Journal of Imaging Science and Technology® 57(1): 010505-1–010505-12, 2013.

Feature-Based Image Segmentation

Meng-Hsiun Tsai, Yung-Kuan Chan*, and An-Mei Hsu
Department of Management Information Systems, National Chung Hsing University, Taichung 402, Taiwan, ROC
E-mail: ykchan@nchu.edu.tw

Chia-Yi Chuang
Department of Computer Science and Engineering, National Chung Hsing University, Taiwan, ROC

Chuin-Mu Wang
Department of Electronic Engineering, National Chin-Yi University of Technology, Taiwan, ROC

Po-Whei Huang
Department of Computer Science and Engineering, National Chung Hsing University, Taiwan, ROC

Abstract. *The real world abounds with textured surfaces. Texture-based object segmentation is one of the early steps towards identification of surfaces and objects in an image. In this article, a feature-based segmentation (FBS) method is provided to isolate objects that consist of similar texture patterns from an image based on the following features: inverse difference moment of gray-level co-occurrence matrix, contrast of Tamura, and gradient. In this article, a genetic algorithm is also provided to decide the most suitable values of the parameters used in the FBS method. The experimental results show that the FBS method can provide expressive segmentation results. © 2013 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2013.57.1.010505]

INTRODUCTION

Texture is one of the most important attributes used in image analysis and pattern recognition. It provides surface characteristics for the analysis of many types of image including natural scenes, remotely sensed data, and biomedical modalities. Hence, it plays an important role in the human visual system for recognition and interpretation. Although there is no formal definition of texture, the patterns can be the result of physical surface properties such as roughness, smoothness, coarseness, and regularity, or oriented strands which often have a tactile quality, or they can be the result of reflectance differences such as the color on a surface.

In many machine vision and image processing algorithms, simplifying assumptions are made from the uniformity of intensities in local image regions. However, real objects do not often exhibit regions of uniform intensity. For example, a wooden surface is not uniform but contains variations of intensity which form certain repeated patterns called texture patterns. This article proposes a method to segment objects, each with a similar texture pattern, from an image.

Gray-level co-occurrence matrix (GLCM) texture measurements[19] have been the workhorse of image texturing. GLCM is a tabulation writing down how often every particular pair of gray levels in the pixel pairs, separated by a certain distance along a certain direction, occurs in an image. Various statistical and information theoretic properties of the co-occurrence matrices can be extracted as textural features (e.g., features such as homogeneity, coarseness, or periodicity), as introduced by Haralick. The features generated by this technique are usually called Haralick features.

Tamura & Mori[29] also proposed six texture features corresponding to human visual perception: coarseness, contrast, directionality, line-likeness, regularity, and roughness. They performed experiments to test the significance of the features and found that the first three features were very important. That is, they correlate strongly with the human perception.

Haralick features[18] and Tamura features[29] are invariable or tolerant to the variation of optic parameters.[5] Hence, the segmentation method proposed in this article will use them to describe the textures of an image. Many image segmentation methods[27,33] detect edges by analyzing pixel gradients. Most of them use traditional gradient operators, such as Roberts, Sobel, and Prewitt Laplacian operators.[2] However, traditional gradient operators are known to be adversely affected by noise; they are not suitable for computation of the contour gradient of a object with complex texture pattern. In this article, a texture-based gradient operator is hence provided. We name the feature, computed by the texture-based gradient operator, a gradient feature. Based on Haralick features, Tamura features, and the gradient feature, in this article, a feature-based segmentation (FBS) method is presented to isolate objects that consist of similar complex texture patterns. This article also uses the genetic-based

* Corresponding author.

Received Nov. 23, 2011; accepted for publication Mar. 27, 2013; published online May 29, 2013.
1062-3701/2013/57(1)/010505/12/$20.00

parameter selector (GBPS)[26] to decide the most suitable values of the parameters used by the FBS method.

The dominant approach in the analysis of texture-based object segmentation is to construct a description of the local neighborhood around each pixel, and then to compare this descriptor to the descriptors of nearby points. This approach is referred to as "patch-based". However, the gray values of two neighboring patches from the same texture could be very different, and more elaborate descriptors are required.[31] In this article, GLCM texture measurements and Tamura features are used to describe the textures of an image, and then the texture-based object segmentation problem, segmenting the objects that have similar texture patterns, is therefore transformed into a contour-based segmentation problem,[23] segmenting the objects that have similar gray levels.

