A Practical Low-Cost Machine Vision Sensor System for Defect Classification on Air Bearing Surfaces

Pichate Kunakornvong[*] and Pitikhate Sooraksa[1]

College of Data Storage Innovation Technology, King Mongkut's Institute of Technology Ladkrabang,
1 Chalongkrung RD, Bangkok, Thailand
[1]Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of
Technology Ladkrabang, 1 Chalongkrung RD, Bangkok, Thailand

In this paper, we present a newly adapted machine vision method and a practical low-cost machine vision sensor for defect classification of the air bearing surfaces (ABSs) of a hard disk drive, which controls the flying height of the recording heads moving above a disk in operation. A defective ABS can cause poor reading and writing performance; hence, it is necessary to verify its integrity before assembling it into the final product. The proposed sensor system was designed and implemented to detect defects by an effective combination of image segmentation and block matrix techniques as well as classifying them using an expert system under dark- and bright-field conditions. Our system processes subregions of interest and sub-blocks in parallel so that they can take advantage of multiple processor cores. From the trial runs, the small fractional error and low average processing time suggested that our proposed system is effective and can be used in an industrial assembly line.

1. Introduction

In a hard disk drive (HDD), the air bearing surface (ABS) is a small but important part of a slider, which is a part of a head gimbal assembly (HGA) that holds recording heads. The location of the ABS on an HGA is illustrated in Fig. 1. The ABS controls the flying height of the slider above a magnetic disk. For a very high density HDD, the recording head on a slider flies extremely close to the magnetic disk with a clearance of only a small fraction of a microinch.[1] Figure 2 shows a schematic of a flying slider over a spinning disk. In its flight, the ABS is lifted by the passive pressure provided by disk platter rotation. The design of a slider has to take into account its weight, flying velocity, and actuator arm skew in order to achieve a uniform flying height. The quality of an HDD is as good as the quality of its ABS; hence, an effective defect detection process is necessary. The types of defects of an ABS focused on in this paper are stains, contaminating particles, and metal scratches. An effective defect detection method is detection by image processing, but a major problem with an image processing system is the inconsistent quality of acquired images in terms of intensity, contrast, and resolution.
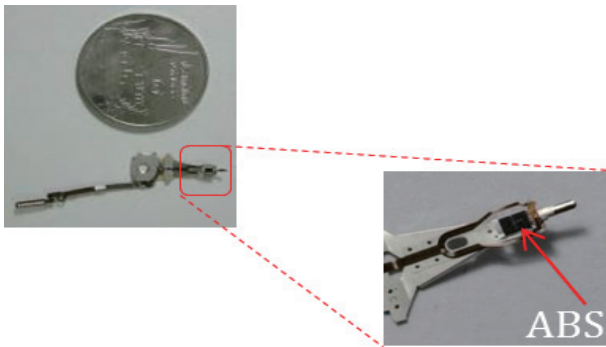
Fig. 1.	(Color online) Air bearing surface on an HGA compared with the size of a one Thai baht coin.
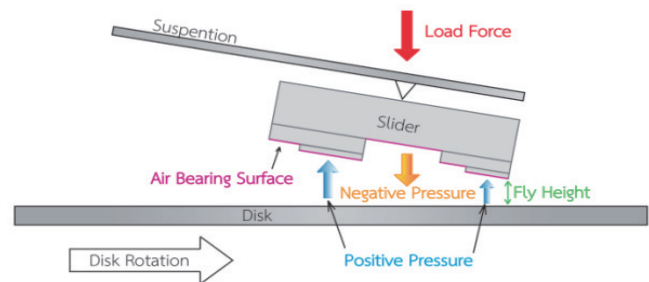

Fig. 2.	(Color online) Schematic for recording head flying.

Currently, offline nanocontamination detection is carried out by automatic force microscopy (AFM) or scanning electron microscopy (SEM). AFM was first introduced in 1996 by Bining *et al.*[2] There are two types of AFM: contact mode and noncontact mode. AFM is not suitable for the inline monitoring of defects because it requires highly skilled human operators to work around the clock as well as lengthy operation time and high operating costs. SEM is usually employed to inspect very small machine parts. The advantages of SEM are high magnification, high resolution, and large depth of field (DOF). The device can achieve a resolution of better than 1 nm. However, its major disadvantage is that it destroys the surface of the scanned object. Thus, SEM is also not suitable for automated inline detection. Other techniques for detecting particulate contamination are based on light scattering, which relies on established laser detector and optics technologies. Most techniques are used for particle counting, as reported in Ref. 3. As for the development of particle counters, Blesener[3] invented a nonimaging laser particle counter, and Bhushan and Chandra[4] proposed a laser particle counter (LPC) and a sampling technique for detecting the size of contaminating particles on a rigid disk drive. Koshevar *et al.*[5] proposed a new nanocontamination monitoring device. None of these light scattering techniques, however, can determine the type of material of the contaminating particles. As for inline detection of other types of ABS defects such as stains, scratches, cracks, or chips, the conventional method is detection by a human operator observing ABSs through a high-magnification microscope. The two major disadvantages of using a human operator are unpredictable human error and slow detection speed.

A good detection method that has been widely used for the inline monitoring of numerous products and processes is a machine vision sensor. Even though machine vision sensor systems have long been utilized in HDD factories for purposes such as QR code checking and mainboard integrity checking, they have not been used for ABS defect detection. At the time of the development of our proposed system, we considered that using a machine vision sensor to detect ABS defects would eliminate human error and increase the number of units processed per hour.

Various defect detection algorithms have been proposed. They can be categorized into four main groups: statistical, structural, spectral and filtering, and model-based algorithms. Examples of applications of these algorithms are shown in Table 1. In this work, a simple yet practical

Table 1
Machine vision algorithms for industrial defect detection.

