c_k)^2 \tag{1}$$

is then the within-group variance. We need a threshold, which we term the "safe distance", to determine the value of v_k that is deemed acceptable for a normal defect-free variation within a cluster. Using a scheme suggested by Montolio *et al.*[31] and adapted in Ref. 30, we first calculate the half distance to the nearest cluster centroid. The minimum of the half distances from the three clusters is then set to be the safe distance. With this, the pixels with a pixel-to-centroid distance larger than the safe distance are extracted out to form the potential missing segment.

For those cases with a removal of edge-like segments beforehand, instead of measuring the compactness of individual clusters, we estimate the correctness of the segment by noting the variance of pixel intensities. This is because if there is no missing segment, the pixel intensities of this segment should be similar to each other. Then, the pixel-to-pixel intensity deviation, which is the difference between the mean intensity of pixels in this segment and the intensity of each pixel, is small. But if there is a missing segment, such as the case described in Fig. 5(c), the variance of pixel intensities in that segment is necessarily large. It should however be noted that pixels in the edge region of the segment may also have a large pixel-to-pixel intensity variation, as these noisy pixels are located between two or more segments. Therefore, the pixels in the edge region of the segment must be excluded in computing the variance. Figure 8(a) shows the pixel-to-pixel intensity variation for the non-edge region pixels of the background segment of Fig. 5(c). It can be seen that the missing segments' pixels have higher variation than the other normal pixels in the segment. The potential missing segments are then extracted using adaptive thresholding with non-maximum suppression, where the adaptive thresholds are set based on the variance of that segment.

After performing these post-processing once per IC image, the final segmentation result of our example shown in Fig. 5(a) is given in Fig. 9. All the major features have been properly identified and segmented.

3. Region Classification

Visually, the IC die defects are very different from the regular circuit patterns both in shape and intensity. For instance, the latter are usually periodic and regular, as they mainly contain straight lines, squares, rectangles or symmetric polygons. The

(a) The pixel-to-pixel intensity deviation for the non-edge region pixels in the background segment of Fig. 5(c).

(b) The potential missing segments (shown in white) of Fig. 5(c).

Fig. 8. An example of extracting missing segments.

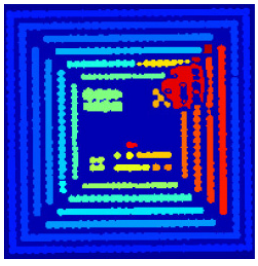


Fig. 9. The completed segmentation result after recovering missing segments.

word "periodic" here means that the circuit patterns commonly appear repeatedly in a die. In contrast, the defects on the IC die are in random shapes, aperiodic, and can be due to scratch, dirt, ink, crack, dust particle, probe mark and even deformation. As opposed to periodic, we identify these defects as "aperiodic" because they usually appear locally as anomaly in a die. Motivated by these differences in properties, we design a classification scheme to distinguish between the defects and normal circuit patterns effectively.

The aim of the region classification scheme is to differentiate defect regions from circuit pattern regions by assigning different scores to the segmented regions according to their properties and our scoring mechanisms. The classification is based on the scores, and thus our aim is to assign distinct ones for circuit patterns compared to the many different types of defects. Note that if there is no defect, all the segmented regions should be classified into the circuit pattern group. The corresponding IC die is then labeled as a normal die. Conversely, the segmented regions of defects should be classified into the defect group, and the corresponding IC die is then considered defective.

The scoring represents the segmented region mathematically in terms of the probability that a region is a defect. This is decided based on certain selected properties of the region. We first explain the characteristics of the target defects in

Sec. 3.1. Then, Sec. 3.2 describes the selection of region properties that are used to compute the score. Next, the rules for the scoring are then explained in Sec. 3.3.

3.1. *Defect characteristics*

In our reference-free approach of defect detection, we aim to identify defects that have clear visual differences with the circuit patterns. But because of the presence of noise, we must set certain criteria as the basic "requirement" for a defect. A feasible one that we use is shown in Table 1. Certainly, when the application calls for even more stringent or a relaxation of these requirements, the defection detection scheme can be adapted accordingly.

