

7 Lens Shading Correction for Dirt Detection

Chih-Wei Chen¹ and Chiou-Shann Fuh¹

Abstract We present a novel inspection framework to detect dirt and blemish in production line of optical fabrication automatically. To detect dirt defect from vignette surface, we use B-Spline to generate an ideal vignette surface and subtract the contaminated surface from the generated vignette surface to locate the defect. Besides, we apply some image processing methods to optimize our inspection framework. Experimental results show the effectiveness and robustness of the proposed approach regarding dirt detection in automatic optical inspection.

7.1 Introduction

There are recently been massive growth in the application of camera module. The following Sections 7.1.1 and 7.1.2 describe the concept of image-based biometrics and illustrate some practical issues of camera module respectively.

7.1.1 Biometrics

In recent years there has been renewal of interest in the application of biometrics. Although there are some privacy issues, the potential application of biometrics are still immense. In biometrics, the physiological biometric characteristic includes iris recognition [1–4], fingerprint [5–7], facial recognition [8–10], retina [11, 12] and so on.

The biometrics features can be extracted in various ways. Some of the biometrics data are stored as image format for the purpose of post processing. The iris image [4] shows in Fig. 7.1(a) is shoot in visible wavelength environment. Fig. 7.1(b) to 1(d) are fingerprint image [13], diagram of facial recognition [9], and retinal image [14] respectively. We may need to use image sensor to obtain the desired images. Therefore, the quality of image directly affects to the accuracy of the post processing of biometrics. Fig. 7.2 shows the comparison between favorable image and blemished image.

Generally, we cannot just apply image de-noise approach to reduce the effect of small blemish on input image because the area of blemish is unpredictable. In this paper, we introduce a framework to detect the defect

¹ Department of Computer Science and Information Engineering, Taiwan University, Taipei 106, Taiwan, Chain. E-mail: {d95013, fuh}@csie.ntu.edu.tw.

from input image. While the defect been detected, manufacturer can apply the proposed method on production line to decrease the influence caused by blemish.

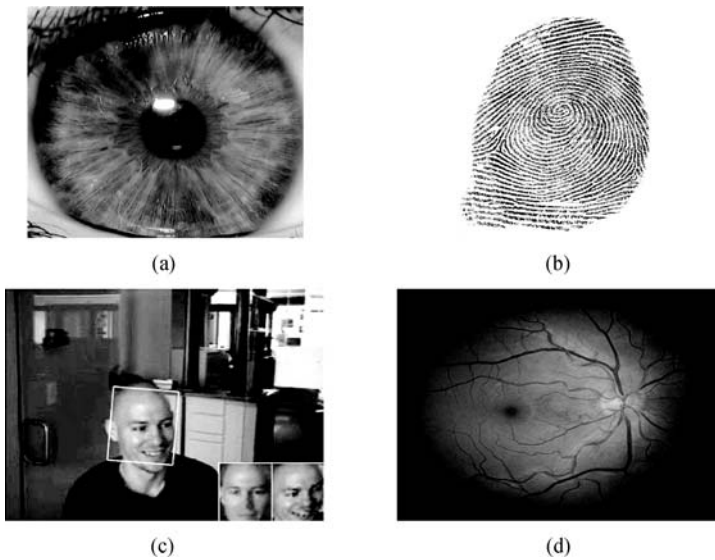


Fig. 7.1 (a) The captured iris image. (b) Fingerprint image. (c) Facial recognition in biometrics. (d) Retinal image.

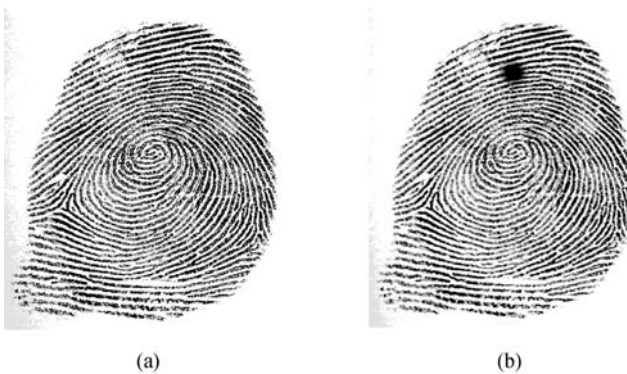


Fig. 7.2 (a) Favorable fingerprint image. (b) Blemished fingerprint image.

7.1.2 Issues of Camera Module

Camera module can be applied to acquire the required biometrics image

because the related devices are obtainable and inexpensive. There are some practical issues should be dealt with before applying camera module over biometrics. This section will briefly introduce camera module and illustrate the issues caused by blemish.

Recently, the application of camera module becomes more widespread. To provide more portability and feasibility of application, the developing trend of camera module is manufactured toward small dimension and low cost.

The conventional approaches to achieve the above mentioned object is to use fewer optical lenses to lower camera module dimension and substitute plastic material for glass material to decrease cost. Unfortunately, when we use fewer optical lenses, some aberration effects become conspicuous. One of the effects shown in Fig. 7.3 is called lens shading. Lens shading phenomenon is introduced in Section 7.2.1. Simultaneously, dirt and blemish detection is needed in production line during camera module fabrication. The red circle in Fig. 7.4 shows contaminated region.

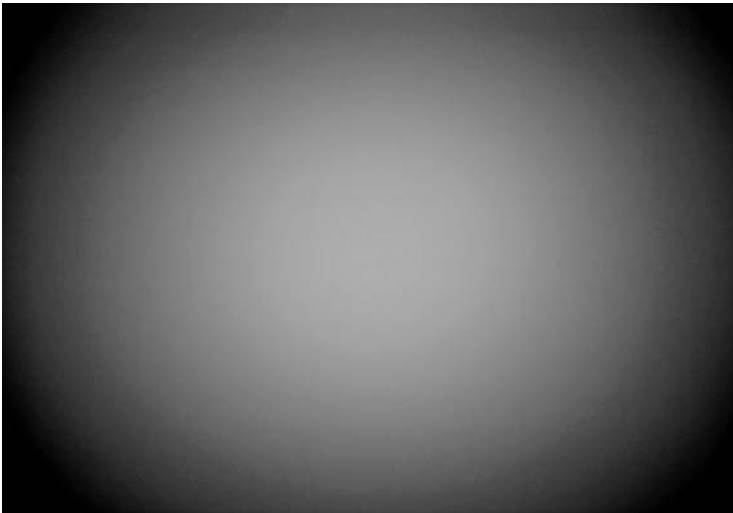


Fig. 7.3 The vignette effect shows up while shooting the Kodak Gray Card.

Occasionally, the difference between defect and vignette of lens shading is not conspicuous. It is difficult to perceive even through manual verification. Fig. 7.5 gives an example of inconspicuous defect; the dirt region in red circle is similar to the background.

In this article, we propose a systematic framework to achieve the objective of automatic optical dirt detection. The remainder of this paper is organized as follows. In Section 7.2, we introduce some background knowledge of our proposed framework. Section 7.3 presents our system framework and describes the framework step-by-step. In Section 7.4, the realistic defect image and experimental results are presented. Finally, the conclusion is drawn in Section 7.5.

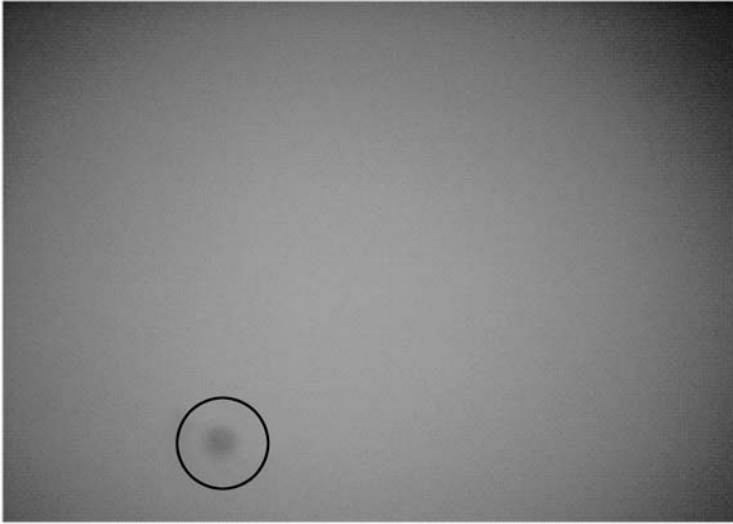


Fig. 7.4 Lens shading image with contaminated region.

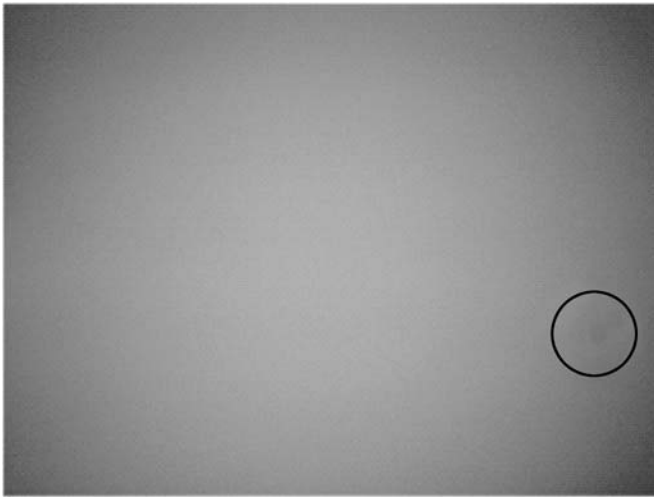


Fig. 7.5 Lens shading image with inconspicuous contaminated region.

7.2 Background

Background knowledge include lens shading phenomenon and color filter array is introduced in Section 7.2.1 and Section 7.2.2 respectively. By integrating a series of techniques into our proposed framework, the following Sections 7.2.3 to 7.2.6 show how it is possible to overcome the limitations of each individual processing step.

