2011 Sixth International Conference on Image and Graphics

A Method for Detection and Classification of Glass Defects in Low Resolution Images

Jie Zhao, Qing-Jie Kong*, Xu Zhao, Jiapeng Liu, Yuncai Liu
Department of Automation and Key Laboratory of China MOE for System Control and Information Processing
Shanghai Jiao Tong University
Shanghai, China
qjkong@sjtu.edu.cn

***Abstract*—This paper presents a novel method for detection and recognition of glass defects in low resolution images. First, the defect region is located by the method of Canny edge detection, and thus the smallest connected region (rectangle) can be found. Then, the binary information of the core region can be obtained based on a specific filter. After noises are removed, a novel Binary Feature Histogram (BFH) is proposed to describe the characteristic of the glass defect. Finally, the AdaBoost method is adopted for classification. The classifiers are designed based on BFH. Experiments with 800 bubble images and 240 non-bubble images prove that the proposed method is effective and efficient for recognition of glass defects, such as bubbles and inclusions.**

***Keywords*-defect detection and recognition;glass inspection; low resolution; computer vision**

I. Introduction

In the production process of glass, some defects, such as bubbles and inclusions, will be produced due to the existing production technique. Different defects will make different impacts on the application of glass productions. For example, bubbles may have little effect on the ordinary household glass, but large effect on the car safety glass. Therefore, it is necessary to recognize bubbles and other defects. In order to reduce the costs and losses caused by manual inspection and increase quality of final products, applying computer vision technology to detect and recognize the defects has become a trend. Glass can be easily inspected using computer vision due to its no pattern, monochrome and transparency. However, resolution of defect images is usually very low, which means the area sometimes only has 25 pixels, as Fig. 1 shows. This fact makes it not easy to extract effective features for the recognition. In order to solve this problem, this paper presents a method to specially detect and classify the small defects (the area is around 25-100 pixels) in the low resolution images. In this method, according to the quantities of sample images obtained from the glass manufacture plant, two reasonable assumptions can be made: a) the bubble defects are internally empty; b) the stone or other inclusion (non-bubble) defects are internally filled. Then we propose a new Binary Feature Histogram (BFH) to describe the main characteristic of the defects area and work out the criteria to classify the defects in the AdaBoost algorithm. In the experiment section, besides discussing the accuracy and operating speed experiments of the proposed algorithm, a series of comparison experiments between the proposed features and the classic Local Binary Patterns (LBP) feature are systematically analyzed as well.

This paper is organized as follows: Section 2 first reviews the related works on this problem; then, Section 3 discusses the whole framework of the proposed method; after that, the proposed algorithms of foreground detection and feature extraction are described in Section 4 detailedly; the adopted classification method based on the AdaBoost is introduced in Section 5; Section 6 discusses the accuracy experiments with large number of real image data, and the comparison experiments with the Local Binary Patterns (LBP); Section 7 comes the conclusion and future work.

II. Related works

Researchers have done some works to find the defect areas and compute the total number of the defects, such as bubbles or blisters. Zhou and Liu [1] proposed a method

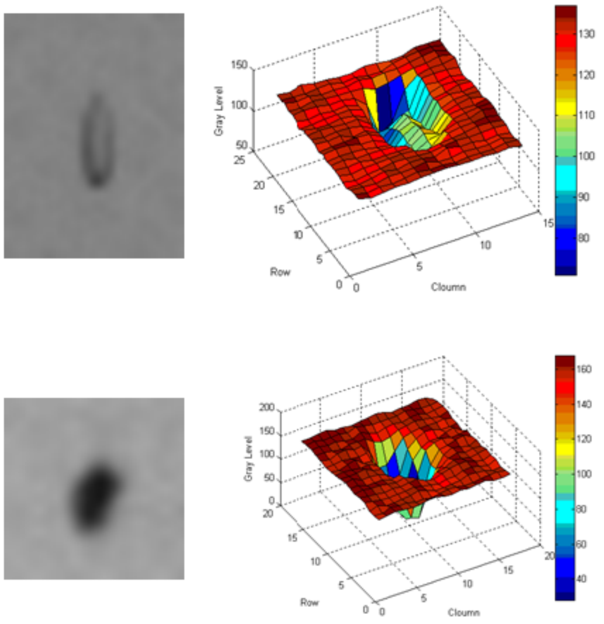


Figure 1. Images of glass defects and their three dimension figures. Above: bubble; Below: non-bubble.