In Table 1, the intensity contrast means the difference of intensity values between a region and the corresponding neighboring regions. A defect is often very distinct from its background and has a large intensity contrast. In the gray-level IC die image, the dark region usually represents the scratch, dirt, ink, crack, dust particle and even probe mark, hence the defect is highly possible to be low in intensity. Moreover, since a defect occurs randomly, the shape of the target defect is often random. Lastly, the minimum size of defect is adopted to effectively exclude the image noise.

3.2. *Selection of region properties*

For a segmented region, properties such as area, compactness,[32] eccentricity, major and minor axis lengths, mean intensity, rectangularity,[33] rectilinearity,[34] solidity and standard deviation of intensity are important characteristics for classifications. However, it should be noted that the speed or computation complexity in calculating these properties must be taken into account, as the overall defection detection scheme must be completed within a short time. In addition to these properties, which are intrinsic to the segmented regions in general, we also consider extrinsic properties, which are the characteristics of a segmented region in relation to its surroundings. The rationale is that any significant dissimilarity between a region and its neighborhood is also a clue that the region is likely to be defect, again because circuit patterns tend to be similar in nature and thus the difference between adjacent ones in terms of their properties is ordinarily small. Based on the above principle, five region properties (three of them intrinsic, two extrinsic) are selected for computing the score for classification, as described below.

Table 1. Some possible characteristics of defects.

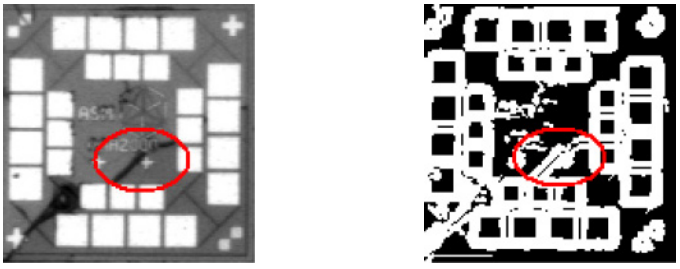
Minimum intensity contrast:	25 gray–levels
Intensity:	Dark
Shape:	Random
Minimum defect size:	3 pixels $\times$ 3 pixels

Intrinsic properties

- *Region area*
  The region area is essentially the number of pixels in the corresponding segmented region, which is useful for checking the validity of a segmented region in possibly being a defect. Given the criteria stated in Table 1, if the region is smaller than 3 pixels $\times$ 3 pixels, we would give a score of zero, meaning that it cannot be a defect.
- *Mean intensity*
  The mean intensity is the average intensity value of all the pixels within the segmented region. It is expected that the defect has a different mean intensity from the other circuit patterns. The mean intensity value can also be used to calculate the contrast of the region.
- *Rectangularity*
  The rectangularity measure is the area ratio of the segment to the segment's minimum rectangular bounding box,[33] which is the minimum rectangle that can enclose the segment. This is an effective measure because circuit patterns tend to be rectangular, whereas for random defects the shapes are highly unlikely to be so.

Extrinsic properties

- *Intensity contrast*
  Intensity contrast measures the difference between the intensity value of the segmented region as opposed to its neighbor segments. We expect the defects to score sufficiently high in this respect, because of their general dissimilarity with its neighborhood.
- *Contrast map*
  The second way to model the visual contrast is specially designed for the line type or scratch-like defects. We develop a novel contrast map to record the place of the IC die image where the intensity contrast is sufficiently large according to the defect characteristics. Figure 10(b) shows an example of contrast map, which is generated by comparing the IC die image pixels' intensities with its extended neighborhood. Based on this contrast map, we model the visual contrast as the ratio of the number of contrast pixel of a region to the area of that region. From Fig. 10(b), it can be seen that the ratio of the number of contrast pixel of the scratch-like defect region to the area of that region is higher than that of the white square circuit pattern region. Almost all the pixels in the thin line type or scratch-like defect region are contrast pixels. On the other hand, for the normal circuit pattern region, such as the white square in Fig. 10(b), only the pixels near the edge are contrast pixels. Hence, the contrast pixel ratio of a region can distinguish between the line type or scratch-like defect region and the normal circuit pattern region.



(a) A sample IC die image. (b) The contrast map, with white pixels indicating places where the intensity contrast is sufficiently large.