7.2.1 Lens Shading Phenomenon

The phenomenon of lens shading is the reduction in light falling on the image sensor away from the center of the optical axis caused by physical obstructions. Fig. 7.3 shows a phenomenon of lens shading while we shoot the Kodak Gray Card. In general, the vignette effect of lens shading may be asymmetric.

The effect of relative illumination can be calibrated by well optical design. But it becomes a challenge to calibrate vignette effect without increasing the dimension of optical design. There are many patents proposed to correct the shading effect by image post-processing. X. Y. Wang et al. proposed a lens shading correction approach by using B-spline curve fitting [15]. In their patent, the inventors reveal that the feasibility of conventional symmetric method for lens shading correction was limited by optical alignment and radial symmetry. Furthermore, the inventors also indicate that the two-direction multiplication method often inadequately corrects the shading effect. In this research, we apply B-spline curve to simulate the shading surface. The operating process is presented in Section 7.3.

7.2.2 Color Filter Array

In photography, since the typical image sensors cannot distinguish specific wavelength, color filter arrays are needed [16]. Commonly, color filter arrays have been placed above the image sensor. Fig. 7.6 shows the diagram of the assembling organization about microlens, color filter array, and image sensor.

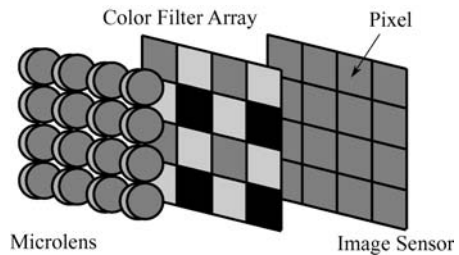


Fig. 7.6 This schematic diagram illustrates the relation between color filter array and image sensor.

There are many different types of color filter arrays proposed. Fig. 7.7 shows some types of color filter arrays.

In this research, we use conventional RGB (Red, Green, Blue) images as input data. The conventional RGB image is demosaiced from raw image by using the color interpolation algorithm [17]. The color filter array of the input images in this article is Bayer filter. By observation, the green channel has occupied fifty percent of the whole image; the red channel and blue channel have occupied twenty five percent respectively. In other words, the green channel has higher chance to collect the integral information from real

scene. Therefore, we only use the green channel in each input image to save computation cost.

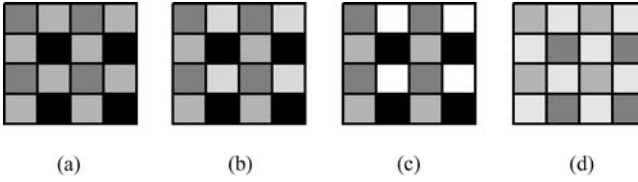


Fig. 7.7 Various types of color filter arrays. (a) Bayer filter. (b) RGBE (Red, Green, Blue, Emerald) filter. (c) RGBW (Red, Green, Blue, White) filter. (d) CYYM (Cyan, Yellow, Yellow, Magenta) filter.

7.2.3 Image De-Noise

After taking green channel of input image, the next step is to lower this image's noise. In this research, the objective of noise reduction is to make the sampling point (control point) in B-Spline more generalized.

Many of noise reduction approaches have been proposed to decrease the defect (or distortion data) from given image without affecting the image detail. For instance, A. Buades et al. proposed non-local means algorithm [18]; C. Tomasi et al. proposed bilateral filter approach [19]; and so on.

By observation, the noise patterns shown in images can be effectively reduced by using conventional 9×9 pixels median filter. The following figures show the result of noise reduction after applying the 9×9 pixels median filter.

Although the visual difference is slight, the process of image de-noise is indispensable for the following processes. Fig. 7.9 shows the result of Sobel edge detection in Fig. 7.8. After image de-noising, the intensity variation in Fig. 7.9(b) becomes stable and more suitable for the sampling control point of the B-Spline algorithm.

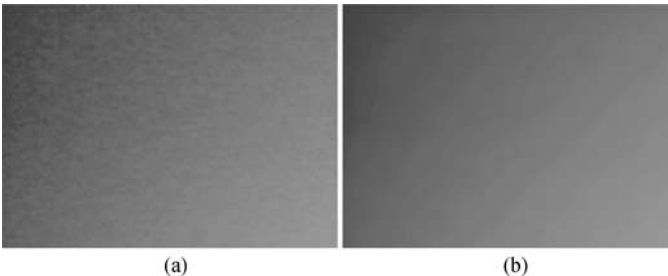


Fig. 7.8 The visual difference between input image and de-noised image. (a) Input image. (b) Result of 9×9 pixels median filter in (a).

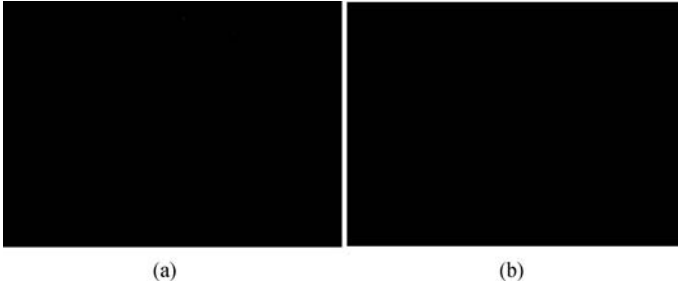


Fig. 7.9 (a) The result of Sobel filtering the image in Fig. 7.8(a). (b) Result of Sobel filtering the image in Fig. 7.8(b).

7.2.4 Histogram Equalization

To enhance the contrast of whole image, we apply histogram equalization to make the slight variance of given image prominent. For discrete application, the probability of occurrence of intensity r_k in an image can be denoted by [20]

$$p_r(r_k) = \frac{n_k}{n}, \quad k = 0, 1, 2, \dots, L - 1,$$

where n_k means the total pixels of specific intensity r_k ; n means total pixels of whole image; and L means the number of intensity gray levels in the image ($L = 256$ for 8 bit/pixel image). Let r means the normalized gray level (from $[0, 255]$ to $[0, 1]$), Fig. 7.10 shows the concept of histogram equalization.

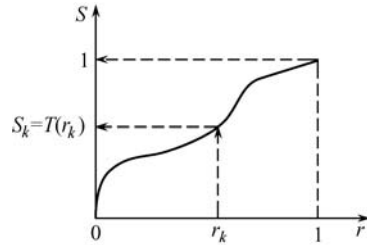


Fig. 7.10 Concept of transformation from gray level r to level s .

The transformation function of histogram equalization can be defined by

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{n}, \quad k = 0, 1, 2, \dots, L - 1.$$

And, the inverse transformation from s back to r is defined by:

$$r_k = T^{-1}(s_k), \quad k = 0, 1, 2, \dots, L - 1.$$

Fig. 7.11 shows the effect of the image enhancement by histogram equalization.



Fig. 7.11 (a) An image with slight variance of intensity. (b) Result of histogram equalization in (a). Note: The bright spot at the lower right corner is emphasized.

By observing Fig. 7.11(b), a bright spot appears at the lower right corner of the image. Besides, there are many sparse spots spread in the image. To increase the accuracy of our blemish detection, we apply morphological image processing to decrease the effect of those sparse spots.

7.2.5 Morphological Operation

The theory of morphology is based on set theory [20], and can be applied to achieve many image processing techniques. For instance, image component extracting, image thinning, image pruning, and so on. In this research, we use mathematical morphological opening to lower the sparse spots mentioned in Fig. 7.11(b). Opening can be denoted by

$$A \circ B = (A \ominus B) \oplus B,$$

where $A \circ B$ means the opening operation of set A by B ; \ominus means the erosion operation; and \oplus means the dilation operation. We use the following rule to implement erosion during realistic implementation:

$$I(x, y) = \arg \min I(x + x', y + y'),$$

where $I(x, y)$ means the intensity after erosion at position (x, y) . By using 3×3 pixels square structuring element, we select the minimum intensity from current position (x, y) and its eight neighboring pixels, denoted by (x', y') , as current intensity. Similarly, we use the following rule to implement dilation during implementation:

$$I(x, y) = \arg \max I(x + x', y + y').$$

We select the maximum intensity from position (x, y) and its eight neighboring pixels as current intensity. The structuring element is fixed in this

article; we apply 3 times of erosion before 3 times of dilation to achieve preferable result. Fig. 7.13(b) shows the result of opening operation in the Fig. 7.12(a).



Fig. 7.12 (a) An image with many sparse spots. (b) Result of morphological operation based on 3 times of erosion operation and 3 times of dilation operation.

The iteration of erosion and dilation can be modified to suit the requirement of specific inspection.

7.2.6 B-Spline

The B-Spline curve is proposed by Isaac Jacob Schoenberg in 1946 [21–24]. B-Spline curve is a generalization of the Bézier curve, and can be applied to many fields, such as, computer-aid design, computer graphics, engineering mechanics, and so on [25, 21, 23].

We fit the lens shading image by using the B-Spline curve, because the B-Spline curve can avoid the Runge phenomenon. In this article, we implement the uniform quadratic B-Spline to generate the ideal vignette surface. The uniform quadratic B-Spline is denoted by [24]

$$S_i(t) = [t^2 \quad t \quad 1] \begin{bmatrix} 0.5 & -1 & 0.5 \\ -1 & 1 & 0 \\ 0.5 & 0.5 & 0 \end{bmatrix} \begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \end{bmatrix},$$

for $t \in [0, 1]$, $i = 1, 2, \dots, m - 2$,

where p_{i-1} , p_i , and p_{i+1} mean the set of control points; S_i means the i th B-Spline segment. The detail of the generated surface depends on the number of sampling control points. The more control points we sample, the more exquisite surface we get. We use vertical control points to generate vertical B-Spline. Then, we apply the generated vertical B-Spline to generate whole vignette surface. The diagram of vertical B-Spline generation is shown in Fig. 7.13.