978-0-7695-4541-7/11 $26.00 © 2011 IEEE
DOI 10.1109/ICIG.2011.187

IEEE computer society

on binarization based on the gray intensity graph. They computed the square error image and converted the gray image to the binary image to decide the local region of the bubbles located in the glass image. In all of the local regions, the illumination was regarded as uniform. Then, in the local region, they applied the threshold method into the binary square error image to keep the brighter object, so as to extract the bubbles. In Peng et al.'s work [2], the downward threshold based on the adaptive surface was used to remove the background composed of the stripes and strengthen the defect features. The distortion part and core part of the defects were obtained through the fixed threshold and the OTSU algorithms with gray range restricted, respectively. Only a few researchers focused on recognition of the defects. For example, Han et al. [3] classified two types of the blister (circle and ellipse) based on the information, including area, boundary, length, position of blister, etc., which were extracted from the binary images. Hu and Wei [4]proposed the wavelet packet decomposition algorithm to solve the similar problem. The characteristics, such as the core area of defects, the ratio of length and width, mean of gray, standard deviation and circularity, were used to distinguish different types of the defects. The recognition ratio reached more than 93.3% in the experiment with 150 samples for training the SVM classifier and 60 samples for testing.

LBP, which was proposed by Ojala et al. [5] , is a sort of classic feature used for object recognition. This feature is also useful to solve the problem we are studying in this paper. More details about LBP and the comparison results with the proposed BFH are given in Section 6.

III. Framework of the proposed method

The flow chart of the whole process is shown in Fig. 2. First, the defect region is segmented based on the method of Canny edge detection. Thus, the smallest connected region (rectangle) can be located. Then the binary information of the core region can be obtained based on a specific filter and the w-shaped feature. Meanwhile, noises are removed by a simple method. After that, we propose the BFH feature to describe the main characteristic of the glass defects.

IV. Algorithms of foreground detection and feature extraction

This section describes the algorithms of object foreground detection and feature extraction. The whole algorithm consists of four parts, which are foreground segmentation, binarization, noise removal, and feature extraction.

A. Foreground segmentation

As Fig. 1 shows, the defect region only occupies a quite small part of the image, compared to the whole glass material. Segmenting the foreground region beforehand and performing the processing algorithms directly to the foreground can greatly increase the efficiency of the whole algorithm. Therefore, for the first step, we need to do the foreground segmentation to every object image.

Fortunately, in the object images, the difference between the foreground (the defect region) and background is relatively obvious, due to the glass material. Thus, using the method of Canny edge detection to find the defect region can obtain satisfactory results. After the smallest connected region (rectangle) is located, we can synchronously obtain the width and height of the target area: the distance between the left-most point and right-most point is taken as the width of the defect; the distance between the top-most point and bottom-most point is taken as the height.

B. Binarization

Before extracting the features, we need to perform the binarization processing to the obtained foregrounds. Several methods can be chosen to convert the gray image to the binary image, such as the OTSU [2]and adaptive thresholding. However we do not choose the traditional methods but propose a novel binarization method in this work. The reason lies in that in the low resolution images, the binary results obtained by the traditional methods cannot work as well as that of the proposed novel method in the following task of feature extracting.

For every pixel $p(i,j)$, first calculate its binary value $b(i,j)$.

$$b(i,j) = \begin{cases} 1,\text{if } p(i,j) \geq T(i,j) \text{ or w-feature exists} \\ 0,\text{if } p(i,j) < T(i,j) \text{ and no w-feature exists} \end{cases} \quad (1)$$

where $p(i,j)$ is the gray value of pixel (i,j). $T(i,j)$ can be

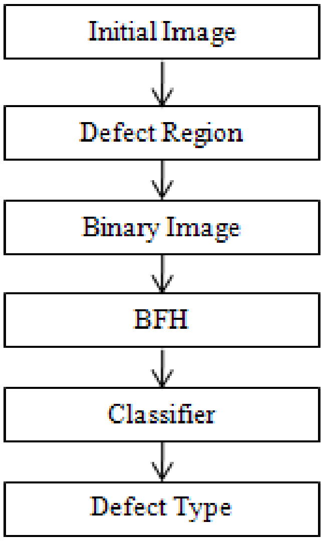


Figure 2. Flow chart of the whole algorithm.

calculated as follows.

$$T(i,j) = \sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} p(m,n)\omega(m,n) \qquad (2)$$

where $\omega(m,n)$is the weight of pixel (m,n). The factor matrix is $\frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$ (labeled as mask A).

