Int J Adv Manuf Technol (2014) 73:1605–1614
DOI 10.1007/s00170-014-5871-y

ORIGINAL ARTICLE

Defect inspection and extraction of the mobile phone cover glass based on the principal components analysis

Di Li · Lie-Quan Liang · Wu-Jie Zhang

Received: 13 January 2014 / Accepted: 14 April 2014 / Published online: 25 May 2014

Abstract Surface defect inspection is an important part of quality control in mobile phone cover glass manufacturing. The traditional method is usually carried out manually by experienced inspectors and thus lacks sufficient efficiency and accuracy. In this paper, an automatic defect inspection system based on the principal components analysis is proposed for five typical cover glass defects: scratch, crack, deformation, edge broken, and angle cutting. This inspection system is robust for the defect shapes and obtains high recognition accuracy. The inspection system includes three parts: pre-processing, PCA-based defect recognition, and defect edge extraction. After pre-processing, most of noise and outliers are eliminated and the pixels of defects in the image are well enhanced for inspection and recognition. The eigen-defect matrix is constructed to characterize the variation between defect images. Additionally, time consumption in constructing the eigen-defect matrix is also discussed. The experimental results show that the inspection system has achieved high accuracy for inspection and recognition.

Keywords Mobile phone cover glass · Defect inspection · PCA · Machine vision · Edge extraction

D. Li · L.-Q. Liang · W.-J. Zhang
School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China

D. Li
e-mail: itdili@scut.edu.cn

W.-J. Zhang
e-mail: zwjllhtt@scut.edu.cn

L.-Q. Liang (✉)
Guangdong Province Key Lab of Electronic Commerce Market Application Technology, Guangdong University of Finance and Economics, Guangzhou 510320, China
e-mail: lianglq@gdufe.edu.cn

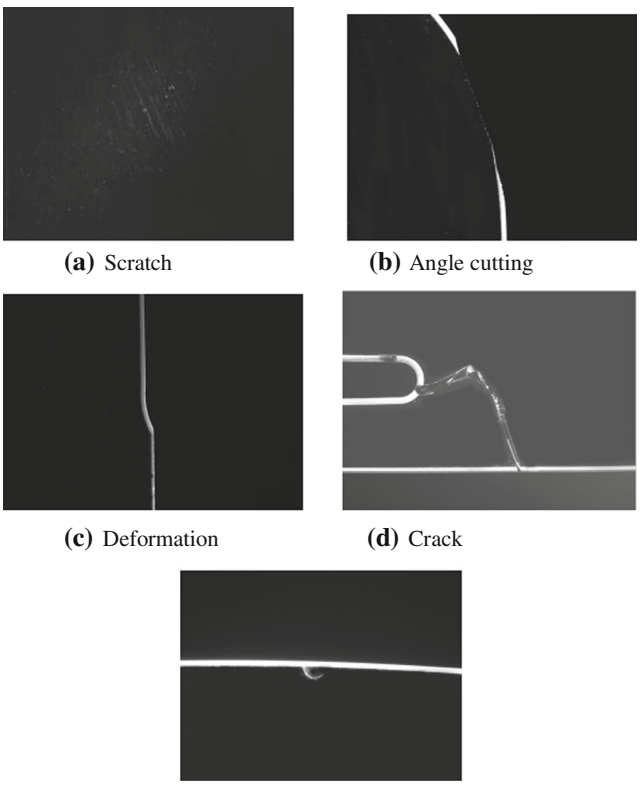
1 Introduction

The quality control concept is the most vital aspect of the glass manufacturing industry. The mobile phone cover glass have become increasingly important in recent years due to the quickly development of the mobile network and mobile phone. In order to monitor the manufacturing process stability and guarantee the display quality of the mobile phone cover glass, the inspection of defects on the cover glass becomes a critical task in manufacturing. Human visual inspection and electrical functional tests are the most commonly methods for the cover glass defect detection. However, the manual inspection is a time-consuming and tiresome task. The manual activities of inspection could also be subjective and highly dependent on the experience of human inspectors. The electrical functional test is inherently limited to offline operations and generally can only be accomplished after the fabrication of the mobile phone cover glass is complete. In this paper, an automatic visual system for the mobile phone cover glass defect inspections is proposed.

With regard to the glass industry, analyses and methodologies employed to detect the defects in the glass sheets mainly use image processing techniques because of their higher precision and speed. A number of techniques that use machine vision defect detection system have been presented in the past by some researchers. In this area, Tsai et al. proposed many valuable approaches for the defect inspection of various product surfaces. In [1], by using the singular value decomposition (SVD) of the image matrix of the LCD panel, the foreground with defects was reconstructed and segmented from the original textured image. In [2], two entropy measures of chromatic and structural regularities were proposed for the automatic defect inspection of gold-plated fingers on PCBs. Other algorithms proposed by Tsai et al. could be found in [3–5]. In addition, Shimizu [6] proposed a method for detecting foreign materials in the inspection of an LCD with its protective film

in place, without being affected by scratches or dust on the surface of the protective film. [7] presented a method for detection and recognition of bubble and non-bubble of glasses in low-resolution images whose main contribution was that a so-called binary feature histogram (BFH) was proposed to describe the characteristic of the glass defect. Then the AdaBoost method was adopted for classification. According to the image filtration based on gradient direction, downward threshold based on adaptive surface and OTSU algorithm, Peng et al. presented a distributed online defect inspection system for float glass fabrication in [8]. This distributed online defect inspection system could detect glass defects such as bubbles, lards, and optical distortion fast. In [9], Perng et al. proposed a two-phase algorithm (a pre-training phase and a testing phase) for LED defect inspection. In terms of the wavelet analysis and artificial neural network and fuzzy *k*-nearst neighbor, [10] introduced an approach of glass defect identification.