RELATED WORK

This section will briefly review some techniques that will be used by the FBS method, and the CSGV (composite sub-band gradient vector) based image segmentation method[21], the performance of which will be compared with the FBS method.

CSGV-based image segmentation method

Huang and Dai[21] proposed a texture descriptor, called the composite sub-band gradient vector (CSGV) descriptor. The CSGV descriptor combines the techniques of wavelet decomposition[8,9] and gradient vector.[11,14] The discrete wavelet transform (DWT) uses a low-pass filter (L) and a high-pass filter (H) to divide an image into four different frequency bands; the lowest frequency band can be repeatedly split in the same way at half the rate of the previous frequency. The CSGV descriptor decomposes an image into four frequency bands (LL, HL, LH, and HH) by one-level DWT and then constructs the gradient vectors of the four frequency bands as the feature vectors of the image. The gradient vectors of the four frequency bands are named as SVG1, SVG2, SVG3, and SVG4, respectively.

Huang and Dai[21] also applied the CSGV descriptor to extract objects from an image based on their textures; we call it the CSGV-based image segmentation method. The CSGV-based image segmentation method includes three stages — split, merge, and boundary refinement.

Split stage: This stage is to divide an original image into quadrants with homogeneous texture via DWT. For each quadrant, if the Euclidean distance between the CSGVs of any two subquadrants is less than a threshold, the quadrant is regarded as a homogeneous quadrant; otherwise, the quadrant is defined as non-homogeneous and the quadrant will be repeatedly split into subquadrants until each quadrant is homogeneous or consists of 16×16 pixels.

Merge stage: If the Euclidean distance between two neighboring quadrants is less than a threshold, both of them will be combined into one.

0	0	0	1	2
1	1	0	1	1
2	2	1	0	0
1	1	0	2	0
0	0	1	0	1

(a) A 5×5 image I

4	2	0
2	3	2
1	2	0

(b) The co-occurrence matrix

Figure 1. An example of co-occurrence matrix.

Boundary refinement stage: This stage is to smooth the boundaries of objects. Each pixel P in an edge block will be re-classified into the neighboring block of the edge block that has the minimum distance from the virtual block of which the center is at P.

GLCM

For a given image I, a co-occurrence matrix C will be generated. The element C_{ij} of the ith row and the jth column of C counts the number of times a pixel with gray level i occurs at a position relative to another pixel with gray level j. For example, if there are three distinct gray levels 0, 1, and 2 in the image I shown in Figure 1(a), and the specified relative position is "lower right", the co-occurrence matrix C of I is shown in Fig. 1(b).

C is generally normalized by the total number of pixels so that each element in C is between 0 and 1; we name C a gray-level co-occurrence matrix. Let K be the maximum gray level in I and $\mu = \sum_i^K \sum_j^K C_{ij}/K^2$ be the mean of the elements in C. To analyze the gray-level co-occurrence matrix C used to categorize the textures of an image, some statistical parameters used as a set of descriptors are computed as follows.[7]

(a) The energy $\sum_i^K \sum_j^K C_{ij}^2$ of C is a measure of textural uniformity of an image. The energy reaches its highest value when the gray-level distribution has either a constant or a periodic form. A homogeneous image contains very few dominant gray tone transitions; therefore the matrix C for this image will have fewer entries of a larger magnitude resulting in a greater value for the energy feature.

(b) The entropy $-\sum_i^K \sum_j^K C_{ij} \log C_{ij}$ of C measures the disorder of an image and it achieves its largest value when all the elements in matrix C are equal. When the image is not texturally uniform, many GLCM elements have smaller values, which imply that the entropy is larger. Therefore, the entropy is inversely proportional to the GLCM energy.

(c) The contrast $\sum_i^K \sum_j^K (i-j)^2 C_{ij}$ of C measures the local variations of illumination in I. If the contrast values differ a lot in a given window, there is a set of sudden strong illumination changes in the local area and it always corresponds to an edge.