For every pixel p(i,j), existence of the w-shaped feature is decided by the conditions of Equation (3) and (4).

$$((F_1 > 2\&F_2 > 2)\|(F_3 > 2\&F_4 > 2)) = 1 \qquad (3)$$

$$((F_1 > 1\&F_2 > 1)\&(F_3 > 1\&F_4 > 1)) = 1 \qquad (4)$$

where
$F_1 = p(i,j) - p(i-1,j)$
$F_2 = p(i,j) - p(i+1,j)$
$F_3 = p(i,j) - p(i,j-1)$
$F_4 = p(i,j) - p(i,j+1)$

If the conditions of Equation (3) and (4) are both satisfied, then the w-shaped feature of pixel $p(i,j)$ will be regard as existence.

It should be noted that the mask decides the results of the binarization. If we choose another mask, the binarization results will be changed. Two examples can be seen in Fig. 3, where mask B's factor matrix is $\frac{1}{32}\begin{pmatrix} 3 & 3 & 3 \\ 3 & 8 & 3 \\ 3 & 3 & 3 \end{pmatrix}$. The value of the mask matrix is decided beforehand by supervising learning in the training sets.

C. Noise removal

Because the defect region is small, we propose the following method to remove noises.

For every pixel whose $b(i,j)$ is equal to 0, adjust the $b(i,j)$ according to Equation (5).

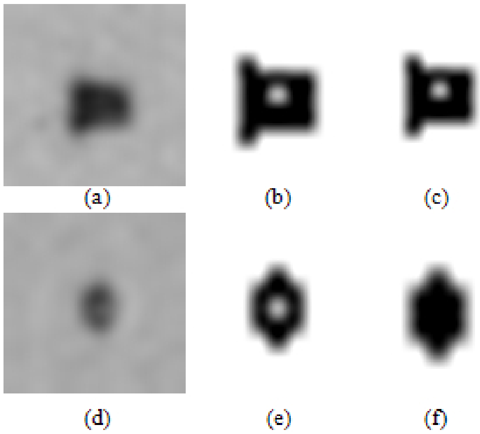


Figure 3. Examples of binary image (Left: origin image, middle: binary image based on mask A, right: binary image based on mask B.)

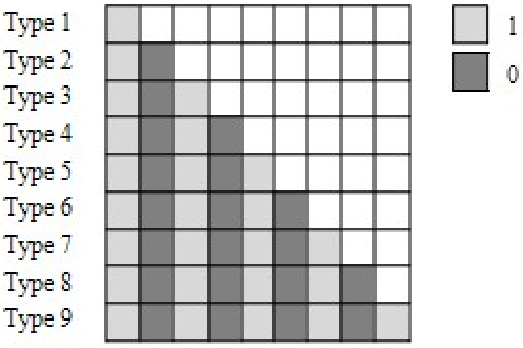


Figure 4. Nine types of feature sequence.

$$b(i,j) = \begin{cases} 1,\text{if } \sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} b(m,n) - b(i,j) > 2 \\ 0,\text{otherwise} \end{cases} \qquad (5)$$

D. Feature extraction

The extracting methods of the BFH are presented in this section.

According to the length of the binary sequences, we define nine types of binary feature sequences (shown in Fig. 4),and use L_1,...,L_9 to record the times that the nine types of feature sequences occur in the defect area.

For every row and column, extract the binary feature sequence L by Algorithm 1.

Algorithm 1 Binary feature histogram.

```
1) The first code of $L$ is equal to the binary value of the
   first pixel in this row (or column).$L[1] = b(i,1)$
   Set $k = 1; j = 2$.
2) Read the value of the $j_{th}$ pixel in $i_{th}$ row, $b(i,j)$.
   If $b(i,j)$ is not equal to $L[k]$, update the $L$. $L[k+1] = b(i,j)$ , set $k := k+1$;
   Otherwise, Set $j := j+1$, go to step 2 until finish
   extracting the binary feature sequence of this row.
3) According to the length of $L$ (labeled as $l$.If $l > 9$,
   set $l$=9.), update the value of $L_l$, $L_l = L_l + 1$.
4) Go to step 1 until all of rows and columns are scanned.
5) Normalizing the $L_l$ with Equation (6).
   $$L_l = \frac{L_l}{\sum_{l=1}^{9} L_l} \qquad (6)$$
```

Two examples of the BFH feature are shown in Fig. 5. The sequence feature value of the two samples are (3,0,12,0,7,0,0,0,0) and (4,0,12,0,0,0,0,0,0). Then we can work out the samples' BFHs, as Fig. 5(e)(f) shows.