The abovementioned approaches mainly focus on one or two defect types of glass or LCD, such as bubbles or foreign materials. They are not suitable for the manufacture of the mobile phone cover glass; because in the manufacture process of the cover glass, there are more defect types, such as scratch, angle cutting, deformation, crack, and edge broken, as shown in Fig. 1. For a cover glass image, the inspection system should have two functions, i.e., defect detection and defect recognition (also known as defect classification). Because the defects of a same defect type could be of different sizes and orientation or even locations, the recognition rate would be very low if the geometry or shapes for recognition are used only. This case is similar to the face recognition. A typical approach for face recognition is that the test face would be compared with all the faces in the training set according to the face features. A well-known method for face recognition is the principal component analysis (PCA). Therefore, each glass defect image can be treated as a "defect face" and a training set can be established for feature extraction and recognition by using the PCA. Meanwhile, the advantage of this approach is that the inspection process needs not to be divided into two phases that are the defect detection and defect recognition. By using the PCA, once the inspection system finds out an equivalent image from the training set for the test image, it would be very clear that the cover glass of the test image is normal or defective. Thus, PCA can merge the defect detection and defect recognition into one process. This will greatly simplify the defect inspection process. Thus, in this paper, a global approach based on the PCA for automatic visual inspection of the mobile phone cover glass defects is proposed, which is independent of the defect type, shape, and size.



(a) Scratch **(b)** Angle cutting

(c) Deformation **(d)** Crack

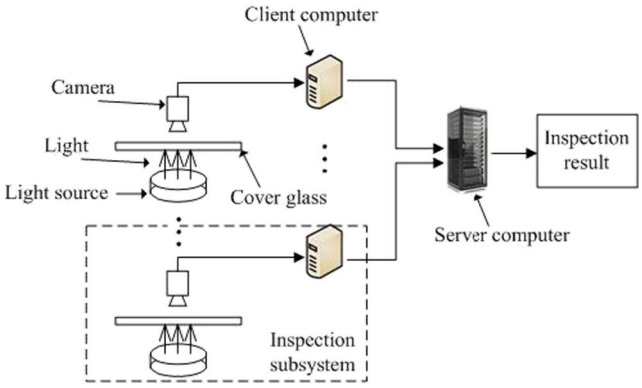
(e) Edge broken.

Fig. 1 Five different defects of the mobile phone cover glass. **a** Scratch. **b** Angle cutting. **c** Deformation. **d** Crack. **e** Edge broken

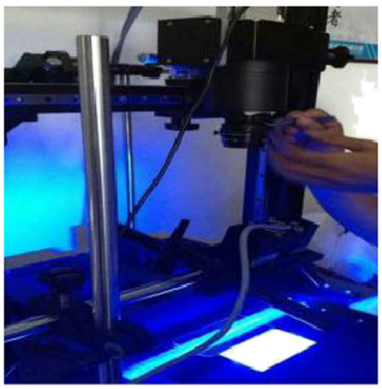
2 Introduction to the inspection system

The inspection system is designed as a distributed processing system which is constructed on server/client model. It composes of several inspection subsystems. Every subsystem includes a high-performance client computer, a plane array CMOS camera, and a light source. Every camera corresponds to a cover glass. All cameras are installed equally spaced at the same vertical line in crossing direction on product line. Suppose that the size of a cover glass is $p \times q$, $p > q$, i.e., p denotes the length of the long edges of the rectangle. All the cover glasses move the same distance of p/l (the distance is depended on the inspection accuracy requirement) from left to right every interval. Thus, for a cover glass, the camera will take l images. The field of vision of each camera covers the area of a cover glass of $(p/l) \times q$. The client computer processes images and transmits inspection results to a server. The server computer acts as the interface between human and machine. It mainly used to manage defects data offline and grade glass block based on the defect data that it received. High-speed network connecting the server and the client computer of inspection subsystems ensures the reliability of defects data transmission. The structure is shown in Fig. 2.

After images are obtained by the cameras, they will be sent to the corresponding client computer for defect inspection. As shown in Fig. 3, the flow chart of the image processing illustrates the inspection process of each inspection subsystem.



(a) Hardware structure.



(b) Picture of the subsystem.

Fig. 2 Hardware structure of distributed inspection system. **a** Hardware structure. **b** Picture of the subsystem

For each image, pre-processing is applied in the first, which includes contrast enhancement, binaryzation, and de-noising. Pre-processing could not only greatly improve the contrast of defect and background but also reduce the influence of dust, electronic noise, etc., then the images after pre-processing can be recognized by the PCA algorithm and the result of defect type is output. If the defect contained in the image cannot be recognized by the PCA algorithm, this image will be marked and inspected manually. Finally, edge extraction is applied on the recognized defect image for defect edge detection. In the next section, the details of the pre-processing, PCA recognition, and edge extraction algorithms are presented.