Approach	Processing algorithm	Industrial defect detection
Statistical algorithm	Gray-level statistic (histogram)	Textile,[6] Texture[7]
	Co-occurrence matrix	Textile[8]
	Principle component analysis	Wood[9]
	Mean shift	Solar wafer[10]
	Weibull	Synthetic aperture radar (SAR) image[11]
Structural algorithm	Edge detection	PZT,[12] Ceramic[13], Flash thermography (FT)[14]
	Morphological operation	Textile,[15–17] Ceramic[18]
Spectral and filtering algorithm	Fourier transform	IC Wafer,[19] Textile,[20] Metal[21]
	Gabor wavelet	Texture[22]
	Gabor filter	Textile[23,24]
	Wavelet transform	Textile[25–27]
	Discrete cosine transform	Textile[28]
Model-based algorithm	Neural network	Textile[29]
	Generic algorithm	Metallic surface[30]
	Support vector machine	Metallic surface[21]
	Independent component analysis	Textile,[31] LCD[32]
	Texem model	Marble[33]

approach for constructing a low-cost machine vision sensor for detecting three types of ABS defects was designed and implemented.

This paper is organized as follows. In Sect. 2, we explain the design of the imaging system. The hybrid defect detection algorithm for our sensor system is discussed in Sect. 3. In Sect. 4, we report the tested detection and classification performances of the sensor. In Sect. 5, we provide conclusions.

2. Imaging System

In this section, the system used for capturing images during inspection is introduced. Because the defects are usually very small, the captured image must collect all of details of the ABS while minimizing noise. A high-resolution industrial CCD camera was used for capturing high-quality images. Our camera selection criteria were a minimum resolution threshold and a maximum lens focal length threshold. The minimum resolution threshold was based on the size of the smallest defect that we wanted to detect. This threshold was calculated using Eq. (1):

$$\#pixel = 2 \times \frac{d_{FOV}}{\mathrm{Re}s},\qquad(1)$$

where $\#pixel$ is the minimum resolution, and d_{FOV} is the width or height of the field of view (FOV). The area of the ABS was around 1×0.8 mm^2 and we aimed to detect a defect as small as 1 μm, so according to Eq. (1), the minimum pixel dimension of the camera should be at least 1600×2000

pixels. This specification was met by a commercially available 5 MP CCD camera, which was capable of capturing images at a dimension of 2048 × 2594 pixels.

On the basis of the designed distance from the lens to the ABS, the focal length of the lens was calculated using Eq. (2):

$$f = \frac{d_{sensor} \times WD}{d_{FOV} + d_{sensor}}, \tag{2}$$

where f is the focal length of the lens, d_{sensor} is the width of the CCD, which was chosen as 5.70 × 4.28 mm^2, and WD is the distance from the lens to the ABS, which was initially designed to be 50 mm. With these values, the lens focal length threshold was calculated to be 42.537 mm. This specification was also met by the same camera, which had a lens of 50 mm focal length. To be able to obtain low-distortion images, we selected a macro lens. The required magnification of the macro lens was calculated using Eq. (3):

$$Mag = \frac{d_{sensor}}{d_{FOV}}, \tag{3}$$

where Mag is the maximum magnification. For our design parameters, the required magnification should not be greater than 5.7×. Thus, we selected a 4× magnification macro lens that produced only −0.073489% distortion. With a 5-mm extension ring, the magnification was increased to 4.3×.

Finally, the illumination for the imaging system was selected. Because the ABS is very small, a wide-length spot light was selected. Two illumination techniques were used, as the bright and the dark field were used to collected images from the ABS for the defect classification. The closed chamber was designed to solve the problem of disturbances by external light.

With the selected imaging system as a CCD camera, lens, and light source, the hardware for our low-cost machine vision sensor system was constructed, a photo of which is shown in Fig. 3. The system was inexpensive compared with an AFM or an SEM. This selected hardware proved to be satisfactory because the acquired images were acceptably clear: two sample images are shown in Sect. 4.
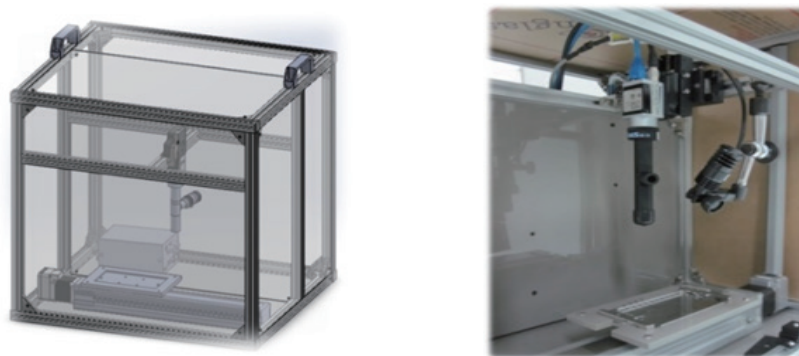

Fig. 3.   (Color online) Inexpensive machine vision sensor system.

3.   Defect Detection and Classification

In this section, we begin by describing the steps in the defect detection procedure in Sects. 3.1–3.5. In the last subsection, Sect. 3.6, we describe the types of defects in our classification scheme and the classification algorithm.

3.1   Extraction of region of interest (ROI)

First, an ABS image was captured by the CCD camera. Then, the captured image was extracted in order to find the ROIs. A well-known extraction method is the Chan–Vese algorithm, which finds a geometric active contour via a level set method. We tried to apply the level set method with our system, but it was not satisfactory owing to not only its low speed and high CPU load for the high-resolution image used in this work but also to the imperfect extraction. The fast level set method, proposed by Y. Li *et al.*[34] and H. Lu and colleagues,[35,36] improved the performance of the level set method and increased the speed of the algorithm. We tried to use this algorithm with our system, but choosing and tuning parameters to satisfy our case were not easy to implement in due time. However, it will be interesting to simplify and integrate the method for further investigation in the future work. Hence, we devised techniques that worked, its function was inspired from the "wand" tool in Photoshop software, and we found it to be satisfactory for our defect detection scheme. The implemented technique was very easy with low complexity. This technique generated the extraction mask by the Euclidean distance function between the reference value and each pixel.