Finally, let x be the feature vector of every sample for classification. It can be written as:

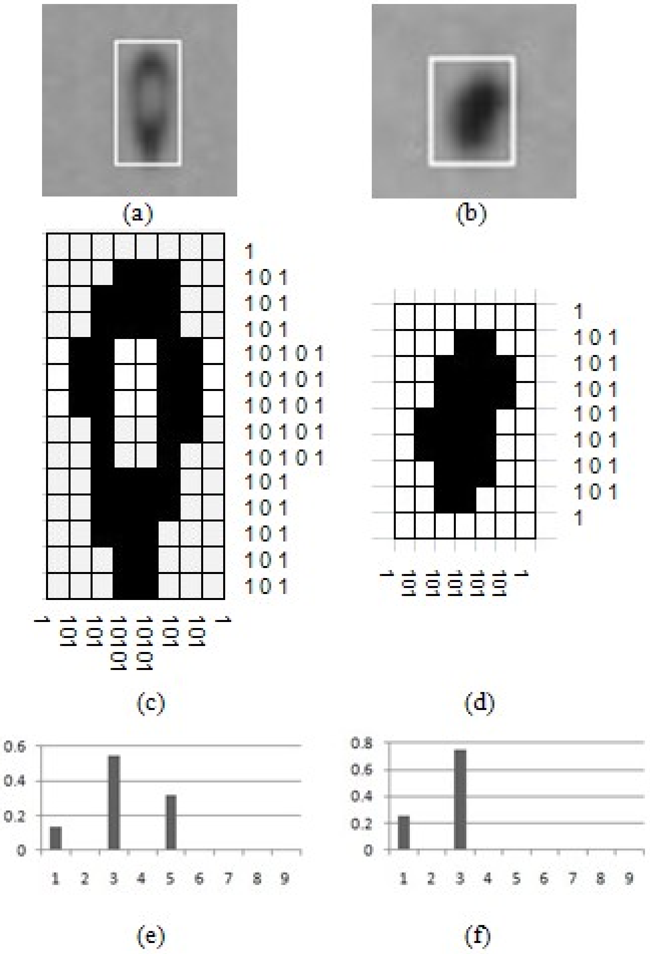


Figure 5. Examples of the BFH feature. Left: bubble, Right: non-bubble. (a) and (b) are the results of defect region detection; (c) and (d) are the binary defect region and binary sequences of every row and column; (e) and (f) are the samples' BFHs.

$$x = \{L_1, L_2, ..., L_9\} \tag{7}$$

V. Classifiers

This section describes an algorithm for learning a bubble detector. The learning procedure based on the AdaBoost algorithm that was proposed by Freund and Schapire [6].

The learning algorithm is described in Algorithm 2.

Algorithm 2 Learning algorithm based on the AdaBoost.

---

1) Given example image(x_1, y_1),...,(x_N, y_N), $y_i \in \{0, 1\}$for negatives and positives examples respectively.

2) Initialize weights

$$\omega_{1,i} = \begin{cases} \frac{1}{2M},\text{if } y_i = 1 \\ \frac{1}{2P},\text{otherwise} \end{cases} \tag{8}$$

where M and P are the number of negatives and positives respectively.

3) For $t = 1$, 2,...,T

a) Normalization for every weight,

$$\omega_{t,i} = \frac{\omega_{t,i}}{\sum_{i=1}^{N} \omega_{t,i}} \tag{9}$$

b) For each feature, train the weak classifier. The error is evaluated with respect to $\omega_{t,i}$

$$\epsilon_t = \sum_{i:y_i \neq h_t(x_i)} \omega_{t,i} \tag{10}$$

c) Choose $h_t(x)$with the lowest error ϵ_t

d) Update weights

$$\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-a_i} \tag{11}$$

where

$$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t} \tag{12}$$

$$a_i = \begin{cases} 0,\text{if } h_t(x_i) = y_i \\ 1, \text{ otherwise} \end{cases} \tag{13}$$

e)Calculate the weight of the weak classifier

$$\alpha_t = log \frac{1 - \epsilon_t}{\epsilon_t} \tag{14}$$

4) The final strong classifier is

$$H(x) = \begin{cases} 1, \sum_{t=1}^{T} \alpha_t h_t(x) \geq 0.5 \sum_{t=1}^{T} \alpha_t \\ 0, \text{ otherwise} \end{cases} \tag{15}$$

In each cycle of the boosting process, the best weak feature is selected according to the step (a) to (e).