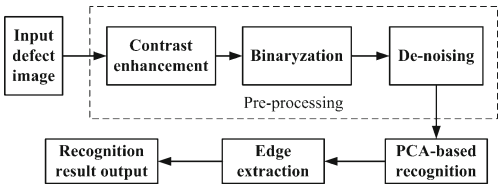


Fig. 3 Flow chart of PCA-based defect inspection subsystem

3 Defect recognition and recognition scheme

3.1 Pre-processing of the mobile phone cover glass image

Before recognition of the defect type of the obtained images by the PCA algorithm, pre-processing algorithm needs to be performed to the images, which includes three parts: contrast enhancement, binaryzation, and de-noising.

3.1.1 Contrast enhancement and binaryzation

The typical method for contrast enhancement is piecewised linear transformation, which can highlight the defect area while the background can be suppressed. To be formally, it can be expressed as follows:

$$\mathrm{Gray}(i,j)=\begin{cases}(c/a)\ f(i,j) & 0<f(i,j)<a\\ [(d-c)/(b-a)][f(i,j)-a]+c & a\le f(i,j)<b\\ [(M_2-d)/(M_1-b)][f(i,j)-b]+d & b\le f(i,j)<M_1\end{cases} \quad (1)$$

where $f(i,j)$ is the gray value at the (i,j)th pixel, a, b, c, d, M_1 and M_2 are the bounds of the piecewise linear transformation function. Eq. (1) can be represented as Fig. 4.

But for the mobile phone cover glass image, the piecewise linear transformation function is not suitable. The histogram for a typical crack defect is shown in Fig. 5.

Theoretically, the pixels of background should be black, i.e., the corresponding gray values should be zeros, but according to the histogram, the pixel numbers corresponding to the gray values less than 50 are almost zeros. For the linear function, the pixels with gray values in [0, a] are also stretched, which will confuse the background and foreground to a certain extent. Obviously, the gray values less than or equal to the peak value of the histogram should be treated as the background. The gray values corresponding to the crack defect and edges of cover glass are assembled between the peak point and

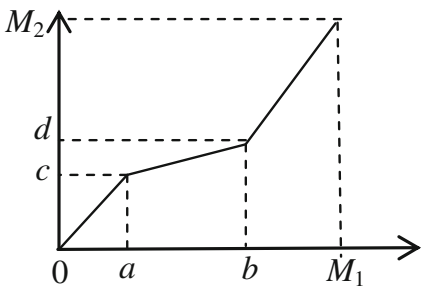


Fig. 4 Piecewise linear transformation

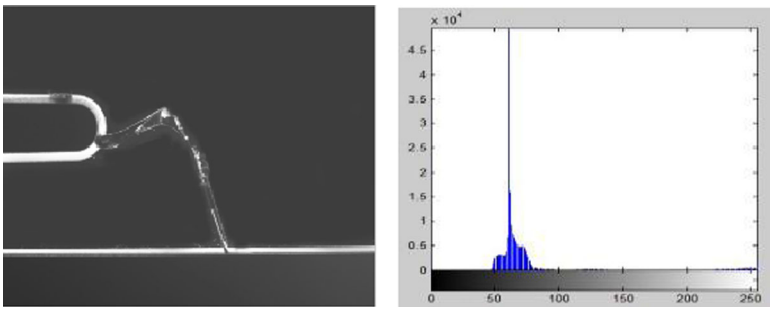


Fig. 5 Original image of crack defect and its corresponding histogram. **a** Original image of crack defect. **b** The corresponding histogram

(a) Original image of crack defect. **(b)** The corresponding histogram.

150. In order to suppress the background as well as highlight the defect, the log function is applied for the contrast enhancement in this algorithm, which is shown in the following Eq. (2) and the corresponding graph is shown in Fig. 6. By Eq. (2), it can be seen that the pixels whose gray values are lower than the threshold peak+τ (here τ is a dynamic constant) will be set to 0. The gray values in the front part of [peak+τ,255] will be greatly stretched so that the defect area can be brighter. Thus, the background can be obviously distinguished from the foreground, i.e., the defect area.

$$\text{Gray}(i,j) = \begin{cases} 0 & f(i,j) < \text{peak} + \tau \\ 40 \times \log(f(i,j) - \text{peak}) & f(i,j) \geq \text{peak} + \tau \end{cases} \quad (2)$$

Figure 7 shows the contrast enhancement result of Fig. 5a. Obviously, the defect area is highlighted, especially for the edges of the defect.

Before defect recognition, the binaryzation processing to the image should be performed after contrast enhancement. Many methods can be chosen to convert the gray image to the binary image. Here, the OTSU [8] algorithm is applied. The binaryzation result is shown in Fig. 8.

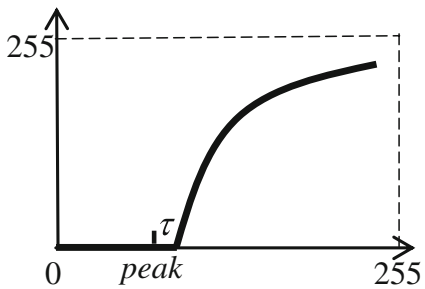


Fig. 6 Log transformation function for the contrast enhancement

3.1.2 De-noising

The image after binaryzation includes background, defect area, and noise. The noise can be caused by many factors, e.g., electronic noise and errors in analog-to-digital converter. Also, blotches and dust will affect the image quality. The noise may include one or several pixels. They may be realized as the fake defects and cause the inspection system to misoperate, so they must be eliminated before inspection results output.