Let us briefly review the Chan–Vese algorithm. This algorithm employs a geometric segmentation approach to minimizing the fitting energy of the input image and the level set function that defines the segmentation. The variational calculus method was used to evolve the level set function. The Chan–Vese model was defined according to the fitting energy function, shown in Eq. (4):

$$F(\varphi) = \mu \left(\int_{\Omega} |\nabla H(\varphi)|\, dx \right)^p + v \int_{\Omega} H(\varphi)\, dx + \lambda_1 \int_{\Omega} |I - c_1|^2 H(\varphi)\, dx$$
$$+ \lambda_2 \int_{\Omega} |I - c_2|^2 (1 - H(\varphi))\, dx, \tag{4}$$

where $F(\varphi)$ is the fitting energy function; μ, v, λ_1, λ_2, and p are the user-selected fitting parameters for a particular class of images; H is the Heaviside function; I is the input image; Ω is the domain of the input image. The terms c_1 and c_2 are the average pixel values of the foreground and background, μ and v are the weights of the total length of the edge contour (boundary) and the total area of the region obtained from segmentation (if the region has a smooth boundary, the weight μ will be large), and λ_1 and λ_2 are the weights of the distance terms and are proportional to the variances of the foreground and background of the image, respectively. The terms $\lambda_1 \int_{\Omega} |I - c_1|^2 H(\varphi)dx$ and $\lambda_2 \int_{\Omega} |I - c_2|^2 (1 - H(\varphi))dx$ were used to represent the uniformity of the region in terms of pixel intensity. Minimizing the sum of these two terms made the foreground and background of the segmentation region as uniform as possible.

We attempted to use this method to extract ROIs. However, after trial runs, it was found to be not suitable for processing our ABS images because it used too much computation time and too many resources. Hence, we developed a simpler algorithm, the procedure of which is described below.

Our algorithm for ROI extraction:

Input: (i) $I(x, y, z)$: the intensity of a pixel at coordinate (x, y) on a color plane z in the color image of $M \times N$ pixels, where z takes a value of 1, 2, or 3, representing red, green, or blue, respectively; and

(ii) x', y', w, h: x' and y' are the coordinates of the upper left hand corner of the user's selected reference area and w and h are its width and height.

Step 1: Determine the reference intensity values ($R(z)$) and tolerance (T) value:
the reference intensity is an intensity value that is used in the process of screening a pixel in or out of the extracted ROI—a pixel with an intensity value not farther away from the reference intensity than the tolerance value is included in the ROI. It is determined as the statistical mode of all intensity values in each color plane of the reference area. Tolerance value is determined as the average value of the three standard deviations of all intensity values in each of the three color planes of the reference area.

Step 2: Create a binary mask $B(x, y)$ for ROI extraction using the following steps:
Step 2a: Calculate the Euclidean distance $d(x, y)$ between $I(x, y, z)$ and $R(z)$,
Step 2b: Generate a binary mask $B(x, y)$ according to the following criteria:

$$d^2(x, y) \leq T, \quad B(x, y) = 1,$$

$$d^2(x, y) > T, \quad B(x, y) = 0.$$

Step 3: Obtain the ROI by masking the image with $B(x, y)$.

3.2   Fuzzy dynamic histogram equalization

In the scheme of image contrast enhancement, the histogram equalization is a well-known technique. Contrast enhancement for images in turbid water[37] is a contrast enhancement method, which is presented the method for correcting the illumination lost by propagation of an underwater image, whereas this work focused on light reflecting, scattering, and light distribution. Then, the fuzzy dynamic histogram equalization[38] was used to enhance the ABS image and minimize the pixel fluctuation without removing defects and noise. This technique consists of the following four steps.

3.2.1   Creation of a fuzzy histogram

A fuzzy histogram is smoother than a crisp histogram and was much more suitable for our application. It is a sequence of real numbers: $h(i)$, $i \in \{0, 1, ..., L-1\}$ where $h(i)$ is the occurrence frequency of gray level i. Assuming a gray level $I(x, y)$ at the coordinate $I(x, y)$ and a fuzzy number $\tilde{I}(x, y)$, the fuzzy histogram was computed using Eq. (5):

$$h(i) \leftarrow h(i) + \sum_x \sum_y \mu_{\tilde{I}(x,y)i}, k \in [a, b], \tag{5}$$

where $\mu_{\tilde{I}(x,y)i}$ is a triangular fuzzy membership function defined as $\mu_{\tilde{I}(x,y)i} = \max\left(0.1 - \dfrac{|I(x,y) - i|}{4}\right)$ and $[a, b]$ is the support of the membership function.

3.2.2   Partitioning of the histogram

The histogram obtained was partitioned into multiple subhistograms that were individually equalized in the next step. Partitioning was carried out by finding the first- and second-order derivatives of the fuzzy histogram using Eqs. (6) and (7), respectively:

$$\dot{h}(i) = \frac{dh(i)}{di} \triangleq \frac{h(i+1) - h(i-1)}{2}, \tag{6}$$

$$\ddot{h}(i) = \frac{d^2h(i)}{di^2} \triangleq h(i+1) - 2h(i) + h(i-1), \tag{7}$$

where $\dot{h}(i)$ and $\ddot{h}(i)$ are the first- and second-order derivatives of the fuzzy histogram $h(i)$, respectively, corresponding to the ith intensity level. Then, the partition points were the intensities $(m_0, m_1, ..., m_{n+1})$ of all of the local maximum frequencies, which were calculated using Eq. (8),

$$i_{\max} = i \forall \left\{\dot{h}(i+1) \times \dot{h}(i-1) < 0, \ \ddot{h}(i) < 0\right\}. \tag{8}$$

In the range $[\tilde{I}_{\min}, \tilde{I}_{\max}]$ of the original fuzzy histogram, the subhistograms $[\tilde{I}_{\min}, m_1]$, $[m_1+1, m_2]$, ..., $[m_n+1, \tilde{I}_{\max}]$ were partitioned with these points.

3.2.3   Dynamic histogram equalization of the subhistogram from the previous stage

The new intensity level can be expressed in the form

$$y(j) = m_{i-1} + (m_i - m_{i-1}) \sum_{k=m_{i-1}}^{j} \frac{h(k)}{M_i}, \tag{9}$$

where $y(j)$ is the new intensity level corresponding to the jth intensity level on the input image, m_i is the ith local maxima in the original fuzzy histogram and $i = [1, 2, ..., n+1]$, $h(k)$ is the histogram value at the kth intensity on the fuzzy histogram, and $M_i = \sum_{k=m_{i-1}}^{m_i} h(k)$ is the total number of pixels contained in the ith partition of the fuzzy histogram.