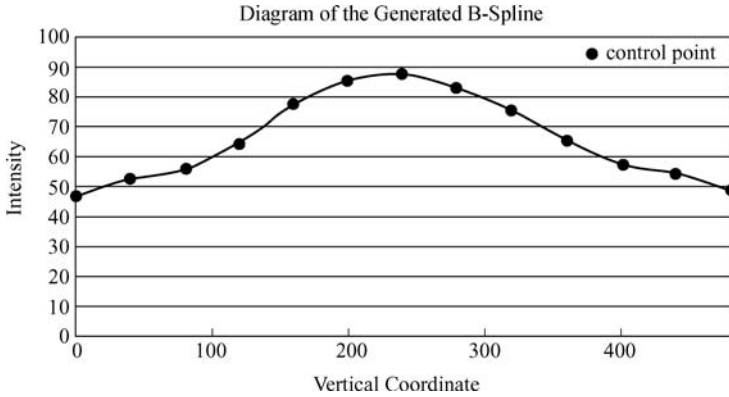


Fig. 7.13 The diagram is a concept of vertical B-Spline generation using 13 control points. After vertical B-Spline generation, we can use them to generate horizontal B-Spline.

In this article, we extract 31 control points in each vertical sample, and 31 control points in each horizontal sample. Thus, the total number of control points is 961 ($=31 \times 31$). Fig. 7.14(b) shows the result of B-Spline generated surface in Fig. 7.14(a).

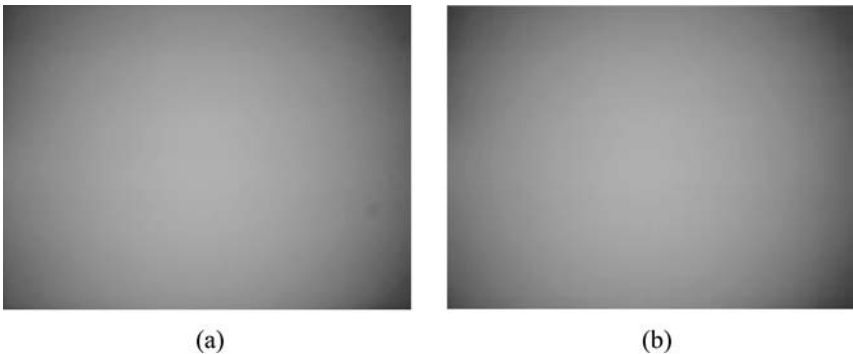


Fig. 7.14 (a) The given image. (b) Result of generated B-Spline surface in (a).

7.3 Our Proposed Method

To detect dirt from vignette image, we use the backgrounds mentioned in Section 7.2. Our proposed framework is shown in Fig. 7.15.

We use the green channel of the input image to lower the issues caused by image demosaicing process. To generate an appropriate B-Spline surface, we have to moderate the intensity variation in the extracted green channel

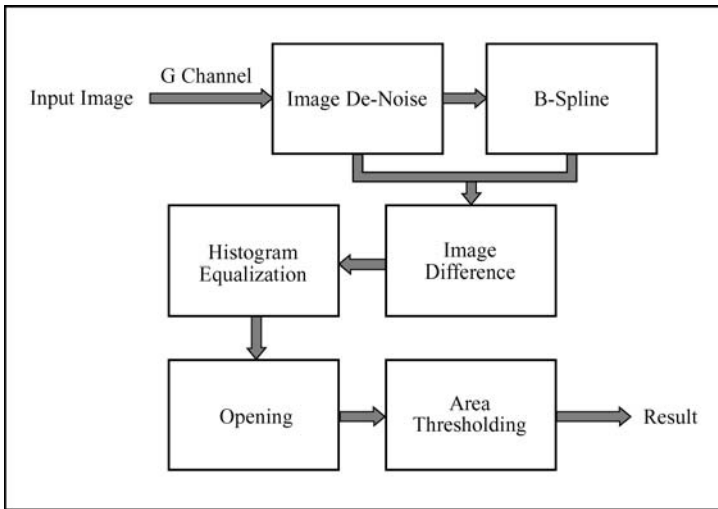
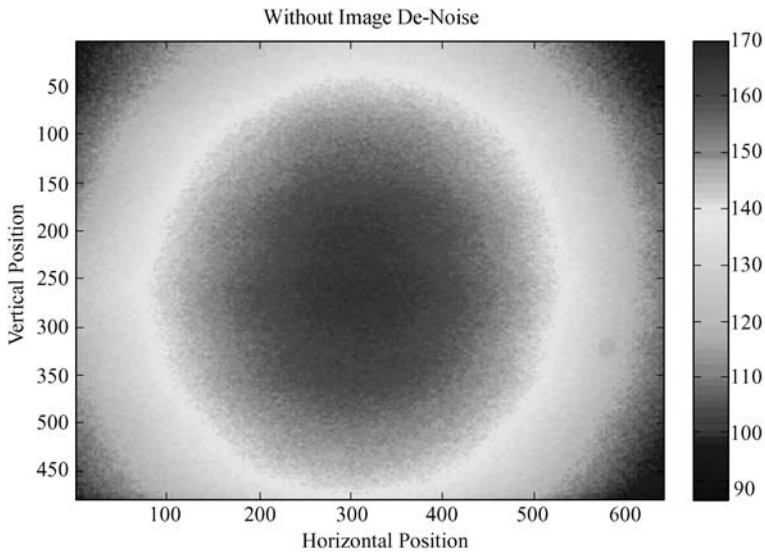


Fig. 7.15 The proposed system flowchart.

image. Thus, we smooth the green channel image by using 9×9 pixels median filter. The original image and smoothed result are shown in Fig. 7.16 and 7.17.



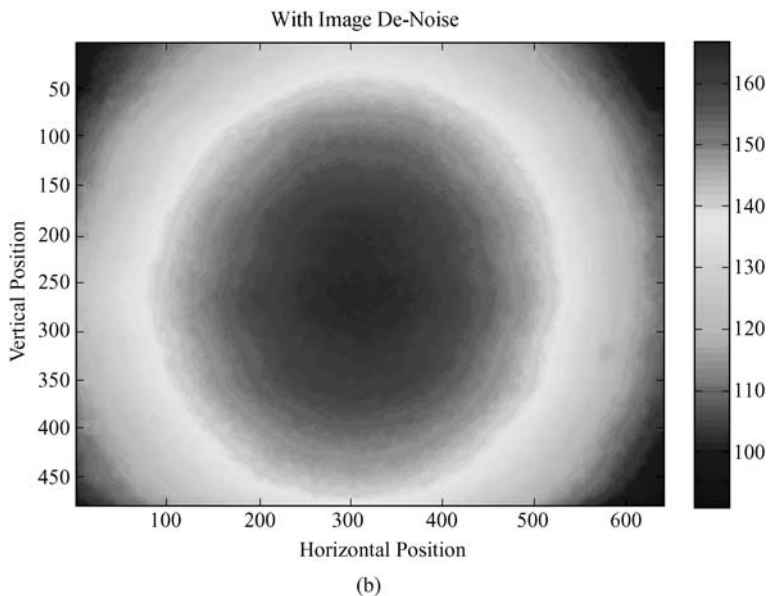
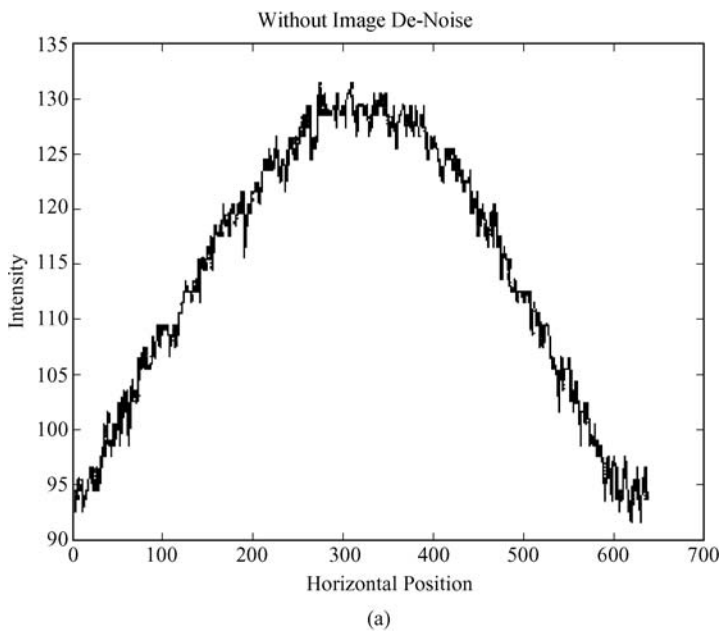


Fig. 7.16 (a) Scalar image of original image. (b) Scalar image of de-noised image.



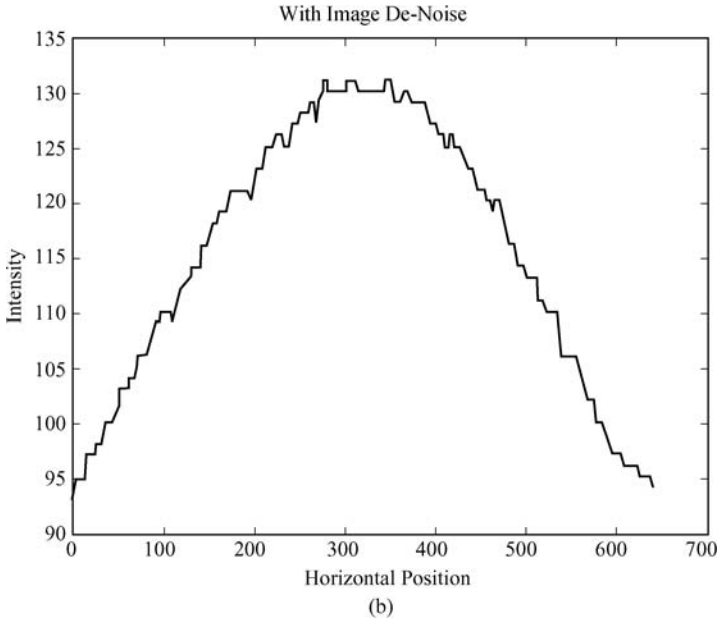


Fig. 7.17 Compare the same image line between the given image and the smoothed image. (a) Image without de-noised process. (b) Result of 9×9 pixels median filter process in (a).