The pixels in the binaryzation image only have two values, i.e., 1 or 0. The pixels with values 0 denote the black background, and the pixels with values 1 denote the defects or the noise. Since the sizes of the noise area are not known, the *blob analysis* [11, 12] is applied to the binary image. The blob analysis is a fundamental step for detection of connected components between pixels in binary images in segmentation of an image objects and regions, or *blobs*. Each blob is assigned a unique label to separate it from other blobs. All the pixels within a blob of spatially connected 1's are assigned the same label. It can be used to establish boundaries of objects, components of regions, and to count the number of blobs in an image. In the following, the multi-pass scanning procedure used for connected component labeling is presented.

Let $B_{M\times N}$ denote the 2D binaryzation image. A pixel is a background pixel if $B(i,j)=0$ and an object pixel if

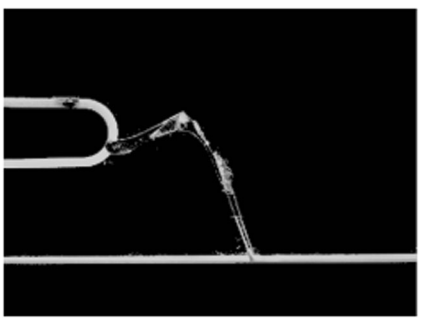


Fig. 7 Contrast enhancement result of Fig. 5a by using Eq. (2)

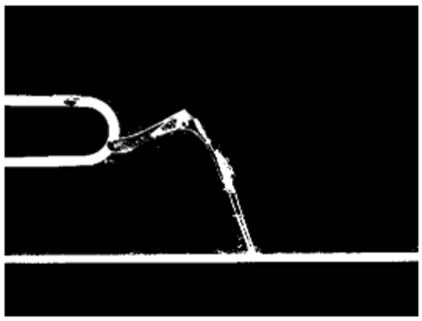


Fig. 8 Binaryzation result of Fig. 7

$B(i,j)=1$. An array L of the same size and shape of B is used for storing the labels. The problem of connected component labeling is to fill the array L with (integer) labels so that the neighboring object pixels have the same label. The current pixel $B(i,j)$ in the scanning process and the four neighbor pixels in the NE, N, NW, and W directions of $B(i,j)$ are denoted as ***a***, ***b***, ***c***, ***d***, and ***e***, respectively. Also, the same letters are used in place of their (i,j) coordinates. With these notations, $L(a)$ denotes the label of the current pixel. $B(c)$ denotes the pixel value of the neighbor in the north direction of ***a***. Let l be an integer variable initialized to 0. The multi-pass algorithm for connected component labeling can be expressed as follows:

Algorithm 1. Multi-pass algorithm for connected- component labeling

Input: Binaryzation image $B_{M\times N}$;

Output: L.

Initialization: $l=0$, $L=O_{M\times N}$, where $O_{M\times N}$ is an $M\times N$ zero matrix.

Step 1. In the first pass through the image, scanning the image pixel by pixel from left to right and top to bottom. The assignment of a provisional label for ***a*** during the first scan can be expressed as:

$$L(a)\leftarrow\begin{cases}0, & B(a)=0\\ l,(l\leftarrow l+1), & B(i)=0,\forall i\in(b,c,d,e)\\ \min\limits_{i|B(i)=1}(L(i)), & otherwise,\forall i\in(b,c,d,e)\end{cases}.$$

Step 2. If $B(a)=1$ and $B(i)=1, i\in(b,c,d,e)$
$L(i)|_{B(i)=1,i\in(b,c,d,e)}\leftarrow L(a)$

Step 3. After the first pass, return to the Step 2 and begin the second pass through the image. Repeat this scan procedure until the label array L no longer changes.

In the first pass through the image, the above algorithm states that $L(a)$ is assigned to 0 if $B(a)=0$. It is assigned a new label l, and l is increased by 1, if its four neighbors in the NE, N, NW, and W directions are all background pixels. Otherwise, it is assigned the minimum of the provisional labels already assigned to a neighbor in the four neighbors. Also, the labels in the four neighbors should be re-assigned to the minimum among them. In the later passes, labels for the current pixel and their four neighbors are modified to be the minimum labels of the provisional labels among these five pixels. In this way, after several passes, all pixels in a connected component will receive the smallest provisional label assigned to the pixels when L no longer changes.

With the label array L, de-noise will be an easy task. The elements with a same label in L belong to a same noise area (or defect). The area of the kth noise area (defect), denoted as $C(k)$, can be calculated by counting the element number of L with element value equals k. If $C(k)<\tau$, then the pixels belong to the kth noise area are set to zeros. Here, τ is a threshold for de-noising. The label array L is not only used for de-noising but also used for edge extraction, which will be presented in Section 3.3.

3.2 PCA for cover glass defects inspection

After the abovementioned pre-processing, most of the noise and outliers are eliminated. The defect pixels in the image are well enhanced for inspection and recognition. The PCA for recognition will be introduced. After recognition, the boundary following algorithm will be applied for the defect edge extraction.

PCA is a technique for dimensionality reduction in computer vision [13, 14]. The PCA techniques, as known as Karhunen-Loeve methods, choose a dimensionality reducing linear projection that maximizes the scatter of all projected samples. More formally, consider a set of N images $X=\{x_1,x_2,\cdots,x_N\}$, $x_N\in R^{n^2}$ taking values in an n^2-dimensional image space, and assume that each image belongs to one of c classes. Also consider a linear transformation mapping the original n^2-dimensional image space into an m-dimensional feature space, where $m<n^2$. The new *feature vectors* $y_k\in R^m$ are defined by the following linear transformation:

$$y_k=W^Tx_k,\quad k=1,2,\cdots,N \tag{3}$$

where $W\in R^{n^2\times m}$ is a matrix with orthonormal columns.