3.2.4   Normalization of the equalized histogram

We then normalized histogram equalization as

$$g(x, y) = \frac{\mu_i}{\mu_o} f(x, y), \tag{10}$$

where μ_i and μ_o are the mean brightness values of the input image and the equalized image ($f(x, y)$), respectively, obtained from the previous stage.

3.3   Generation of a block matrix

The block matrix proposed in Ref. 39 was generated in the form of a two-dimensional array from the tilling process. The tilling process divided the $M \times N$ image into $m \times n$ image blocks of size $p \times q$. Each member in the $m \times n$ image block represented an average intensity of each sub-block, calculated using Eqs. (11) and (12).

$$Y(i, j) = \sum_{k=pi}^{p(i+1)-1} \sum_{l=qj}^{q(j+1)-1} I(k, l), \tag{11}$$

$$B(i, j) = \begin{cases} 1 & if \quad 0 \le Y(i, j) < \bar{X} - 10 \\ 2 & if \quad \bar{X} - 10 \le Y(i, j) < \bar{X} + 10, \\ 3 & if \quad \bar{X} + 10 \le Y(i, j), \end{cases} \tag{12}$$

where $Y(i, j)$ is a tile image generated with a block of size $p \times q$, $B(i, j)$ is the $m \times n$ block matrix represented by two-bit quantization values between levels 1 (01) and 3 (00), and $\bar{X}$ is the mean intensity of the equalized image from the previous stage.

By using Eqs. (11) and (12) , the input image was resized by the tilling process and the block matrix corresponding to the tilling version was obtained. Note that the tilling process has an outstanding advantage for noise removal because the average intensity of the block is employed to represent the value of the block.

3.4   Extraction of defect features

The block matrix B from the previous step can be represented as a function that assigns a quantization level to each block in the domain of the image: $I\colon L_r \times L_c \rightarrow Q$. The terms $L_r = \{0, 1, ..., N_r-1\}$ and $L_c = \{0, 1, ..., N_c-1\}$ are the spatial domains of row and column dimensions, and $Q = \{1, 2, 3\}$ is the domain of quantization levels.

To extract defect features, we used a texture analysis technique called the gray-level co-occurrence matrix (GLCM).[40] It is based on repeated intensities of pixels in the image and can be defined using Eqs. (13) and (14)

$$
\begin{aligned}
P(i, j; d, 0°) &= \#\left\{ \begin{array}{l} ((r, c), (r', c')) \in (L_r \times L_c) \times (L_r \times L_c) \\ r' - r = 0, |c' - c| = d, I(r, c) = i, I(r', c') = j \end{array} \right\}, \\[4pt]
P(i, j; d, 45°) &= \#\left\{ \begin{array}{l} ((r, c), (r', c')) \in (L_r \times L_c) \times (L_r \times L_c) \\ (r' - r = d, c' - c = d) \ or \ (r' - r = -d, c' - c = -d) \\ I(r, c) = i, I(r', c') = j \end{array} \right\}, \\[4pt]
P(i, j; d, 90°) &= \#\left\{ \begin{array}{l} ((r, c), (r', c')) \in (L_r \times L_c) \times (L_r \times L_c) \\ |r' - r| = d, c' - c = 0, I(r, c) = i, I(r', c') = j \end{array} \right\}, \\[4pt]
P(i, j; d, 135°) &= \#\left\{ \begin{array}{l} ((r, c), (r', c')) \in (L_r \times L_c) \times (L_r \times L_c) \\ (r' - r = d, c' - c = -d) \ or \ (r' - r = d, c' - c = -d) \\ I(r, c) = i, I(r', c') = j \end{array} \right\},
\end{aligned} \tag{13}
$$

where the distance matrix can be explicitly defined as

$$d\left((r, c), (r', c')\right) = \max\{|r - r'|, |c - c'|\}. \tag{14}$$

Next, the correlation feature used to identify the defected ABS or nondefected ABS was calculated using Eq. (15):

$$Corr = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} \frac{(i - \mu_i)(j - \mu_j) \cdot C(i.j)}{\sigma_i \sigma_j}, \tag{15}$$

where μ_i is defined as $\mu_i = \sum_{i=1}^{M} i \sum_{j=1}^{N} C(i.j)$, μ_j is defined as $\mu_j = \sum_{j=1}^{N} j \sum_{i=1}^{M} C(i.j)$, σ_i is defined as $\sigma_i = \sum_{i=1}^{M} (i - \mu_i)^2 \sum_{j=1}^{N} C(i.j)$, and σ_j is defined as $\sigma_j = \sum_{j=1}^{N} (j - \mu_j)^2 \sum_{i=1}^{M} C(i.j)$. Note that the index *Corr* measures the linear dependence of the elements in the matrix between the specified elements relative to each other.

3.5   Detection of defect

Defects on the ROIs were detected on the basis of the characteristics of the block matrix, which was generated from the previous step. As mentioned previously, the algorithm began with an image that was partitioned into N sub-ROI regions and then the procedure was repeated in Sects. 3.1, 3.2, 3.3, and 3.4. The search for the defects in each sub-ROI ran in parallel architecture and was synchronized.

In order to identify the defects inside the ROIs, we used Euclidean distance and the selection threshold using the method of the minimum summation of two types of errors, namely, false acceptance (e_1) and false rejection (e_2), from the threshold set. The threshold is defined as

$$\eta = \min(e_1 + e_2). \tag{16}$$

The defective and nondefective ABS were classified by means of the training data set as

$$\begin{array}{ll} Nondefect & \|f\| \geq \eta, \\ Defect & \|f\| < \eta, \end{array} \tag{17}$$

where $\| \ \|$ is the Euclidean distance, and f is the feature parameter obtained from the co-occurrence matrix of the block matrix.

3.6   Classification of defect

In this work, three types of defects—stains, contaminating particles, and metal scratches—were identified. Each defect type had strong physical illumination characteristics, as shown in Table 2; the expert system was designed for classification based on the reflection characteristics. A flow chart of the expert system is shown in Fig. 4.