In the learning algorithm,the final strong classifier $H(x)$ is linear combination of T weak classifiers $h_t(x)$,as Equation (15) shows.

The weak classifier used in this paper is proposed by Paul Viola and Michael Jones [7]. It is expressed as follows:

$$h_j(x) = \begin{cases} 1, \text{ if } p_j f_j(x) < p_j \theta_j \\ 0, \text{ otherwise} \end{cases} \tag{16}$$

where x is the feature vector of a sample.The weak classifier $h_j(x)$ consists of a feature f_j, a threshold θ_j and parity p_j, $p_j \in \{-1, 1\}$,indicates the direction of the inequality sign.

VI. Experiments and discussions

In this section, we first show the accuracy results of the proposed AdaBoost classifier, which is designed based on the BFH. Next, we list the operating time of the proposed algorithm. Moreover, the proposed novel features and LBP are also compared in terms of accuracy and processing time.

We use more than 1000 samples that were obtained from a glass manufacture plant in Shanghai to verify the effect of the proposed method. Some examples are shown in Fig. 6.

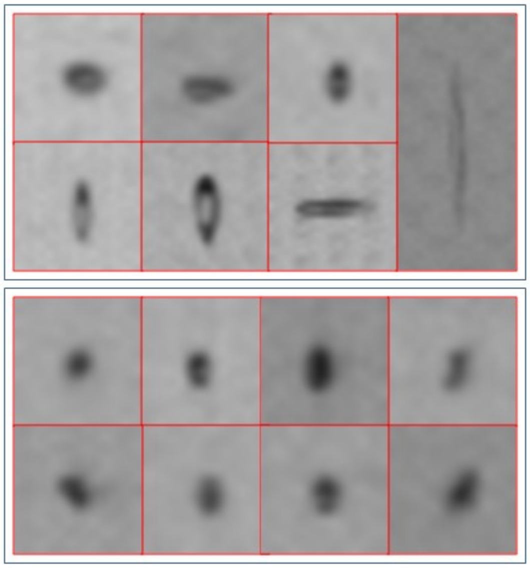


Figure 6. Some examples in the experiment data. Above: bubble, Below: non-bubble.

A. Accuracy experiment

In this experiment, 533 bubble samples and 167 non-bubble samples are randomly selected to train the classifier for bubble detection. By the training, the final classifier consists of 15 weak classifiers. On the other hand, the test data consists of 270 bubble samples and 63 non-bubble samples. The experimental results are listed in Table 1.

From the table, we can find:

1) The recognition ratio to the non-bubbles is lower than that to the bubbles. This result might be caused by the reason that the sample number of the non-bubbles is not very large, and the non-bubble cases are relatively complicated, compared to the bubble ones.
2) The general accuracy of the recognition algorithm is 91.6%.

B. Operating-speed experiment

In this experiment, we employ the computer with 2.6 GHz CPU and the software platform of Visual Studio 2008 with OpenCV 2.0. The operating time of the proposed algorithm is listed in Table 2.

The average processing period for one sample is around 0.048ms, which can satisfy the requirement of the industrial production.

Table I
ACCURACY OF THE TEST DATA.

Features	Total	Bubble	Non-Bubble
BFH	91.59%	92.22%	88.89%

C. Comparison experiment

The purpose we mention the LBP feature here is that this feature is also useful to solve our problem with a high accuracy. However by the experiment, we found this feature is not as good as the proposed features in the recognition performance. Before showing the experiment results, we briefly review the LBP feature.

LBP is a type of feature used for object recognition, which was first proposed in [5]. For each pixel in a cell, the pixel is compared to each of its 8 neighbors, which are followed along a circle. If the center pixel's value is greater than the neighbor, write "1"; otherwise, write "0". This definition gives an 8-digit binary number (feature number). One example of calculating the feature number is given in Fig. 7. Then the histogram is computed over the window of the frequency of each feature number occurring. The average difference of the LBP distribution in the two types is shown in Fig. 8.

In the comparison experiment, we use this 2^8 vectors of LBP as the feature vector to train the Adaboost classifier.The comparison results between the LBP and the proposed features are shown in Table 3.

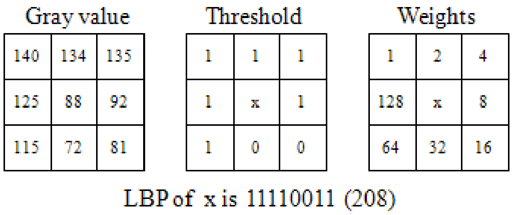
As Table 3 shows, the BFH yields higher performance than the LBP in both accuracy and operating speed. Additionally, the classifier based on the BFH can reach high inspection accuracy for both bubble and non-bubble data. This performance makes the proposed algorithm even closer to the engineering application.