If the total scatter matrix S is defined as follows:

$$S=\sum_{k=1}^{N}(x_k-\mu)(x_k-\mu)^T\in R^{n^2\times n^2} \tag{4}$$

Here, $\mu \in R^{n^2}$ is the mean image of all images, then after applying the linear transformation W^T, the scatter of the transformed feature vectors $\{y_1, y_2, \cdots y_N\}$ is $W^T SW$. In PCA, the projection W_{opt} is chosen to maximize the determinant of the total scatter matrix of the projected samples, i.e.,

$$W_{opt} = \arg\max_{W} \left| W^T SW \right| = [\omega_1, \omega_2, \cdots, \omega_m] \tag{5}$$

where $\{\omega_i | i=1,2,\cdots,m\}$ is the set of n^2-dimensional eigenvectors of S corresponding to the m largest eigenvalues. Since these eigenvectors have the same dimension as the original images, they are referred to as *eigen-defects* in the algorithm.

Much of the previous work on automated defect recognition has ignored the issue of just what aspects of the defect stimulus are important for identification. The main features used in the previous works are the shapes or structures of defects. This suggested that an information theory approach of coding and decoding defect images may give insight into the information content of defect images, emphasizing the significant local and global "features." The relevant information in a defect image could be extracted, and encoded as efficiently as possible, and one defect encoding could be compared with a database of models encoded similarly. A simple approach to extract the information contained in an image of a defect is to somehow capture the variation in a collection of defect images, independent of any judgment of features, and use this information to encode and compare individual defect images.

In the inspection system, the principal components of the distribution of defects or the eigenvectors of the scatter matrix of the set of defect image set could be found. These eigenvectors can be thought of as a set of features which together characterize the variation between defect images. Thus, a training set is constructed, which is used for calculating the eigen-defects. Every image in the training set is labeled with its defect type, i.e., scratch, crack, deformation, edge broken, or angle cutting. Each defect image in the training set can be represented exactly in terms of a linear combination of the eigen-defects. The number of possible eigen-defects equals to the number of defect images in the training set. However, the defects can also be approximated using only the "best" eigen-defects, i.e., those who have the largest eigenvalues, and which therefore account for the most variance within the set of defect images. The primary reason for using fewer eigen-defects is computational efficiency. The best M eigen-defects span an M-dimensional subspace—"defect space"—of all possible images. The eigen-defects are the basis vectors of the eigen-defect decomposition.

According to the above analysis, the PCA-based recognition algorithm for cell phone cover glass defect is described as follows:

Algorithm 2. PCA based recognition algorithm for the mobile phone cover glass defect.

Input: The training set defect images $X = \{x_1, x_2, \cdots, x_N\}$, $x_k \in R^{n^2}$; The test defect image Γ.

Output: The recognition result of Γ.

Step 1. Compute the average defect of the training set $\mu = \frac{1}{N}\sum_{k=1}^{N} x_k$, $x_k \in R^{n^2}$.

Step 2. Compute the difference between each defect image in the training set and the average defect $D_k = x_k - \mu$, where $D = \{D_1, D_2, \cdots, D_N\} \in R^{n^2 \times N}$, $k = 1,2,\cdots,N$.

Step 3. Compute the eigenvalues and orthonormal eigenvectors of $D^T D \in R^{N \times N}$. The diagonal eigenvalue matrix is denoted as $\Lambda = diag\{\sigma_1, \sigma_2, \cdots, \sigma_N\}$ and the eigenvector matrix is denoted as $W = [\omega_1, \omega_2, \cdots, \omega_N]$.

Step 4. Eliminate the eigenvalues and the corresponding eigenvectors satisfied $\sigma_k < \theta, \quad k = 1,2,\cdots,N$, where θ is a threshold. The remain eigenvectors are collected as the eigen-defect matrix $\tilde{W}$.

Step 5. Project the defect images in the training set into the defect space $\tilde{D} = D\tilde{W}$.

Step 6. Project the test defect image into the defect space by $\tilde{\Gamma} = \tilde{W}^T(\Gamma - \mu)$.

Step 7. Compute the minimal norm distances of $\tilde{\Gamma}$ and $\tilde{D}_k$. Suppose that $K = \arg\min_k \| \tilde{\Gamma} - \tilde{D}_k \|_2 \quad k = 1,2,\cdots,N$. If $\min_k \| \tilde{\Gamma} - \tilde{D}_k \|_2 < \theta$, then the defect type of Γ is classified as that of x_K. Otherwise, output the image of Γ for manual recognition and go to the Step 9.

Step 8. Output the test image Γ and the matching image x_K for comparison. Also, the defect type of Γ is output.

Step 9. If the number of images in the training set is less than a pre-setting integer number T, add Γ into the training set and labeled by the corresponding defect type.

According to Eq. (4), the eigenvalues and eigenvectors of the $n^2 \times n^2$ scatter matrix S in the step 3 of algorithm 2 could be computed. Based on the linear algebra theory, for a $p \times q$ matrix, the maximum number of non-zero eigenvalues that the matrix can have is $\min(p,q)$. Since the number of training images (N) is usually less than the number of pixels (n^2), the most non-zero eigenvalues that can be found are equal to N.