Fig. 10. An example of scratch-like defect (marked by red oval) and contrast map.

3.3. *Rules for scoring*

Once the region properties are obtained, the score of each region can be calculated according to certain rules. They are set according to the characteristics of defects commonly found in the IC dies. For our purpose here, we want to achieve high scores for defects regions and low scores for segments that belong to normal circuit patterns. The scores can be interpreted as probabilities of defects, which are computed from individual sub-scores as reflected from the various intrinsic and extrinsic features. As an example, ink defect is a common defect of the IC die which is dark and irregular in shape. Hence, a segmented region with low intensity value and a small rectangularity measure is set to a high mark. On the other hand, if a segmented region is likely a circuit pattern or does not match with the defect characteristics, such as a region with a small area or with a very high rectangularity measure, the region's score would be set to very low.

In order to speed up the processing time, we may not necessarily calculate all of the intrinsic and extrinsic features for every segment. For instance, there is no need to calculate the intensity contrast if a region is found to have a very high rectangularity measure or with a small region, both of which suggesting that it cannot be a defect irrespective of the intensity contrast.

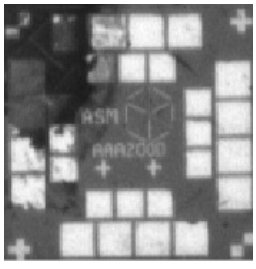
After computing the score for each segmented region, the regions with a score higher than a certain threshold are classified into the defect group. The value of the threshold is determined *a priori* for the type of IC dies at hand. Note that often some experimental trial-and-error may be needed to adjust this value.

4. Results and Discussion

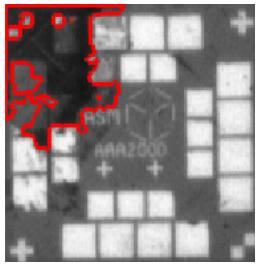
Different IC die images are tested with the proposed reference-free IC die defect detection algorithm. Each IC die image is first segmented into different regions. Next the region properties stated in Sec. 3.2 are calculated and then a score is computed for each segmented region. Classification of the region, and in turn the IC dies, into normal or defective is performed based on the scores.

4.1. *Defect detection results*

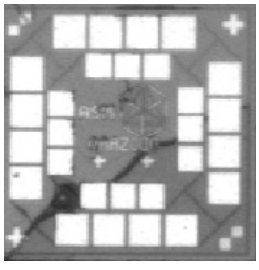
Both defective IC dies and normal IC dies are tested. The proposed algorithm is capable of differentiating the defective regions and normal circuit pattern regions in an IC die. Figure 11(a) shows an example where the ink defects spread over the surfaces of the IC dies. The major part of these ink defects are detected and, for the purpose of illustration, circled in red on the defect detection result as seen in (b). Figure 11(c) is the same as the defective die shown in Fig. 10(a), and the defect detection result, where the defect regions are circled in white, is shown in (d). Figure 11(e) presents a defect due to the probe mark on an IC die. Although it is often washed out by chemicals, sometimes it is possible that only part of it is removed and the remaining mark forms a semi-transparent defect as illustrated in (e). The detection of this defect is given in (f).



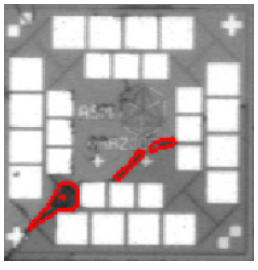
(a) The IC die with ink defect at the top left corner.



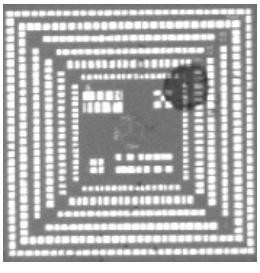
(b) The defect detection result.



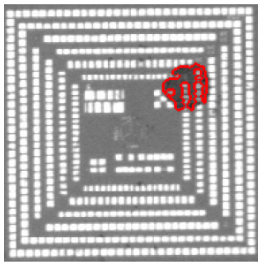
(c) The IC die with ink defects at the bottom.



(d) The defect detection result.



(e) The IC die with semi-transparent defect in top right.