Table 2
Physical illumination characteristics of the defect on the ABS.

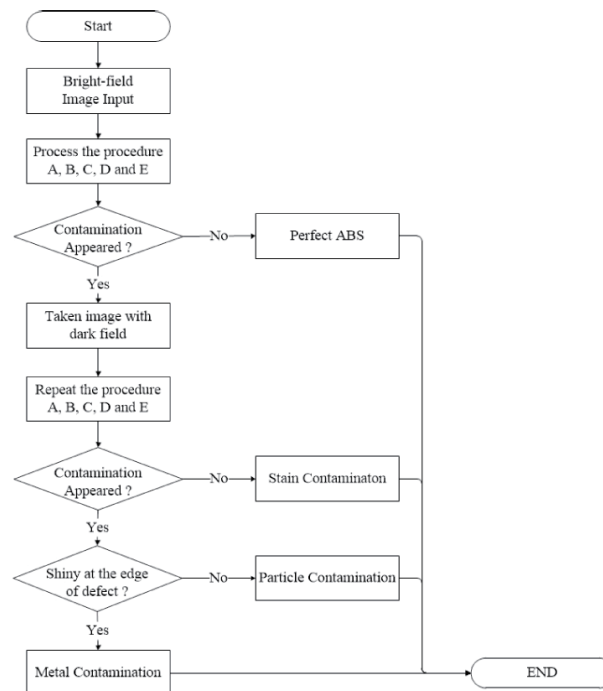| Defect Type	Illumination techniques	
	Bright field	Dark field
Stain	Rainbow	Not shown
Particle	Unshiny	Brighter at the edge of the defect
Metal scratch	Shiny	Shiny at the edge of the defect

Fig. 4.   Designed expert system.

4.   Results and Discussion

In order to test the designed system, the ABS images were first captured from the designed machine vision system as shown in Fig. 5.   The figure shows Quadrants #1 and #2 on the ABS images.   The results of Sect. 3.1 are shown in Figs. 6 and 7.   In Fig. 6, the results of the corresponding ROIs on the ABS using the level set method based on the Chan–Vese model[34–36] are presented.   The specific values used herein for Eq. (4) were $\lambda_1 = \lambda_2 = 1$, $\mu = 0.1 \times 255^2$, and $v = 0$. As can be seen, the extracted ROIs were not perfectly removed from the background of the original image.   Some unwanted background is shown as something protruding at the bottom right in Fig. 6(a) of the images appeared.   Likewise, some unwanted background also appears at the bottom left of Fig. 6(b).   The unwanted background indicates that the algorithm suggested in Refs. 34–36 may not suitable for this case.   Otherwise, the values of the parameters need to be optimized.   Perhaps numerical steps for the integral parts of Eq. (4) need to be fine-tuned until satisfactory results are finally achieved.   It takes time to proceed with the fine tuning.   In order to cope with this difficulty,
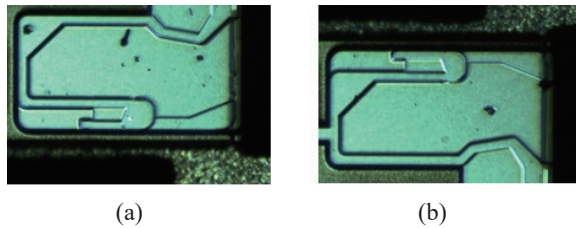
(a)　　　　　　　　　　　(b)

Fig. 5.　(Color online) Some parts of the ABS image captured from the designed system: (a) Quadrant #1 of the ABS and (b) Quadrant #2 of the ABS.

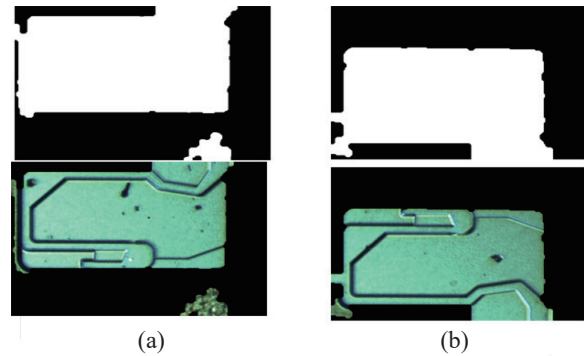(a)　　　　　　　　　　　(b)

Fig. 6.　(Color online) Extraction of ROI using the level set method[34] in Sect. 3.1: (a) results on Quadrant #1 and (b) results on Quadrant #2.
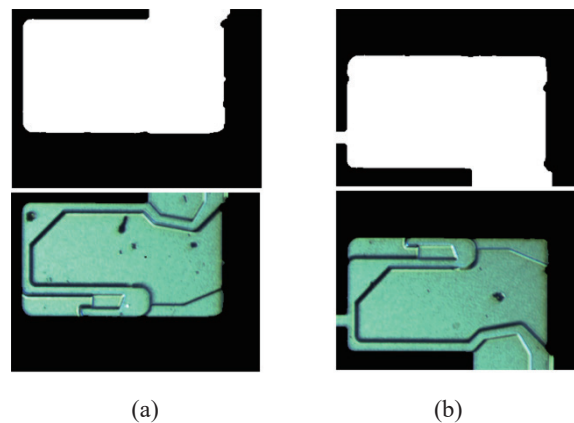

(a)　　　　　　　　　　　(b)

Fig. 7.　(Color online) Extraction of ROI using the proposed method in Sect. 3.1: (a) results on Quadrant #1 and (b) results on Quadrant #2.

the proposed algorithm presented in Sect. 3.1 was presented as an alternative method. Figure 7 shows the results obtained from using the proposed method. As one can see, the extracted ROIs are clearer and better than the ROIs from the level set method. Unwanted background was removed. In addition, the proposed method had a better processing time, as shown in Table 3. The processing time for using the Chan–Vese model was 6.15 s vs 1.62 s for the proposed method. The ROI was extracted from the image size of 1942 × 2590 pixels with the pre-defined reference area for the size of 40 × 40 pixels.