So eigenvalues of $D^TD \in R^{N\times N}$ instead of $DD^T \in R^{n^2\times n^2}$ in the step 3 can be calculated. It is clear that the dimensions of DD^T is much larger than that of D^TD. So, the dimensionality will decrease.

In the algorithm 2, when the image number of the training set is less than a pre-setting integer number T, every test image will be added into the training set after recognition. This will reduce the inspection error. For a same defect type, the defect shape of one cover glass may differ from another, but they belong to a same defect type so that the new defect image can be represented by the linear combination of the eigen-defects. Thus, this approach is robust for the defect shapes and structures.

3.3 Edge extraction by boundary following algorithm

The final step of this inspection system is the edge extraction of the defect. A great diversity of edge detection algorithms have been devised with differences in their mathematical and algorithmic properties such as Roberts, Sobel, Prewitt, Laplacian, and Canny, all of which are based on the difference of gray levels [15]. In this inspection system, because the label array L of an image has been constructed in Section 3.1, the edge extraction will be simplified by using the boundary following algorithm.

Algorithm 3. Boundary following algorithm for edge extraction

Input: Original gray defect image Γ
Label array L of Γ obtained by the Algorithm 1

Output: The defect image with defect edges.

Initialization: $l = 1$.

Step 1. Search in L from left to right and from bottom to top, find the first element of L satisfies $L(i, j) = l$.

Step 2. Start from $L(i, j)$, define the initial searching orientation as upper left. If the value of upper left point is the same with $L(i, j)$, i.e., $L(i-1, j-1) = L(i, j)$, then (i, j) is the next edge point. Otherwise, the searching orientation turns 45 degree clockwise until a new point with a same value of $L(i, j)$ is found. Set this new point as the new edge point.

Step 3. Start from the new edge point, turn the current searching orientation 90 degree count-clockwise and use the same method as descript in Step 2 for searching the next edge point until returning to the original start point $L(i, j)$.

Step 4. Set the gray values of the corresponding edge points in the original gray defect image as 255.

Step 5. $l = l + 1$, return to the Step 1.

Figure 9 gives a demonstration of the boundary following process. The numbers labeled in Fig. 9 denote the sequence of searching orientations. The black points denote the edge pixels. According to the above pre-processing, recognition, and edge extraction, the defects of a new cover glass can be inspected. Also, the size of each defect can be calculated by simply counting the number of pixels in one defect.

4 Experiment results

To assess the viability of the PCA-based defect inspection system, experiments with stored defect images are performed. The first experiment is performed to test the recognition rate of this system. In this experiment, the total 100 defect images (there are no normal/defect-free images in these 100 images) belonging to five defect types (scratch, crack, deformation, edge broken and angle cutting) are used as the candidate set. The gray image size is 320×200. Two image sets are constructed: a training set and a test set. The computer used in the experiment is ThinkPad X61 with Duo 2G CPU and 2G memory. In each experiment, 10 defect images are randomly selected from the candidate set as the test images.

In the PCA-based recognition algorithm, the construction of *eigen-defect matrix* $\widetilde{W}$ in step 1 to step 5 of algorithm 2 is time consuming. For example, when the defect image number of the training set is 20, the time for the construction of $\widetilde{W}$ is 0.297 s, but the time for recognition of a defect image is only 0.05 s. When the image number of training set increased to 60, the time for the construction of $\widetilde{W}$ is 2.8 s and the time for recognizing an image is 0.53 s.

In order to improve the real-time performance of the system, the grouping strategy is adopted, i.e., the image number in the training set was initialized as 20. They are randomly selected from the 100 defect images (these 20 defect images

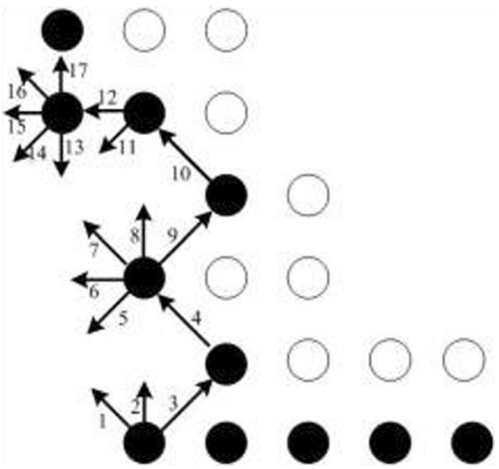


Fig. 9 A demo for algorithm 3

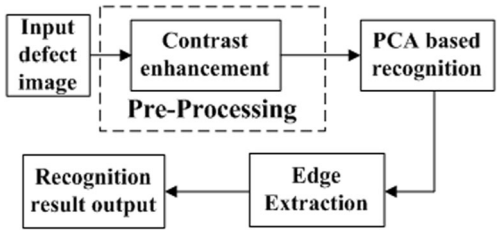


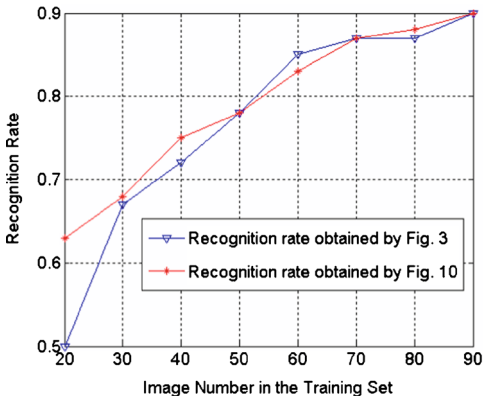
Fig. 10 Flow chart of PCA-based defect inspection subsystem removed binaryzation and de-noising