After image de-noise, we extract M pixels' intensities as control points from N vertical sampling. Fig. 7.18 shows the concept of the above mentioned sampling process; red points in Fig. 7.18 are sampled control points.

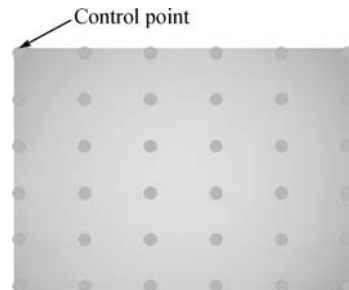


Fig. 7.18 The red points in this image are sampled control points. In this diagram, we extract 6 pixels ($M=6$) from each sampled column, and 6 columns ($N=6$) from the whole image.

Then, we use the $M \times N$ control points to first generate vertical direction B-Spline curves. The hollow blue rectangles shown in Fig. 7.19 are the required B-Spline curves, and the solid rectangles shown in Fig. 7.20 are the generated B-Spline curves. Finally, we generate horizontal B-Spline curves to complete whole B-Spline surface by using existing vertical B-Spline information. The hollow green rectangle shown in Fig. 7.20 is one of the required horizontal B-Spline curves, and the blue parts inside green rectangle are control

points. The above mentioned B-Spline curve generation can be accelerated by using parallel computation.

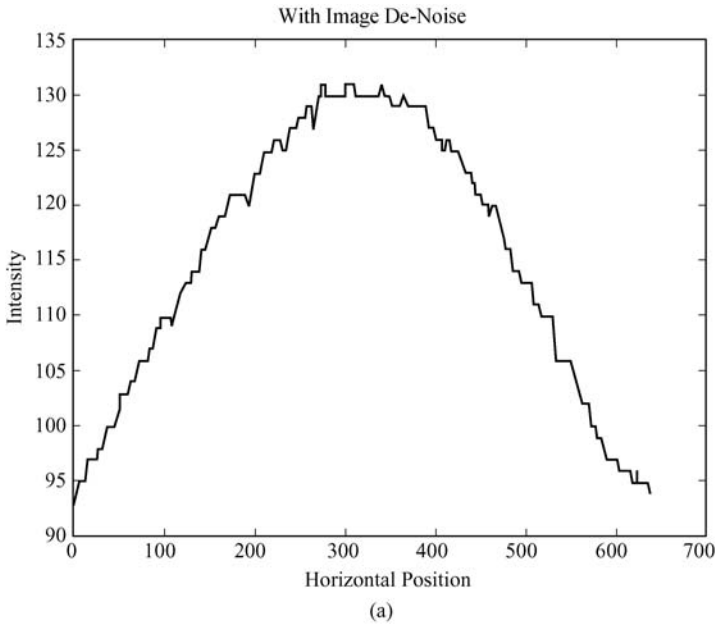


Fig. 7.19 The hollow blue rectangles are the required B-Spline curves.



Fig. 7.20 The solid blue rectangles are the generated B-Spline curves from Fig. 7.19, and the hollow green rectangle is one of the required B-Spline curves. We can use blue parts inside the green rectangle as control points to compute its B-Spline curve.

Fig. 7.21(b) shows the performance of the generated B-Spline curve, and Fig. 7.22(b) shows the performance of the generated B-Spline surface.



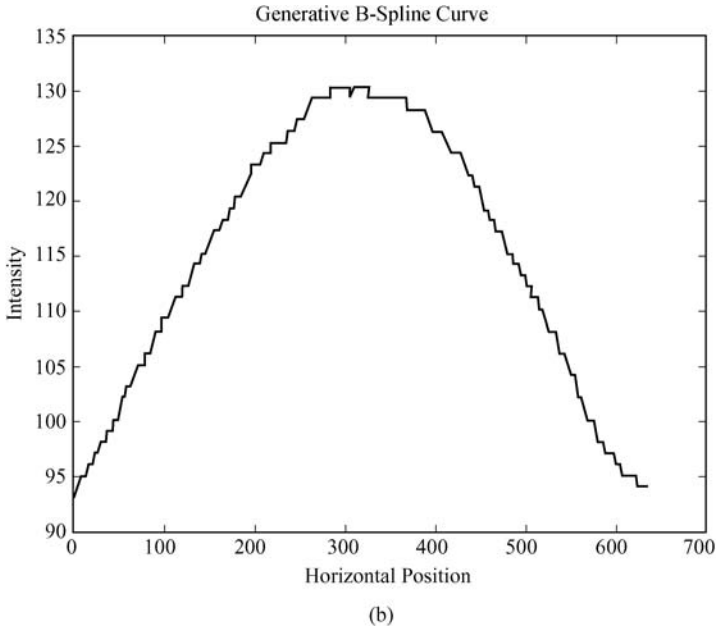
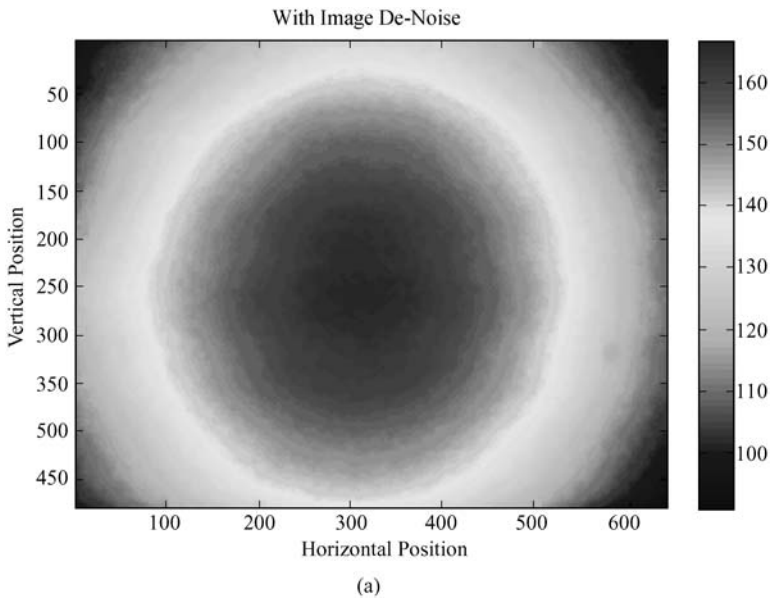


Fig. 7.21 Compare the same image line between the de-noising image and the generated B-Spline surface. (a) This curve shows the intensity variation of the de-noising image. (b) This curve means the generated B-Spline image.



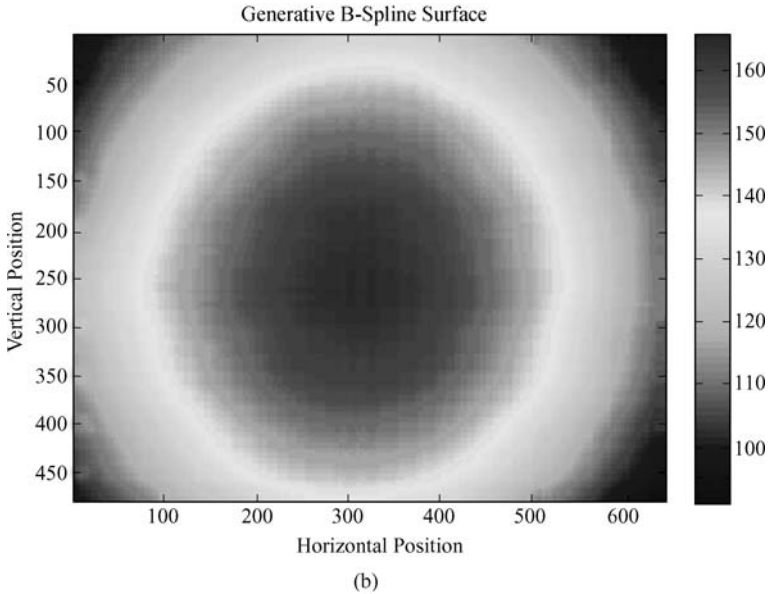


Fig. 7.22 (a) Scalar image of de-noised image. (b) Scalar image of generated B-Spline surface.

Next, we subtract the generated B-Spline surface from de-noised image to compute image difference. Fig. 7.23 shows the result of image difference. To prevent the intensity variation of image difference from over-smoothing, we apply histogram equalization to enhance the result of image difference.

Unfortunately, some conspicuous perturbations in input image may be enhanced simultaneously. To lower the effect of the perturbations of input image, we use morphological operation to remove small isolated areas. Here, we use 3×3 pixels structuring element mentioned in Section 7.2.5 to process 3 times of erosion before 3 times of dilation to achieve preferable result.

After the morphological opening, most of small isolated areas caused by conspicuous perturbations have been removed. To preserve the flexibility of application, we add an area threshold function as the final step. The objective of area threshold function is to fill the area whose pixel number is larger than or equal to threshold as a specific color, and to fill the area whose pixel number is smaller than threshold as another color. Thus, the dirt detection result can be easily rechecked by user if necessary.