　　Figures 8(a) and 8(c) show the results for Sect. 3.3 without processing through the fuzzy dynamic histogram equalization, as described in Sect. 3.2. There is still noise in those figures; however, Figs. 8(b) and 8(d) show that the fluctuations due to noise were removed by using the procedure in Sect. 3.2. The images in the subfigures are clearer than those obtained without using the procedure in Sect. 3.2. Note that the support membership function used in this procedure is a triangular one in the form [0, 1, 2, 3, 4, 5, 4, 3, 2, 1, 0]. Note also that, in Sect. 3.3, the tiled image was generated from the ROI image with the block size containing 30 × 30 pixels. Figure 9 shows the zoom version of Fig. 8(d), whereas Fig. 10 shows the corresponding numbering values of the block matrix (shown in the red box of Fig. 9) calculated from Eqs. (5) and (6). Numbers 2 and 3 in

Table 3
Processing time.

Procedure	Algorithm	Average processing time (s)
Extraction of ROI (Sect. 3.1)	Chan–Vase[34–36]	6.15
	Proposed	1.62
Histogram equalization (Sect. 3.2)	Fuzzy DHE	0.36
Block matrix (Sect. 3.3)	Block matrix	0.51
Extraction of defect feature (Sect. 3.4)	GLCM without block matrix	9.58
	GLCM with block matrix	1.01
Detection of defect (Sect. 3.5)	Euclidean distance and threshold	3×10^{-6}
Classification of defect (Sect. 3.6)	Expert system	7.000006
Human expertise	Visual inspection using human eyes	5–15

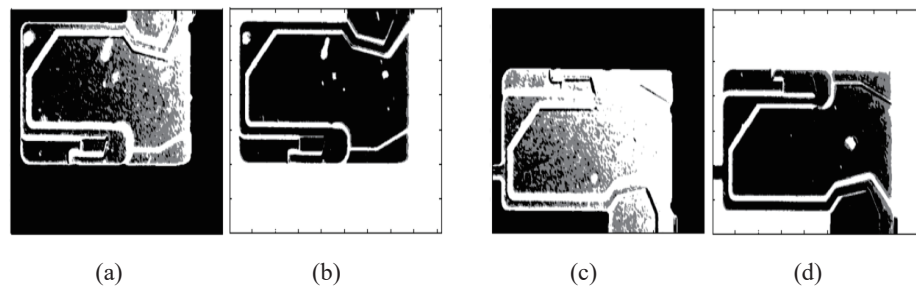| (a) | (b) | (c) | (d) |

Fig. 8.   Results for Sect. 3.3: (a), (c) without Sect. 3.2 and (b), (d) with Sect. 3.2 (better images).
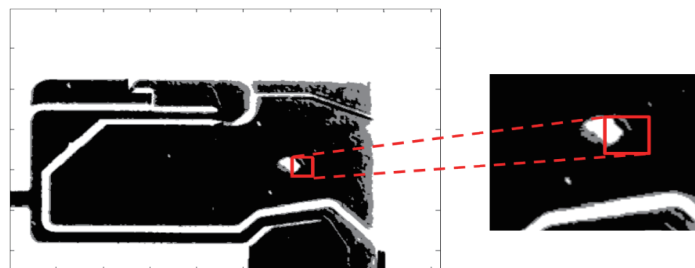

Fig. 9.   (Color online) Zooming results from Sect. 3.3 in Fig. 8(d).

Fig. 10 indicate the defect areas. For Sect. 3.4, the GLCM features were calculated from the co-occurrence matrices with the distance $d = 1$ with $\theta = 0, 45, 90$, and $135°$. Figure 11 shows the plots of the defect detection using the Euclidean distance of the correlation features and the selected threshold presented in Sect. 3.5. These thresholds were calculated automatically and adaptively because the errors had been dynamically changed. The results from this procedure are presented in Tables 3 and 4.

Table 4 shows the results for the three classified types obtained from defect detection, which were stains, contaminating particles, and metal scratches. Two types of error, false rejection and false acceptance, are shown in the table. False rejection means that the expert system identifies the perfect ABS as the defected ABS. On the other hand, false acceptance means that the expert

2	2	2	2	2	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	3	3	2	2	1	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	2	1	1	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	1	1	1	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	2	1	1	1	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	2	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	1	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	3	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	3	3	2	1	1	1	2	2	2	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	3	3	2	1	1	1	2	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	3	3	3	2	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	3	3	3	2	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	3	3	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 10.   Sample values of the block matrix (red box in Fig. 9).
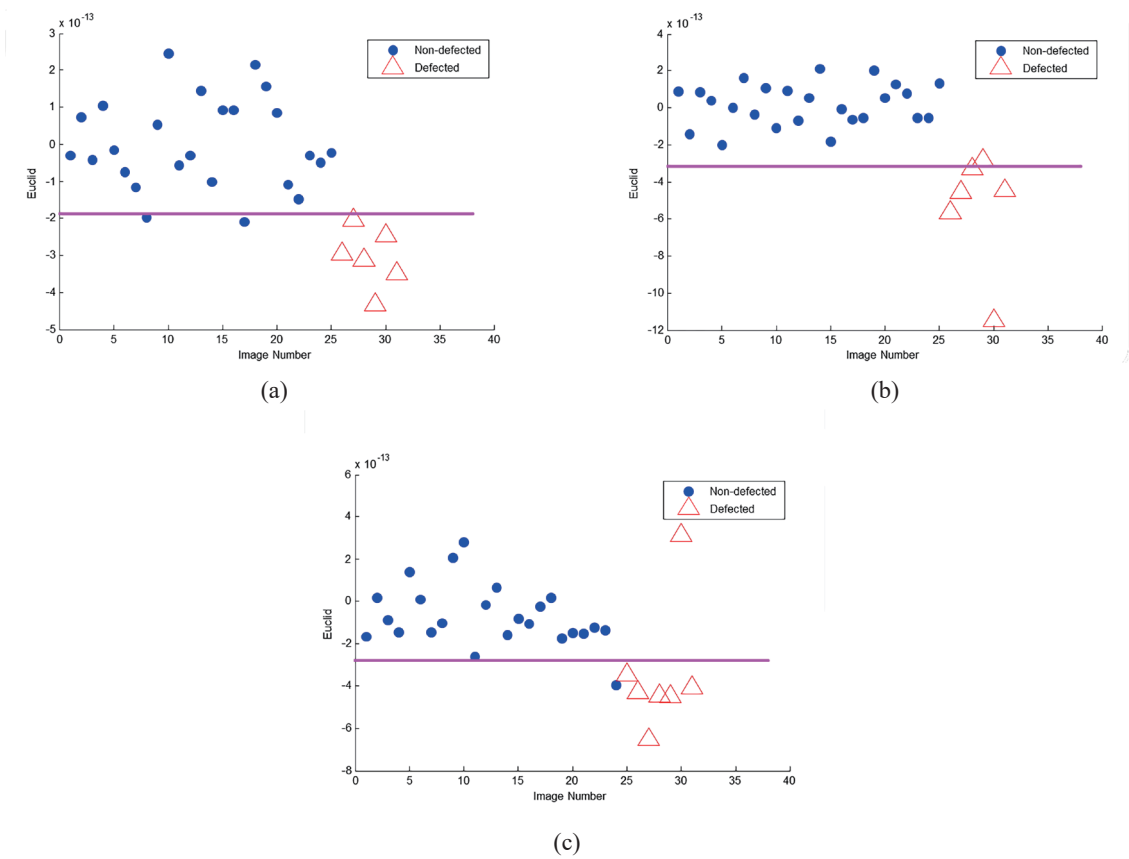