must be recognized manually and they should be removed from the candidate set to the training set). Then 10 images of the candidate set are randomly selected from the remaining 80 images and moved into the test set. By using the 20 images, the eigen-defect matrix $\widetilde{W}$ can be constructed. Then the 10 images in the test set are projected into the defect space and recognized by steps 6 to 9 of algorithm 2. After finishing the recognition of these 10 images, they will be moved into the training set, and the total 30 images in the training set are used as the training images for the recognition of next 10 new images. In the next experiment, 10 images are selected from the remaining 70 images in the candidate set. The eigen-defect matrix $\widetilde{W}$ should be reconstructed by the 30 images in the training set. According to this procedure, 100 defect images in the candidate set are divided into eight groups. The image numbers in the training set and test set of each group are {20, 10}, {30, 10}, {40, 10}, {50, 10}, {60, 10}, {70, 10}, {80, 10}, and {90, 10}, respectively. The advantage for grouping is that for every 10 test images (in each group), an eigen-defect matrix $\widetilde{W}$ only needs to be constructed once. This will significantly reduce the average running time for recognition such that the recognition system is efficiency and satisfying the real-time requirement of the assembly line.

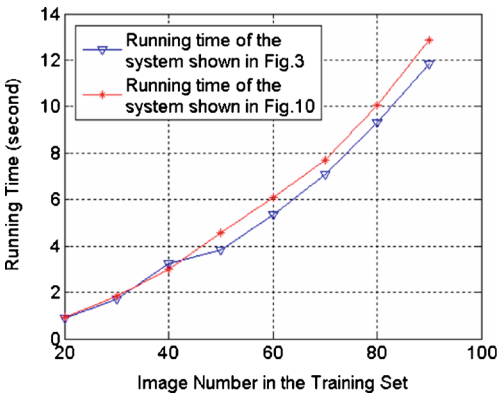
Here, the recognition accuracy of the inspection system will be tested. Thus, in this experiment, a defect image could be considered as being *recognized correctly* if and only if a test defect image is recognized as its actual defect type. For example, if a crack defect image is input and a crack defect image is output as the equivalent image, then this defect image is recognized correctly. Otherwise, the recognition of this defect image is counted as *failure* (the output equivalent image's defect type is not the actual defect type of the input image). In addition, it's well known that from the PCA-based face recognition, the recognition results are relied on the segmentation of the image. As shown in Fig. 3, the segmentation of this inspection system occurred in the binaryzation and de-noising. Thus, another experiment is made as shown in Fig. 10, which removed the steps of binaryzation and de-noising. The recognition rates obtained by Figs. 3 and 10 are compared as Fig. 11.

For each experiment, 10 independent trials of the experiment are performed and the average recognition rate is calculated while the selection of images from the 100 images is random. The recognition rates of the eight groups are shown in Fig. 11a. Furthermore, the average running time of each group of the test images is calculated. The relationship between the running time for each group (10 test images) and the image numbers of the training set is depicted in Fig. 11b.

According to Fig. 11a, the influence of segmentation (bianryzation and de-noising) is not significant for the PCA algorithm in this application. This is because that during the process of the glass manufacturing, the defect inspection of cover glass is processed in the dust-free workshop after cleaned by ultrasonic washing. Thus, the images obtained by the camera have a related clean background. Thence, the cover glass has no textures. These will be very helpful for the defect inspection and recognition. In Fig. 11a, the recognition rates obtained by Figs. 3 and 10 are comparable, but binary images are more convenient for storage and transmission. From Fig. 11b, the running time of the system shown in Fig. 3 is less than the system shown in Fig. 10. Moreover, binaryzation is a necessary step for blob analysis and edge extraction. Thus, the system shown in Fig. 3 is applied for defect inspection.



(a) Recognition rate



(b) Average running time for each set of test images

Fig. 11 Recognition rate and running time. **a** Recognition rate. **b** Average running time for each set of test images

(a) Test image **(b)** Equivalent image

Fig. 12 Recognition result for a scratch defect. **a** Test image. **b** Equivalent image

Another experiment is performed for testing the defect detection rate. Here, the test outcome is defined as *positive* if an image is predicted as a defect image, and the test outcome is defined as *negative* if an image is predicted as a normal image without defects (defect-free). It should be noted that in this experiment, the output equivalent image's defect type can be different from the actual defect type of the input defect image. Because although the recognition result of the defect type is wrong, the input defect image still recognized as defective and the corresponding cover glass cannot be considered as an acceptable product. The *true positive*, *false positive*, *true negative*, and *false negative* can be defined as follows:

True positive (TP) a defect image correctly classified as defective.
False positive (FP) a normal image incorrectly identified as defective.
True negative (TN) a normal image correctly identified as normal.
False negative (FN) a defect image incorrectly identified as normal.