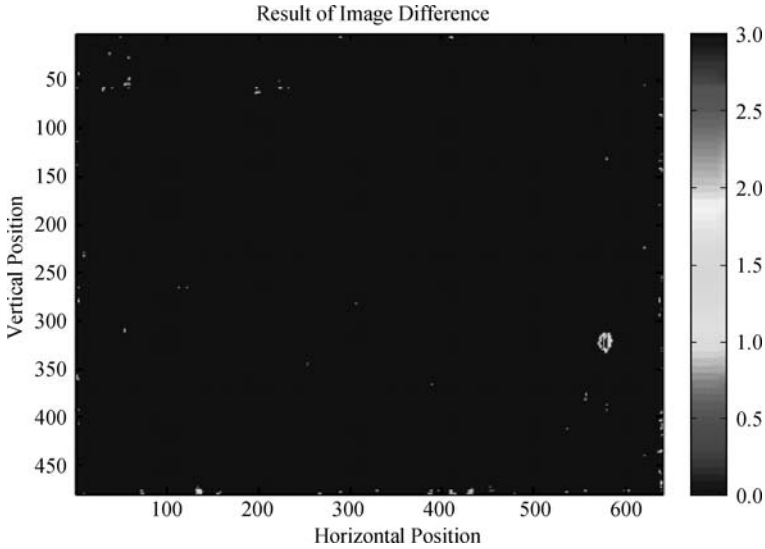


Fig. 7.23 The result of image difference. We subtract the generated B-Spline surface from de-noised image to obtain this result.

7.4 Experimental Results

In this article, we use real 640×480 pixels inspection image as input image. The median filter kernel is set as 9×9 pixels. We use $961 (= 31 \times 31)$ control points to generate ideal B-Spline surface. Moreover, we apply 3 times of erosion before 3 times of dilation with 3×3 pixels structuring element mentioned in Section 7.2.5 to achieve preferable result. Besides, we set the area threshold as 85 pixels. Here, if the area of dirt particle is larger than or equal to 85 pixels, then the dirt particle will be drawn with green color, otherwise, the dirt particle will be drawn with red color.

The experimental environment is on Intel Core 2 CPU Q6600 2.4 GHz (our program only uses single core), and 2 GB RAM. The average processing time is about 2.8 seconds without using any code optimization or hardware acceleration.

The respective results of our proposed framework are shown in Fig. 7.24 to 7.32. In our experiment of 13 images, we achieve 0 misdetections and 0 false alarms.

A special case is shown in Fig. 7.28 where many noises distribute over input image. The sparse spots appear in enhanced result in Fig. 7.28(f) seriously. By using mathematic morphological opening, most of the sparse spots are removed. But, there still are some noises after opening. Therefore we apply area threshold function to emphasize the region with larger area. Users can easily distinguish the obvious or the slight defect region.

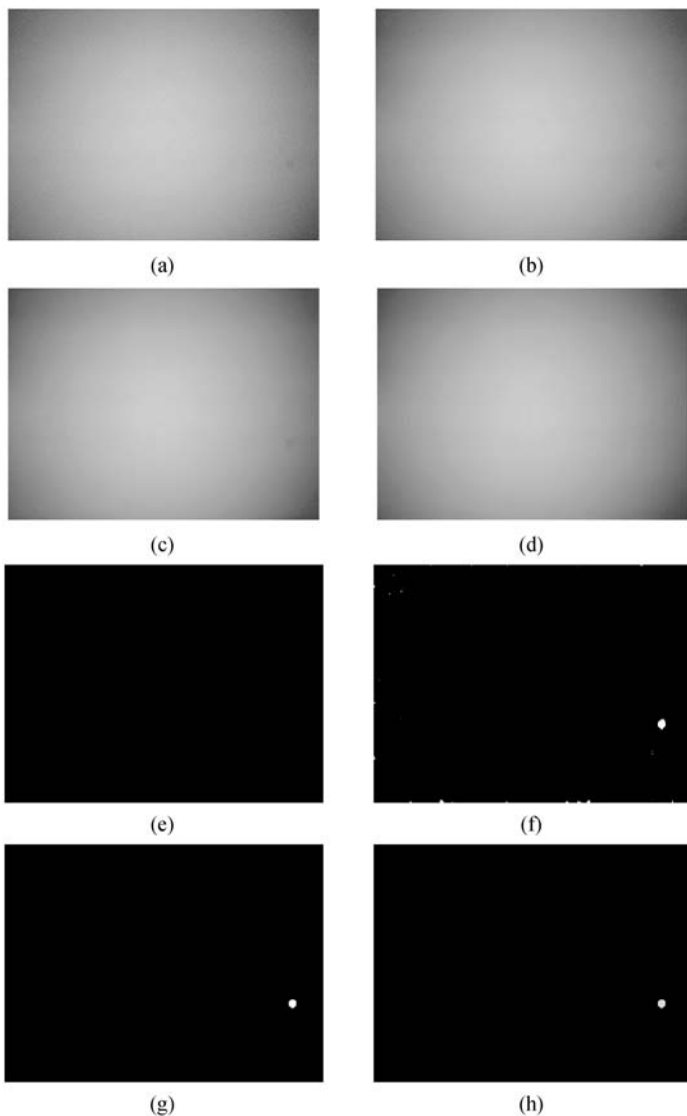


Fig. 7.24 Test Image 1. (a) Input image. (b) The green channel of input image. (c) De-noised result in (b). (d) The generated B-Spline surface. (e) Result of image difference. (f) Result of histogram equalization in (e). (g) Result of Opening in (f). (h) Result of area threshold in (g).

Besides, when the size of input image is large, user can apply down-sampling to lower processing cost. Of course, the parameters can be modified to fit different requirements.

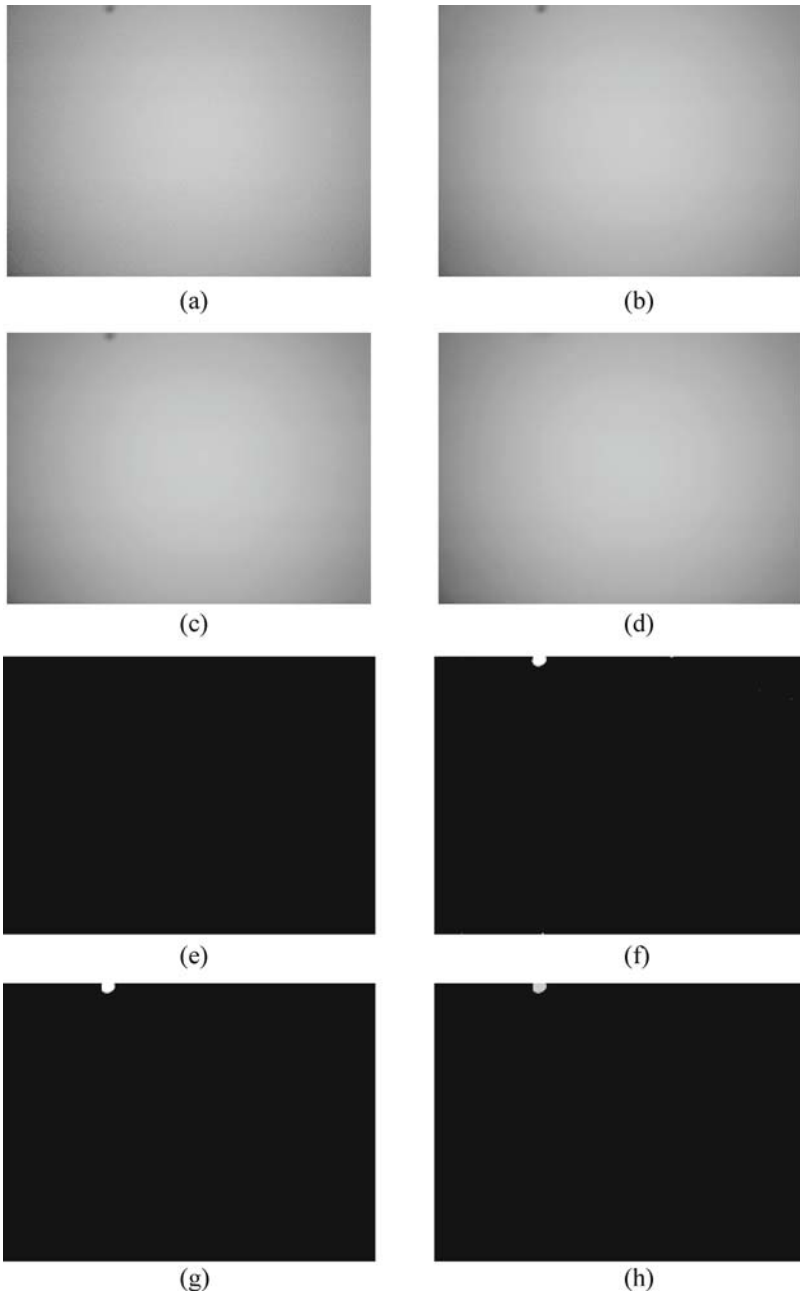


Fig. 7.25 Test Image 2. (a) Input image. (b) The green channel of input image. (c) De-noised result in (b). (d) The generated B-Spline surface. (e) Result of image difference. (f) Result of histogram equalization in (e). (g) Result of Opening in (f). (h) Result of area threshold in (g).