(a)

(b)

(c)

Fig. 11.   (Color online) Euclidean distances and thresholds of defect detection with the correlation feature on the selected ROIs: (a) stains, (b) particles, and (c) metal scratches.

Table 4
Classification results.

Error type	Number of samples	Error
False rejection (stain)	24	2
False acceptance (stain)	7	0
False rejection (particle)	25	0
False acceptance (particle)	6	0
False rejection (metal scratch)	25	1
False acceptance (metal scratch)	6	1
Group missing	19	3

system identifies the defected ABS as a perfect one. Over-rejection may not cause as much trouble as over-acceptance. This is because over-acceptance may accept the defective parts that will be delivered to customers. A defective sample found by a customer would cause the rejection of the whole lot of the product shipment, and the products would be returned to the factory, creating a bad reputation for the company. This is serious since it is necessary to completely prevent false acceptance errors. In Table 4, all of the results were acceptable, except for the case of the false acceptance for metal scratches. Hence, false acceptance in the case of metal scratches is a concern. More rule bases for the case of metal scratches need to be adjusted or added, which would be a challenging classification enhancement for future work. Nevertheless, for defect detection only (regardless of classification), the method works very well, as it can reject all defect types.

In Table 3, the total run time for all procedures was around 7 s, which was in an acceptable operation range compared with using human operators (around 5–15 s). The computer used in this test was an Intel Core i7 CPU with 4 GB RAM. For speed performance comparison, if we used the proposed inspection without the procedure in Sect. 3.3, the run time was around 13 s, which was 2 times higher than that with the procedure in Sect. 3.3. This is why the procedure in Sect. 3.3 was introduced and included. As one can see, the advantage of the proposed method not only speeded up the process but also performed as well as human visualization. If the defect can be seen in an image, the detection result is still accurate, even though the lighting environment has changed. Note that one may conjecture about the robustness of the method over the lighting conditions. However, for a well-developed industrial line, the lighting conditions can be completely controllable. In our case, we set up the image acquisition procedure using an industrial camera inside the well-designed lighting chamber, and there was no need to check for other lighting conditions in this situation.

5.   Conclusions

A practical machine vision design and implementation have been illustrated in this paper. A simple yet effective method was proposed to identify the defects on the ABS of a HGA. The six simple procedures suggested in this paper were ROI extraction, fuzzy dynamic histogram equalization, block matrix generation, extraction of defect features, defection detection, and classification of the defects. The key idea for speeding up automatic inspection was to divide the area of investigation into each block matrix. In doing so, searching areas can be divided as desired by the designer. The concept of using sub-blocks and sub-ROI processing in parallel was revealed

to be effective, as demonstrated by the results, because the concept markedly helps to reduce the size of the input matrix before feeding to the feature extraction procedure. As the size of the matrix is reduced, the run time for extracting the features of the blocks is also reduced.

Finally, the expert system for classification was designed from the illumination characteristics on the ABS. The system was tested by using normal samples and samples containing three types of defects: microcontamination with stains, contaminating particles, and metal scratches. The results were seen to be satisfactory for defect detection and classification. However, for the classification, the rules bases for classification may need to be modified in the case of metal scratches owing to the existence of false acceptances in the test. In general, the results can be verified by the proposed expert system, indicating that they were good as these visualized by the human eye. However, this process is an automatic operation in the production line, which can replace human operators. The human operators can then perform more intelligent and healthier tasks. Further work needs to be carried out on the classification system in order to classify more types of defects in the manufacturing system. The searching algorithm itself can also be enhanced by using an intelligent search, such as chaotic path, pheromone tracking, or other combinations of heuristic and analytic approaches. These topics may be viewed as subregional searches inside the blocks for fine tuning and are still a challenge.