According the above definitions, the true positive rate (TPR, namely *sensitivity*) can be defined as follows:

$$\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) = \mathrm{TP}/\mathrm{P}.$$

The false positive rate (FPR) is defined as follows:

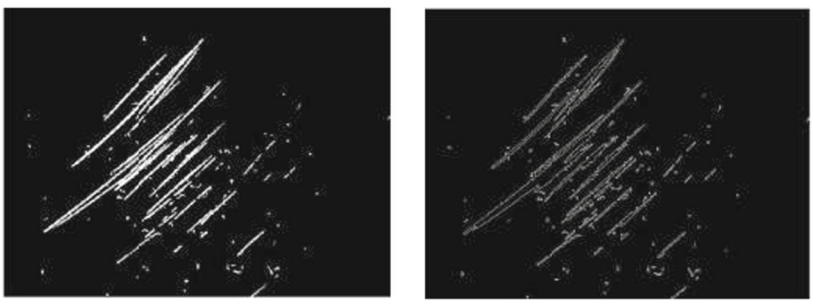
$$\mathrm{FPR} = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN}) = \mathrm{FP}/\mathrm{N},$$

where N denotes the actual normal image number and P denotes the actual defect image number.

In this experiment, 60 images (the 60 images consisted by the normal images and five types of defect images) are used in the training set and 100 images are used in the testing set. There are 50 normal cover glass images and 50 defect images consisted by the five types of defect images in the 100 test images. Thus, P=N=50. Finally, 44 defect images are correctly detected as defective, and only three normal images are incorrectly recognized as "scratch" images. The experiment results are as follows:

$$\mathrm{TP} = 44;\ \mathrm{FN} = 6;\ \mathrm{FP} = 3;\ \mathrm{TN} = 47.\ \text{Thus},$$
$$\mathrm{TPR} = \mathrm{TP}/P = 0.88 \text{ and } \mathrm{FPR} = \mathrm{FP}/N = 0.06.$$

It can be seen that a high TPR and a very low FPR are obtained. This is because that the normal cover glass image is almost all black and obviously different from the defect images. The reason for the incorrectly recognition of the three normal images is that there is some noise caused by the uneven illumination of the cover glass in the normal images. After pre-processing, the noise is treated as scratches and leads to incorrectly recognition.



(a) Image after pre-processing **(b)** Edge extraction result by Alg.

Fig. 13 Edge extraction result for a scratch defect. **a** Image after pre-processing. **b** Edge extraction result by algorithm 3

Figure 12 is a recognition result. Although the test image and the equivalent image picked from the training set are different in the scratch shapes and number, both of them belong to a same defect type. In the experiment, there are not quite similar scratch images as shown in Fig. 12a in the training set. There are only a few scratch images that are similar to the image of Fig. 11b.

After recognition, algorithm 3 is applied for edge detection. Figure 13 shows the edge detection result.

5 Conclusions

In this paper, the PCA-based defect inspection system is proposed for the mobile phone cover glass. The inspection system can not only recognize several defect types simultaneously but also recognize the defect type robustly different from others based on the defect shapes or structures. By utilizing the eigen-defects of the defect images, an eigen space for all possible defect images can be spanned. For a new test defect image, it is projected into the eigen space and compared with other projections of the known defect images. Also, the boundary following algorithm is used for the edge extraction of the recognized defect images. Experiment results show that this defect inspection system has achieved good performance for the test defect images.

References

1. Lu CJ, Tsai DM (2005) Automatic defect inspection for LCDs using singular value decomposition. Int J Adv Manuf Technol 25:53–61
2. Tsai DM, Lin BT (2002) Defect detection of gold-plated surfaces on PCBs using entropy measures. Int J Adv Manuf Technol 20:420–428
3. Tsai DM, Lin CP (2002) Fast defect detection in textured surfaces using 1D Gabor filters. Int J Adv Manuf Technol 20:664–675
4. Tsai DM, Chiang IY, Tsai YH (2012) A shift-tolerant dissimilarity measure for surface defect detection. IEEE Trans Ind Inform 8:128–137
5. Tsai DM, Chen MC, Li WC, Chiu WY (2012) A fast regularity measure for surface defect detection. Mach Vis Appl 23:869–886
6. Shimizu M, Ishii A, Nishimura T (2000) Detection of foreign material included in LCD panels. IECON 2:836–841
7. Zhao J, Kong QJ, Zhao X, Liu JP, Liu YC (2011) A method for detection and classification of glass defects in low resolution images. Image and Graphics (ICIG), 2011 Sixth International Conference on Image and Graphics, 2011: 642–647
8. Peng X, Chen Y, Yu W, Zhou Z, Sun G (2008) An online defects inspection method for float glass fabrication based on machine vision. Int J Adv Manuf Technol 39:1180–1189
9. Perng DB, Liu HW, Chang CC (2011) Automated SMD LED inspection using machine vision. Int J Adv Manuf Technol 57:1065–1077
10. Liu H, Chen Y, Peng X, Xie J (2011) A classification method of glass defect based on multiresolution and information fusion. Int J Adv Manuf Technol 56:1079–1090
11. Park JM, Looney CG, Chen HC (2000) Fast connected component labeling algorithm using a divide and conquer technique. Comput Appl 2000:373–376
12. He L, Chao Y, Suzuki K, Wu K (2009) Fast connected-component labeling. Pattern Recogn 42:1977–1987
13. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. Pattern Anal Mach Intell, IEEE Trans Pattern Anal Mach Intell 19:711–720
14. Turk MA, Pentland AP (1991) Face recognition using eigenfaces. Comput Vis Pattern Recognit CVPR 1991:586–591
15. Parker JR (2010) Algorithms for image processing and computer vision. Wiley, New York