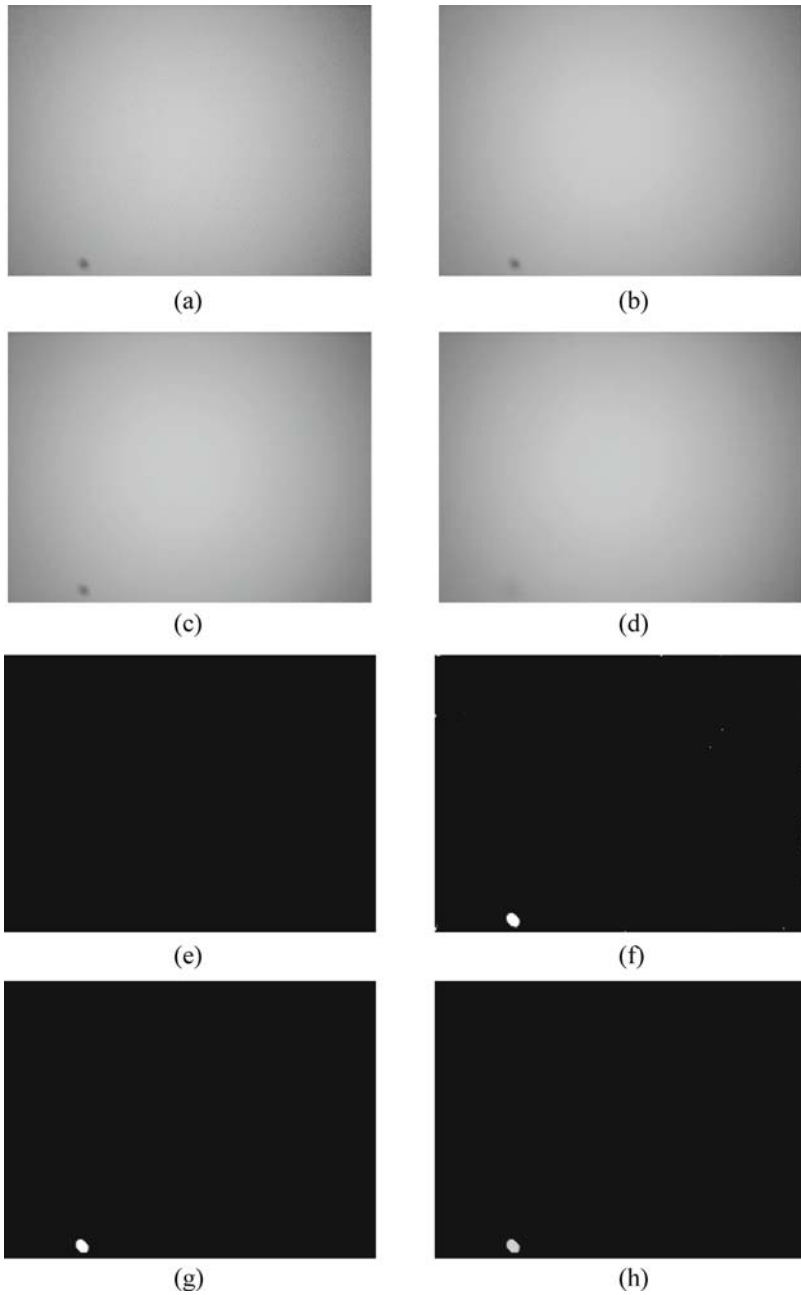


Fig. 7.26 Test Image 3. (a) Input image. (b) The green channel of input image. (c) De-noised result in (b). (d) The generated B-Spline surface. (e) Result of image difference. (f) Result of histogram equalization in (e). (g) Result of Opening in (f). (h) Result of area threshold in (g).

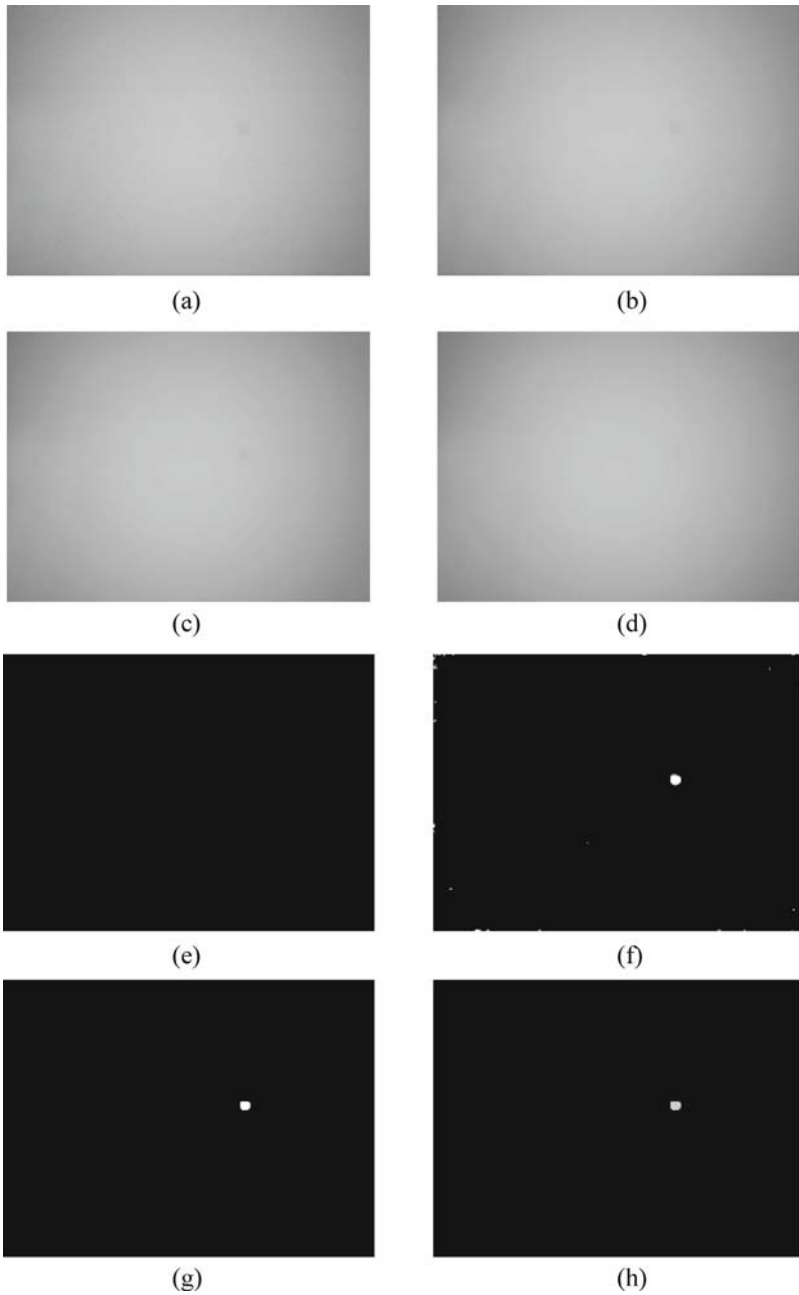


Fig. 7.27 Test Image 4. (a) Input image. (b) The green channel of input image. (c) De-noised result in (b). (d) The generated B-Spline surface. (e) Result of image difference. (f) Result of histogram equalization in (e). (g) Result of Opening in (f). (h) Result of area threshold in (g).

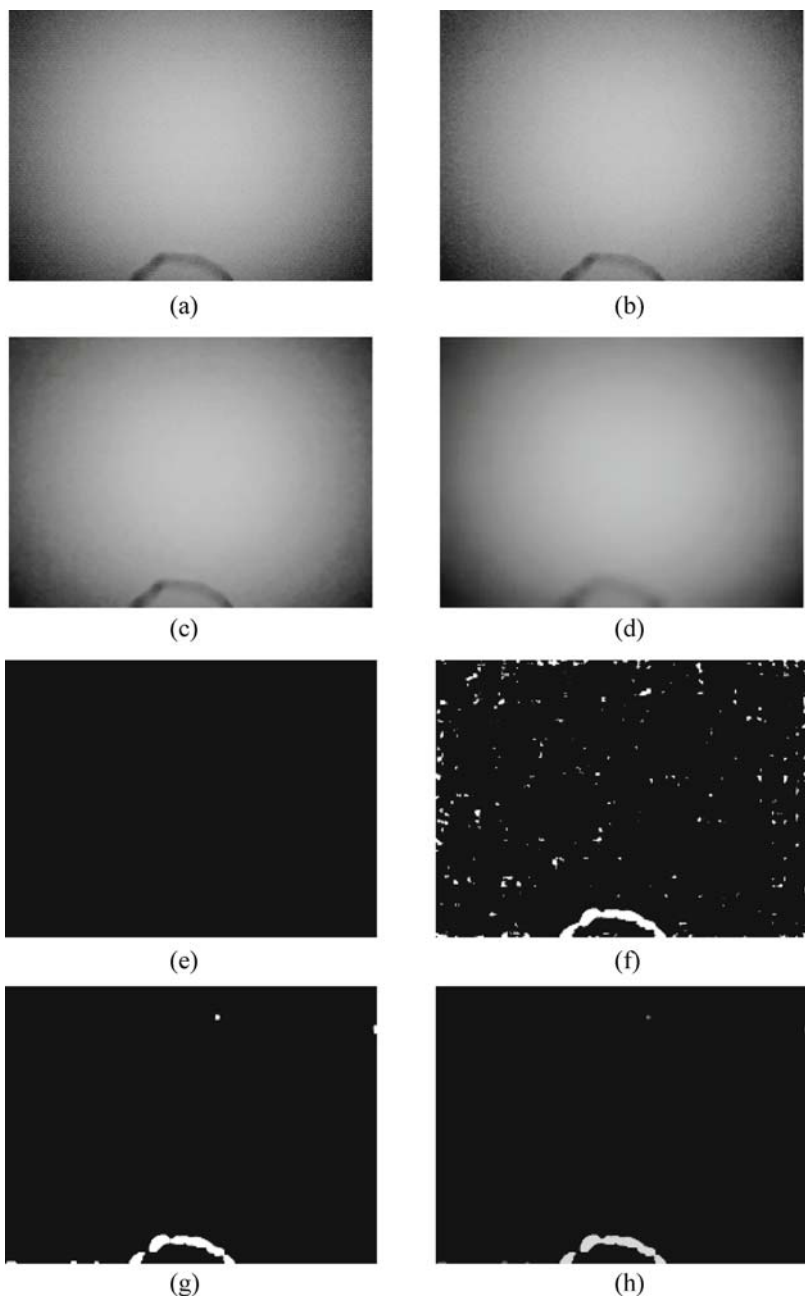


Fig. 7.28 Test Image 5. (a) Input image. (b) The green channel of input image. (c) De-noised result in (b). (d) The generated B-Spline surface. (e) Result of image difference. (f) Result of histogram equalization in (e). (g) Result of Opening in (f). (h) Result of area threshold in (g).