VII. Conclusion and future work

This paper has presented a novel method for detection and classification of glass defects in low resolution images. First, the defect region is segmented by the method of Canny edge detection. Further, the smallest connected region (rectangle) is located. Then, the binary information of the core region can be obtained based on the specific filter and w-shaped

Table II
OPERATING TIME OF THE PROPOSED ALGORITHM.

Time	BFH Extraction	Classification	Total
Average	0.0245	0.0231	0.0476
Max	0.3656	0.0461	0.4117
Min	0.0093	0.0298	0.0391



Gray value

140	134	135
125	88	92
115	72	81

Threshold

1	1	1
1	x	1
1	0	0

Weights

1	2	4
128	x	8
64	32	16

LBP of x is 11110011 (208)

Figure 7. Example of one pixel's LBP value.

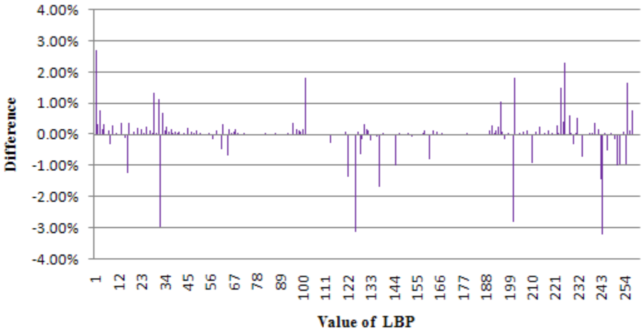


Figure 8. Average difference of the LBP distribution in the two types of defects.

Table III
COMPARISON BETWEEN LBP AND BFH

Item	LBP	BFH
Accuracy(Total)	88.89%	91.59%
Accuracy(Bubble)	90.59%	92.22%
Accuracy(Non-bubble)	83.33%	88.89%
DFV	256	9
NWC	91	15
Average time (ms)	0.1799	0.0476

DFV=Dimension of feature vectors; NWC=number of weak classifiers.

feature. Meanwhile, noises are removed. After that, a new binary feature histogram (BFH) is proposed to describe the characteristic of the glass defect. Finally, the classifier based on the AdaBoost algorithm is designed with the novel feature to realize the classification. The experiment shows this method can implement detection and classification of the glass defects in low resolution images exactly and rapidly. The accuracy has met the industrial standard of some sorts of products, and this method is going to be tested in the real industry locale. This detection method can also provide a reference for solving defect detection problems on other flat materials, such as steel.

Our future work will focus on: 1) improving the correct rate of the binarization; 2) combining BFH with other features to obtain higher accuracy; 3) improving the machine learning method (e.g., attempt the algorithm of boosting for transfer learning), in order to ensure robustness of the proposed algorithm.

ACKNOWLEDGMENT

This work was supported in part by the National Science Key Foundation of China under Grant 60833009 and in part by the National Science Foundation of China under Grant 60975012.

REFERENCES

[1] Zhou X and Liu X,“The detection algorithm of bubbles in safety glass image based on local region threshold,”Microcomputer Information,vol.23,2007,pp.252-254

[2] Peng X,Chen Y and Yu W,“An online defects inspection method for float glass fabrication based on machine vision,”International Journal of Advanced Manufacturing Technology, vol.39,2008,pp.1180-1189

[3] Han C,Ryu Y and Oh C,“A Study on Enhanced Algorithms for Detecting Defects of Glasses,”International Conference on Convergence and Hybrid Information Technology, vol.1,2009,pp.521-524

[4] Hu L and Wei Z,“An Algorithm of Glass-Image Recognition Based on Wavelet Packet Decomposition,”International Conference on Computational Intelligence and Natural Computing,vol.1,2009,pp.206-209

[5] Ojala T,Pietikainen M and Darwood D,“A comparative study of texture measures with classification based on feature distributions,”Pattern Recognition,vol.29,1996,pp.51-59

[6] Freund Y and Schapire R,“A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,”Journal of Computer and System Sciences,vol.55,1997,pp.119-139

[7] Viola P and Jones M,“Rapid object detection using a boosted cascade of simple features,”Proc. IEEE Conf. Computer Vision and Pattern Recognition,vol.1,2001,pp.511-518