(f) The defect detection result.

Fig. 11. Some defect detection example.

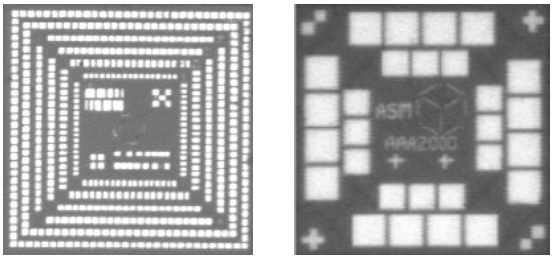
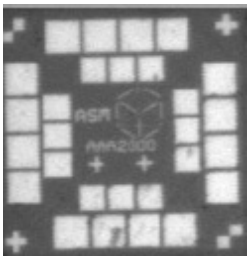


Fig. 12. Some examples of non-defective IC dies.

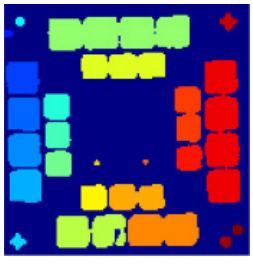
For our algorithm to be useful, we also require that normal IC dies are not classified as containing defective segments. In fact, in a typical industrial process, most of the IC dies should be defect-free. Some normal IC dies are given in Fig. 12, which are successfully classified as non-defective dies by the proposed algorithm of reference-free IC die defect detection.

4.2. *Discussions on defect detection results*

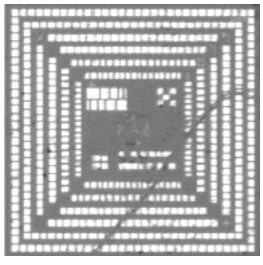
From the above defect detection results, it can be seen that a variety of defect patterns can be detected by the proposed method. But we should also point out that invariably, there are still some challenging situations where our algorithm is not yet capable of detecting the defects. To illustrate, we present two such cases in Fig. 13. Figure 13(b) and (d) show the segmentation results of the respective dies



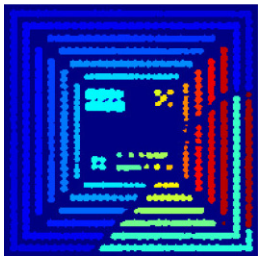
(a) A sample IC die image with defect.



(b) The final segmentation result of (a).



(c) Another sample IC die image with defect.



(d) The final segmentation result of (c).

Fig. 13. Examples of defects that are challenging to detect.

shown in (a) and (c). In the former case, the defect region on the second bottom left white square merges with the background of the IC die. In the latter, the scratch from bottom left to middle right has similar gray-level with the corresponding background, so it cannot be segmented out individually. Defects essentially hide themselves in the segmentation process, and thus cannot be detected with any classification scheme afterwards.

5. Conclusions and Recommendations

In this paper, we present a novel reference-free IC die defect detection algorithm for the semiconductor assembly and packaging processes by performing image analysis of a single captured image. More than 20 different types of IC dies are tested under the proposed scheme. Results indicate that it can distinguish between the defective IC dies and non-defective IC dies effectively. This shows the feasibility of the reference-free defect detection approach. While there are still challenging cases where the defects merge with the circuit patterns, making them very difficult to detect, we expect that further fine-tuning to the algorithm can improve the detection capability of the method.

Acknowledgments

This work is supported in part by the Innovation and Technology Fund UIM/111 of the Hong Kong Special Administrative Region Government in co-operation with ASM Assembly Automation Ltd.