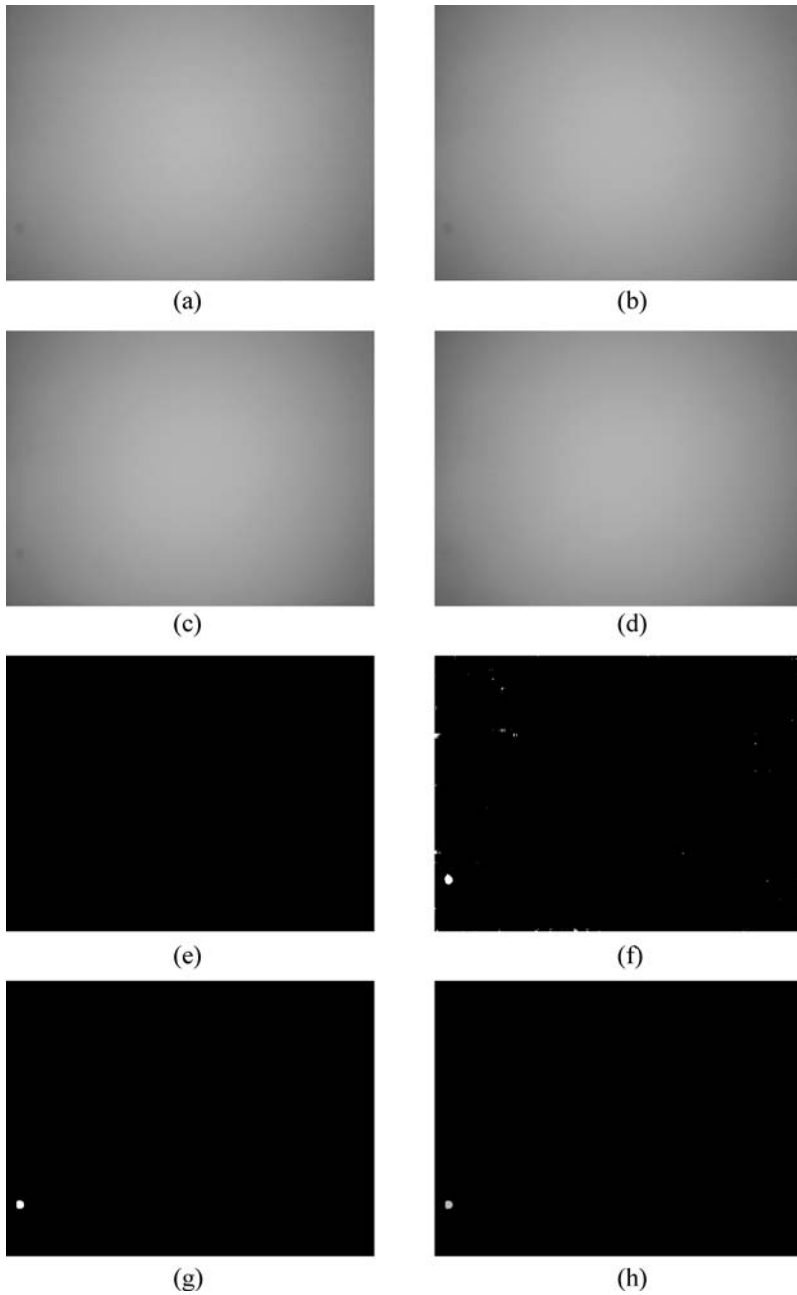


Fig. 7.29 Test Image 6. (a) Input image. (b) The green channel of input image. (c) De-noised result in (b). (d) The generated B-Spline surface. (e) Result of image difference. (f) Result of histogram equalization in (e). (g) Result of Opening in (f). (h) Result of area threshold in (g).

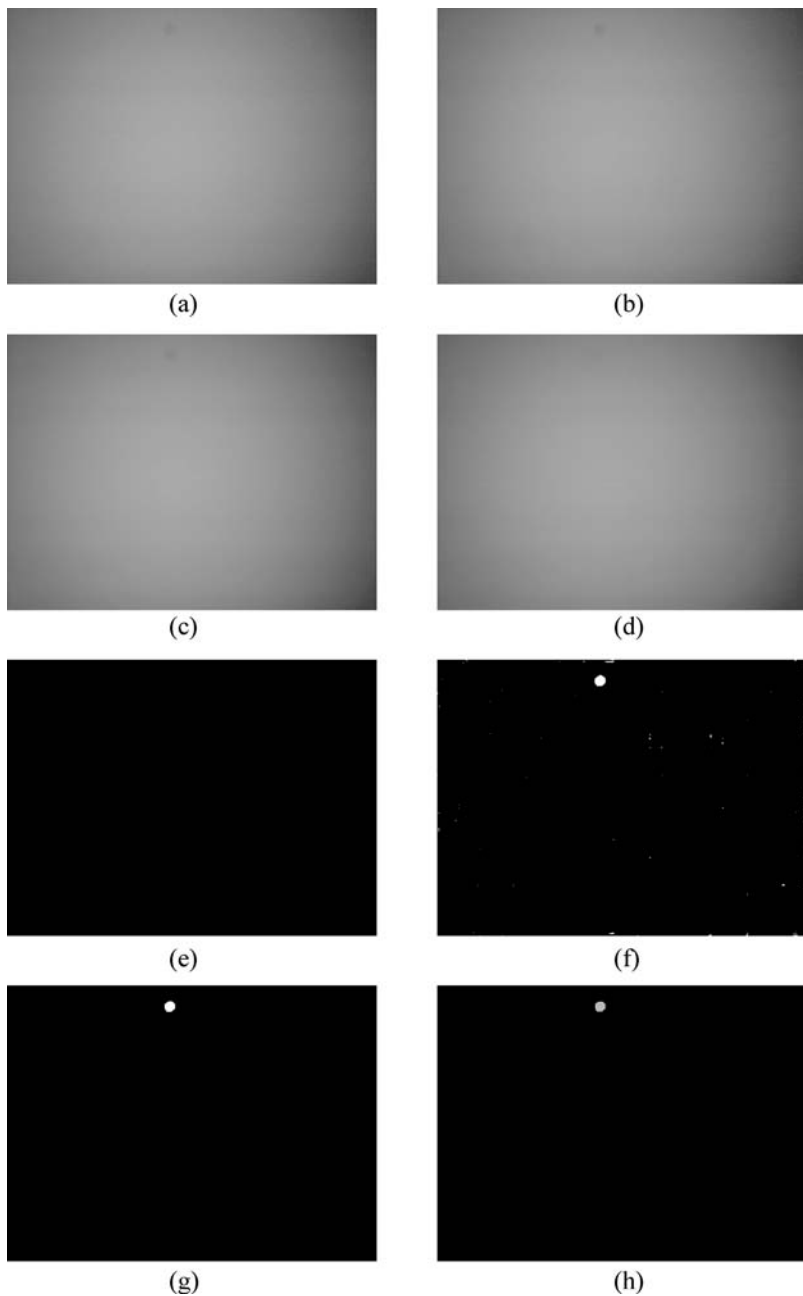


Fig. 7.30 Test Image 7. (a) Input image. (b) The green channel of input image. (c) De-noised result in (b). (d) The generated B-Spline surface. (e) Result of image difference. (f) Result of histogram equalization in (e). (g) Result of Opening in (f). (h) Result of area threshold in (g).

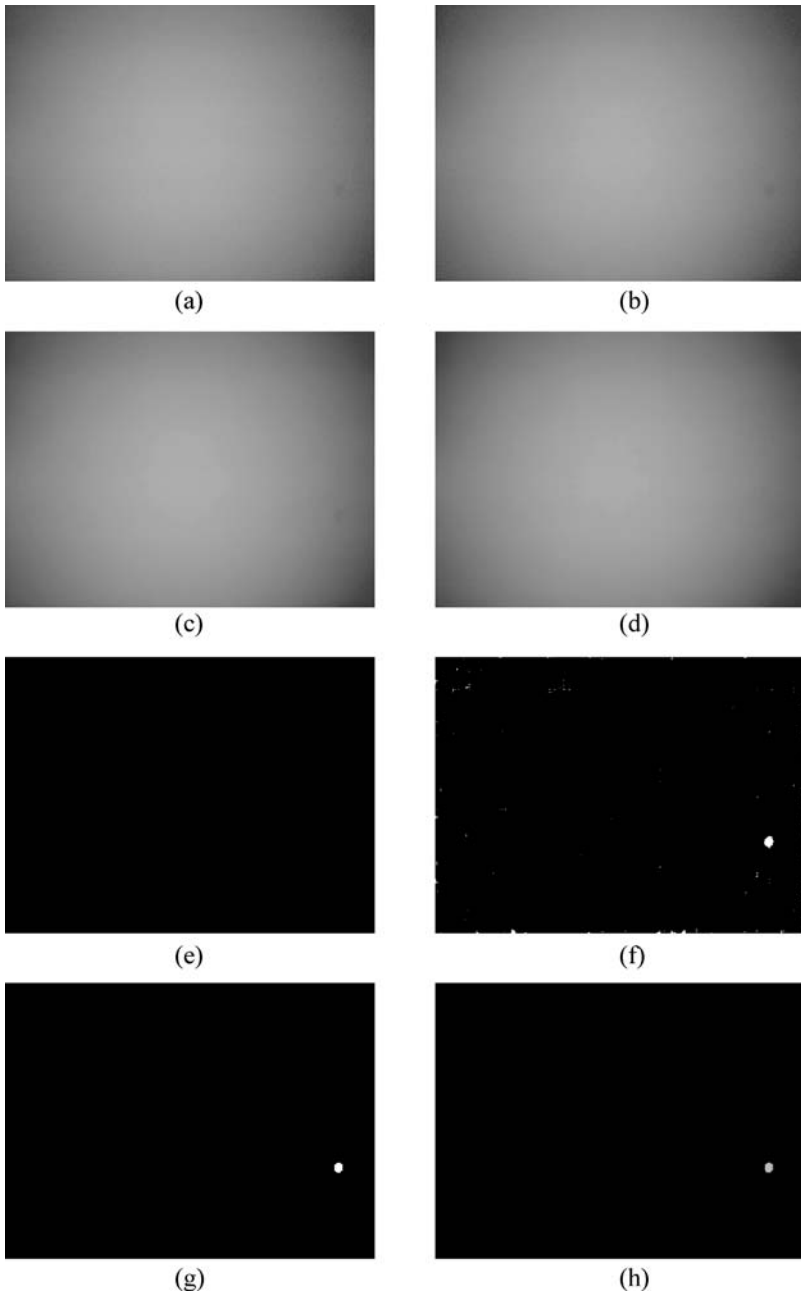


Fig. 7.31 Test Image 8. (a) Input image. (b) The green channel of input image. (c) De-noised result in (b). (d) The generated B-Spline surface. (e) Result of image difference. (f) Result of histogram equalization in (e). (g) Result of Opening in (f). (h) Result of area threshold in (g).