Acknowledgments

References

1   R. Wood: J. Magn. Magn. Mater. **321** (2006) 555.
2   G. Binnig, C. F. Quate, and Ch. Gerber: Phys. Rev. Lett. **56** (1986) 930.
3   J. L. Blesener: U.S. Patent No. 5 (1992).
4   B. Bhushan and S. Chandra: IEEE Trans. Magn. **35** (1999) 956.
5   S. Kochevar, T. Pietrykowski, and D. Rodier: Proc. SPIE 8492 Opt. Syst. Contam., Effects, Measurements, and Control 2012., eds. S. A. Straka, N. Carosso, and J. Egges (SPIE, San Diego, 2012) pp. 84920H–84920H.
6   S. Singh and M. Kaur: IOSR J. Eng. **2** (2012) 429.
7   D. Popescu, R. Dobrescu, and M. Nicolae: Int. J. Circuits Syst. Sig. Process **1** (2007) 79.
8   H. Zuo, Y. Wang, X. Yang, and X. Wang: Proc. 5th Int. Cong. Image and Signal Processing (CISP) 2012, ed. Q. Chen (IEEE, Chongqing, Sichuan, 2012) pp. 876–880.
9   L. Tomczak, V. Mosorov, D. Sankowski, and J. Nowakowski: Proc. 9th Int. Conf. CAD Systems in Microelectronics 2007, ed. M. Lobur (IEEE, Lviv-Polyana, 2007) pp. 454–456.
10 D. M. Tsai and J. Y. Luo: IEEE Trans. Ind. Inf. **7** (2011) 125.
11 I. B. Ayed, N. Hennane, and A. Mitiche: IEEE Trans. Image Process. **15** (2006) 3431.
12 W. Withayachumnankul, P. Kunakornvong, C. Asavathongkul, and P. Sooraksa: Int. J. Adv. Manuf. Technol. **64** (2013) 1.
13 G. M. Atiqur Rahaman and Md. M. Hossain: Int. J. Comput. Sci. Inf. Secur. (IJCSIS) **1** (2009) 22.
14 F. C. Sham, N. Chen, and L. Long: J. Insight-Winston the Northampton **50** (2008) 240.
15 K. L. Mak, P. Peng, and K. F. C. Yiu: Image Vision Comput. **27** (2009) 1585.
16 B. Mallik-Goswami and A. K. Datta: Text. Res. J. **70** (2000) 758.
17 K. Y. Song, J. Kittler, and M. Petrou: Image Vision Comput. **14** (1996) 667.
18 H. Elbehiery, A. Hefnawy, and M. Elewa: Int. J. Comput. Electr. Autom. Control Inf. Eng. **1** (2007) 1488.
19 H. Liu, W. Zhou, Q. Kuang, L. Cao, and B. Gao: IEEE Trans. Semicond. Manuf. **23** (2010) 141.

20 C. H. Chan and G. K. Pang: IEEE Trans. Ind. Appl. **36** (2000) 1267.
21 Z. Xue-wu, D. Yan-qiong, Lv. Yan-yun, S. Ai-ye, and L. Rui-yu: Expert Syst. Appl. **38** (2011) 5930.
22 V. Asha, N. U. Bhajantri, and P. Nagabhushan: Proc. 5th Int. Conf. Information Processing (ICIP) 2011., eds. K. R. Venugopal and L. M. Patnaik (Springer Berlin Heidelberg, Bangalore, 2011) pp. 548–553.
23 A. Bodnarova, M. Bennamoun, and S. Latham: Pattern Recognit. **35** (2002) 2973.
24 A. Kumar and G. K. Pang: IEEE Trans. Ind. Appl. **38** (2002) 425.
25 X. Yang, G. Pang, and N. Yung: IEE Proc. Vision, Image and Signal Processing **152** (2005) 715.
26 A. Fathi, A. H. Monadjemi, and F. Mahmoudi: Int. J. Soft Comput. Eng. (IJSCE) **2** (2012) 30.
27 M. Ghazvini, S. A. Monadjemi, N. Movahhedinia, and K. Jamshidi: World Acad. Sci. Eng. Technol. **3** (2009) 773.
28 M. A. Aziz, A. S. Haggag, and M. S. Sayed: Proc. Communications, Signal Processing, and their Applications (ICCSPA) 2013, ed. M. El-Tarhuni (IEEE, Sharjar, 2013) pp. 1–4.
29 S. Rimac-Drlje, A. Keller, and Z. Hocenski: Proc. IEEE Int. Symp. Industrial Electronics 2005, eds. N. Peric and K. Jezernik (IEEE, Dubrovnik, 2005) pp. 1255–1260.
30 Y. Frayman, H. Zheng, and S. Nahavandi: J. Adv. Comput. Intell. Intell. Inf. **10** (2006) 281.
31 O. G. Sezer, A. Ertüzün, and A. Erçil: Pattern Recogn. Image Anal. **14** (2004) 303.
32 D. M. Tsai, Y. H. Tseng, S. M. Chao, and C. H. Yen: Pattern Recognit. **39** (2006) 1679.
33 X. Xie and M. Mirmehdi: Pattern Recogn. Image Anal., eds. S. Singh, M. Singh, C. Apte, and P. Perner (Springer Berlin Heidelberg, UK, 2005) pp. 404–413.
34 Y. Li, H. Lu, L. Zhang, S. Yang, and S. Seiichi: Appl. Mech. Mater. **103** (2012) 705.
35 H. Lu, Y. Li, S. Nakashima, S. Yang, and S. Seiichi: Disaster Adv. **5** (2012) 796.
36 H. Lu, B. Li, J. Zhu, Y. Li, Y. Li, X. Xu, L. He, X. Li, J. Li, and S. Serikawa: Concurrency Comput.: Pract. Experience (2016).
37 H. Lu, Y. Li, L. F. Zhang, and S. Seiichi: J. Opt. Soc. America A, JOSA A. **32** (2015) 886.
38 D. Sheet, H. Garud, A. Suveer, M. Mahadevappa, and J. Chatterjee: IEEE Trans. Consum. Electron. **56** (2010) 2475.
39 Y. G. Yoon, S. L. Lee, C. W. Chung, and S. H. Kim: Comput. Ind. Eng. **55** (2008) 567.
40 R. M. Haralick: Proc. IEEE **67** (1979) 786.

About the Authors

Pichate Kunakornvong received his B.Eng. and M.Eng. degrees in Information Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2005 and 2009, respectively. Currently, he is a Ph.D. student at KMITL in the School of Data Storage Innovation. His research interests include image processing, computer vision, and pattern recognition.

Pitikhate Sooraksa received his B.Ed. (Hons) and M.Sc. degrees in Physics from Srinakharinwirot University, Thailand, an M.S. degree from George Washington University (1992), and a Ph.D. degree from the University of Houston (1996), USA, both in Electrical Engineering. He is currently Associate Professor of Electrical Engineering in the Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Thailand. His research interests include IT-mechatronics, the development of rapid prototypes in embedded systems, and computer-aided control.