References

1. J. Cheng, R. Chung, E. Y. Lam, K. S. Fung and Y. Xu, "Optimization of bit-pairing codification with learning for 3D reconstruction," *International Journal of Image and Graphics* **7**(3), 445–462 (2007).
2. A. N.-Y. Ng, E. Y. Lam, R. Chung, K. S. Fung and W. Leung, "Reference-free detection of semiconductor assembly defect," *Machine Vision Applications in Industrial Inspection XIII, Proceedings of SPIE* **5679**, 27–35 (2005).
3. A. N.-Y. Ng, E. Y. Lam, R. Chung, K. S. Fung and W. Leung, "Automatic segmentation for visual inspection in semiconductor manufacturing using multiscale morphology," *7th International Conference on Quality Control by Artificial Vision*, pp. 215–220 (2005).
4. Y. L. Zhang, N. Guo, H. Du and W. H. Li, "Automated defect recognition of C-SAM images in IC packaging using Support Vector Machines," *International Journal of Advanced Manufacturing Technology* **25**(11–12), 1191–1196 (2005).
5. B. H. Khalaj, H. K. Aghajan and T. Kailath, "Patterned wafer inspection by high resolution spectral estimation techniques," *Machine Vision and Applications* **7**(3), 178–185 (1994).
6. C.-J. Lu and D.-M. Tsai, "Independent component analysis-based defect detection in patterned liquid crystal display surfaces," *Image and Vision Computing*, in press.
7. S.-U. Guan and P. Xin, "A golden block self-generating scheme for continuous patterned wafer inspections," *10th International Conference on Image Analysis and Processing*, pp. 436–441 (1999).

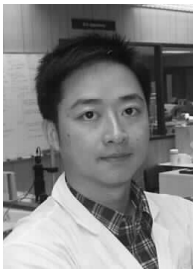
8. H. Yoda, Y. Ohuchi, Y. Taniguchi and M. Ejiri, "An automatic wafer inspection system using pipelined image processing techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**(1), 4–16 (1988).
9. W. B. Jatko, M. A. Hunt and K. W. Tobin, "Nonlinear filter derived from topological image features," *Real-Time Image Processing II, Proceedings of SPIE* **1295**, pp. 8–16 (1990).
10. P. Xie and S.-U. Guan, "A golden-template self-generating method for patterned wafer inspection," *Machine Vision and Applications* **12**(3), 149–156 (2000).
11. J.-M. Zhang, R.-M. Lin and M.-J. J. Wang, "The development of an automatic post-sawing inspection system using computer vision techniques," *Computers in Industry* **40**(1), 51–60 (1999).
12. Y. Shen and E. Y. Lam, "Simultaneous photometric correction and defect detection in semiconductor manufacturing," *Machine Vision Applications in Industrial Inspection XIV*, Vol. 6070 of *Proceedings of the SPIE*, pp. 133–142 (2006).
13. H. Lin, C.-Y. Chung and S. Chiu, "Computer-aided vision system for surface blemish detection of LED chips," *Adaptive and Natural Computing Algorithms*, Vol. 4432 of *Lecture Notes in Computer Science*, pp. 525–533 (2007).
14. R. A. Zoroofi, H. Taketani, S. Tamura, Y. Sato and K. Sekiya, "Automated inspection of IC wafer contamination," *Pattern Recognition* **34**(6), 1307–1317 (2001).
15. C.-T. Su, C.-M. Ke and T. Yang, "A neural-network approach for semiconductor wafer post-sawing inspection," *IEEE Transactions on Semiconductor Manufacturing* **15**(2), 260–266 (2002).
16. R. K. Ferrell, S. S. Gleason and K. W. Tobin, "Application of fractal encoding techniques for image segmentation," *6th International Conference on Quality Control by Artificial Vision* **5132**, pp. 69–77 (SPIE, 2003).
17. S. S. Gleason, R. K. Ferrell, T. P. Karnowski and K. W. Tobin, "Detection of semiconductor defects using a novel fractal encoding algorithm," *Design, Process Integration and Characterization for Microelectronics, Proceedings of SPIE* **4692**, pp. 61–71 (2002).
18. T. Kubota, P. Talekar, X. Ma and T. S. Sudarshan, "A nondestructive automated defect detection system for silicon carbide wafers," *Machine Vision and Applications* **16**(3), 170–176 (2005).
19. H. Zhou, A. Kassim and S. Ranganath, "A fast algorithm for detecting die extrusion defects in IC packages," *Machine Vision and Applications* **11**(1), 37–41 (1998).
20. B. Sumengen, B. S. Manjunath and C. Kenney, "Image segmentation using curve evolution and region stability," *International Conference on Pattern Recognition*, pp. 965–968 (2002).
21. J. Cheng, R. Chung, E. Y. Lam, K. S. Fung, F. Wang and W. H. Leung, "Structure-light based sensing using a single fixed fringe grating: Fringe boundary detection and 3D reconstruction," *IEEE Transactions on Electronics Packaging Manufacturing* **31**(1), 19–31 (2008).
22. E. Regentova, D. Yao, S. Latifi and J. Zheng, "Image segmentation using Ncut in the wavelet domain," *International Journal of Image and Graphics* **6**(4), 569–582 (2006).
23. N. G. Shankar and Z. W. Zhong, "Improved segmentation of semiconductor defects using area sieves," *Machine Vision and Applications* **17**(1), 1–7 (2006).
24. R. O. Duda, P. E. Hart and D. G. Stork., *Pattern Classification*, 2nd ed. (Wiley, 2001).
25. D. T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining* (Hoboken, N.J.: Wiley-Interscience, 2005).