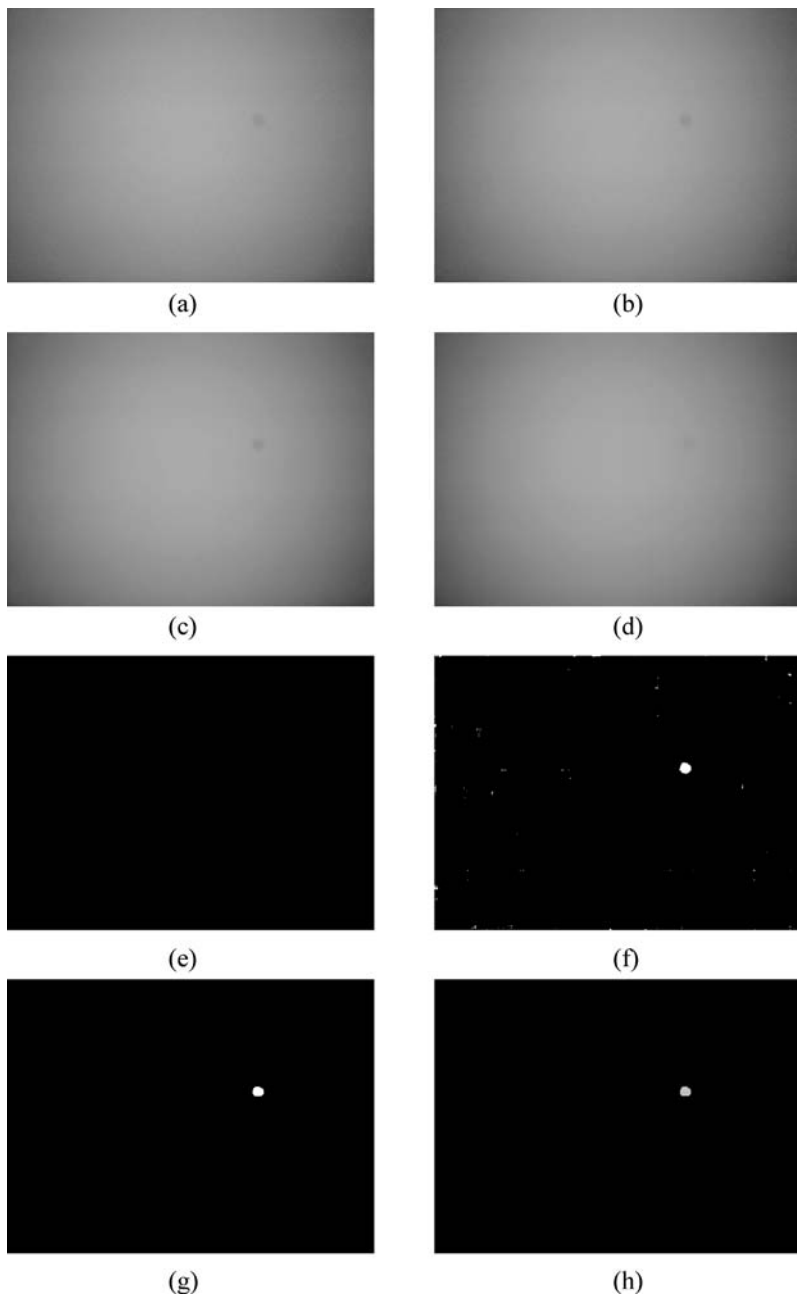


Fig. 7.32 Test Image 9. (a) Input image. (b) The green channel of input image. (c) De-noised result in (b). (d) The generated B-Spline surface. (e) Result of image difference. (f) Result of histogram equalization in (e). (g) Result of Opening in (f). (h) Result of area threshold in (g).

7.5 Conclusions

In automatic optical inspection, the technology to detect the defect parts (or interesting parts) in given image automatically is needed. In this article, we propose a novel approach to extract defect regions from lens shading image. By applying the proposed method to inspect the optical component, we can make the optical device more reliable and more accurate for its future application. Although, the proposed method in this article tends to deal with specific problem, the concept of the approach can be extended to other inspection.

There still are many topics, including automatic defect detection for LCD (Liquid Crystal Display) panel surface, and so on.

Acknowledgements

This research was supported by the Science Council of Taiwan, China, under Grants NSC 98-2221-E-002 -150 -MY3 and NSC 95-2221-E-002-276-MY3, by EeRise Corporation, Jeilin Technology, Alpha Imaging, Winstar Technology, Test Research, Faraday Technology, Vivotek, Lite-on, and Syntek Semiconductor.

References

- [1] Daugman J (2002) How Iris Recognition Works. In: Image Processing. Proceedings of the 2002 International Conference on, vol 31 pp I-33-I-36
- [2] Daugman J (2003) The Importance of Being Random: Statistical Principles of Iris Recognition. *Pattern Recognition*, 36(2): 279–291. doi: 10.1016/s0031-3203(02)00030-4
- [3] Hosseini M S, Araabi B N, Soltanian-Zadeh H (2010) Pigment Melanin: Pattern for Iris Recognition. *Instrumentation and Measurement*. *IEEE Transactions*, 59(4): 792–804
- [4] Wikipedia (2010) Iris Recognition. http://en.wikipedia.org/wiki/Iris_recognition. Nov 2010
- [5] Maltoni D, Maio D, Jain A K et al (2009) *Handbook of Fingerprint Recognition*. Springer, New York
- [6] Ratha N K, Bolle R (2003) *Automatic Fingerprint Recognition Systems*. Springer, Heidelberg
- [7] Wikipedia (2010) Fingerprint. <http://en.wikipedia.org/wiki/Fingerprint>. Accessed 2 November 2010
- [8] Brunelli R, Poggio T (1993) Face Recognition: Features Versus Templates. *Pattern Analysis and Machine Intelligence*. *IEEE Transactions*, 15(10): 1042–1052
- [9] Jain A K, Li S Z (2005) *Handbook of Face Recognition*. Springer, New York
- [10] Wright J, Yang A Y, Ganesh A et al (2009) Robust Face Recognition via Sparse Representation. *Pattern Analysis and Machine Intelligence*. *IEEE Transactions*, 31(2): 210–227
- [11] Jain A, Bolle R, Pankanti S (2002) Introduction to Biometrics. In: Jain A, Bolle R, Pankanti S (eds) *Biometrics*. Springer, New York, pp 1–41. doi: citeulike-article-id: 8006198
- [12] Wikipedia (2010) Retinal Scan. http://en.wikipedia.org/wiki/Retinal_scan. Accessed 5 November 2010

- [13] Department Y P (2010) Fingerprinting. <http://www.yakimapolice.org/police/pages/fingerprinting.htm>. Accessed 4 April 2010
- [14] Jamblichus (2010) A Man's Right Eye as Shown in a Retinal Scan. <http://jamblichus.wordpress.com/tag/michel-foucault>. Accessed 2 October 2009
- [15] Wang X Y, Shan J, Wu D (2008) System and Method for Lens Shading Correction of an Image Sensor Using Splines. US Patent US20090268053
- [16] Nakamura J (2005) Image Sensors and Signal Processing for Digital Still Cameras. Optical Science and Engineering Series. Taylor & Francis
- [17] Davies A, Fennessy P (2001) Digital Imaging for Photographers. 4th edn. Focal Press, Boston
- [18] Buades A, Coll B, Morel J M (2005) A Non-local Algorithm for Image Denoising. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on 20–25 June 2005 vol 62 pp 60–65
- [19] Tomasi C, Manduchi R (1998) Bilateral Filtering for Gray and Color Images. In: Computer Vision, 1998. Sixth International Conference on 4–7 January 1998. pp 839–846
- [20] Gonzalez R C, Woods R E (2002) Digital Image Processing, 2nd Edn. Prentice Hall, New York
- [21] Weisstein E W (2010) B-spline. <http://mathworld.wolfram.com/B-Spline.html>. Accessed 1 January 2010
- [22] Wikipedia (2010) Isaac Jacob Schoenberg. http://en.wikipedia.org/wiki/Isaac_Jacob_Schoenberg. Accessed 1 October 2010
- [23] Wikipedia (2010) Spline (Mathematics). [http://en.wikipedia.org/wiki/Spline_\(mathematics\)](http://en.wikipedia.org/wiki/Spline_(mathematics)). Accessed 1 November 2010
- [24] Wikipedia (2010) B-spline. <http://en.wikipedia.org/wiki/B-spline>. Accessed 1 November 2010
- [25] Shene C-K (2010) Introduction to Computing with Geometry Course Notes. <http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/notes.html>. Accessed 1 July 2010