26. S. I. Gallant, *Neural Network Learning and Expert Systems* (Cambridge, Mass.: MIT Press, 1993).
27. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics* **9**(1), 62–66 (1979).
28. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. (Prentice Hall, 2002).
29. J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6), 679–698 (1986).
30. J. Theiler and G. Gisler, "A contiguity-enhanced k-means clustering algorithm for unsupervised multispectral image segmentation," *Algorithms, Devices and Systems for Optical Information Processing, Proceedings of SPIE* **3159**, 108–118 (1997).
31. P. Montolio, A. Gasull, E. Monte, L. Torres and F. Marques, "Analysis and optimization of the k-means algorithm for remote sensing applications," *Pattern Recognition and Image Analysis, Vol. 1 of Series in Machine Perception and Artificial Intelligence*, (World Scientific, 1992), pp. 155–170.
32. R. J. Schalkoff, *Digital Image Processing and Computer Vision* (John Wiley and Sons Inc., 1989).
33. P. L. Rosin, "Measuring rectangularity," *Machine Vision and Applications* **11**(4), 191–196 (1999).
34. J. Zunic and P. L. Rosin, "Rectilinearity measurements for polygon," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(9), 1193–1200 (2003).

Ada N. Y. Ng received her Master of Philosophy degree in Electrical and Electronic Engineering from The University of Hong Kong in 2005.

Edmund Y. Lam received the BS (with distinction), MS, and PhD degrees in electrical engineering from Stanford University. He is an Associate Professor in Electrical and Electronic Engineering and Director of the Imaging Systems Laboratory at the University of Hong Kong. His research interests include computational optics and imaging, particularly their applications in semiconductor manufacturing and biomedical systems. He is also a topical editor of the Journal of the Optical Society of America A and an associate editor of the IEEE Transactions on Biomedical Circuits and Systems.

Ronald Chung received BSEE from the University of Hong Kong, Hong Kong, and PhD in computer engineering from University of Southern California, Los Angeles. He is currently with the Chinese University of Hong Kong, Hong Kong, as Director of the Computer Vision Laboratory and Professor in the Department of Mechanical and Automation Engineering. His research interests include computer vision and robotics. He is a senior member of IEEE and a member of MENSA. He was the Chairman of the IEEE Hong Kong Section Joint Chapter on Robotics and Automation Society and Control Systems Society in the years 2001–2003.



Kenneth S. M. Fung is the Technical Manager in the Research and Development Department at ASM Assembly Automation Limited. After receiving the PhD degree in 1999 from the Department of Electrical and Electronic Engineering at the University of Hong Kong, he joined ASM Assembly Automation Limited as a senior engineer. He developed a subpixel accuracy, high speed and robust computer vision alignment algorithm that was applied to all ASM products and boosted the capability and vision technology level of the company's semiconductor packaging machines. Currently, he leads a team of research engineers responsible for the projects in machine vision inspection and algorithm development. He also provides technical supervision for a team of vision application engineers who are responsible for the projects in developing machine vision applications on ASM products. Kenneth's research interests are computer vision, pattern recognition, digital image processing, and artificial neural networks.

W. H. Leung received his BSc and MSc in Electrical Engineering from the University of Hong Kong in 1982 and 1988 respectively. He is currently the Technical Director (Computer Vision) of ASM Assembly Automation Ltd. He is a member of IEEE.