(d) The inverse difference moment $\sum_i^K \sum_j^K \frac{1}{|i-j|^d} C_{ij}$, $i \neq j$ of I measures image homogeneity. This parameter achieves its largest value when most of the occurrences in the GLCM are concentrated near the main diagonal. The inverse difference moment is inversely proportional to the GLCM contrast.

(e) The mean $\frac{1}{2}\sum_i^K \sum_j^K (iC_{ij} + jC_{ij})$ of C describes whether I is dark or bright. Generally speaking, a larger mean indicates that I is brighter while a smaller mean indicates that I is darker.

(f) The variance $\frac{1}{2}\sum_i^K \sum_j^K ((i-\mu)^2 C_{ij} + (j-\mu)^2 C_{ij})$ of C shows us the distribution of the elements in C. When most elements in C are close, the variance is near to zero.

(g) The maximum probability $\text{Max}\{C_{ij}\}$ of C gives the maximum occurrence of gray levels in I. It is expected to be high if the occurrence of the most predominant pixel pairs is high.

Tamura features

Tamura et al.[29] proposed six features, coarseness, contrast, directionality, line-likeness, regularity, and roughness, which are often used to describe the texture of an image (or a region).[4,22] The experimental results show that the three features coarseness, contrast, and directionality correlate closely with human perception. The other three features are highly correlated with the above three mentioned features and do not add much to the effectiveness of the texture description. The three Tamura features coarseness, contrast, and directionality of an $m \times n$ image I are defined as follows.

(a) **Contrast:** In the narrow sense, contrast stands for picture quality. Contrast can be influenced by the following four factors:
   - dynamic range of gray levels,
   - polarization of the distribution of black and white on the gray-level histogram,
   - sharpness of edges,
   - period of repeating patterns.

   The contrast T_{con} of I is $T_{con} = \frac{\sigma}{\alpha_4^z}$, where $\mu_4 = \frac{1}{m \times n}\sum_{x=1}^m \sum_{y=1}^n (I(x,y) - \mu)^4$, $\alpha_4 = \frac{\mu_4}{\sigma^4}$, σ is the standard deviation of the gray level of the pixels in I, and z is a constant experimentally determined to be 0.25.

(b) **Coarseness**: The coarseness gives information about the size of the texture elements. The greater the coarseness is, the rougher the texture. If there are two different textures, one macrotexture of great coarseness and another microtexture of low coarseness, the macrotexture is considered. The essence of calculating the coarseness value is to use operators of various sizes. At each pixel $I(x,y)$ located at the coordinates (x,y) on an image I, the coarseness measure is calculated as follows.

(1) Compute six averages for the windows of size $2^k \times 2^k$, $k = 0, 1, \ldots, 5$, around the pixel:

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \left(\sum_{i=y-2^{k-1}}^{y+2^{k-1}-1} \frac{I(i,j)}{2^{2k}} \right). \quad (1)$$

(2) Take the differences between the pairs of averages corresponding to non-overlapping neighborhoods on opposite sides of the point in horizontal and vertical orientations:

$$E_k^h(x,y) = |A_k(x+2^{k-1}, y) - A_k(x-2^{k-1}, y)|$$

and

$$E_k^v(x,y) = |A_k(x, y+2^{k-1}) - A_k(x, y-2^{k-1})|. \quad (2)$$

(3) Select the most suitable size which gives the highest difference value:

$$s(x,y) = \text{ARG}\left(\underset{k=1}{\overset{5}{\text{MAX}}} \underset{d=h,v}{\text{MIN}} E_k^d(x,y) \right). \quad (3)$$

(4) Finally, take the average over 2^S as the coarseness measure T_{crs} of I:

$$T_{crs} = \frac{1}{m \times n} \sum_{x=1}^{m} \sum_{y=1}^{n} 2^{s(x,y)}. \quad (4)$$

(c) **Directionality**: This feature measures the frequency distribution of oriented local edges against their directional angle gradient. Let $W(x,y)$ with 3×3 pixels be the corresponding window of $I(x,y)$. The Sobel operator can be used to compute the horizontal difference $\Delta G_x(x,y) = G_x \oplus W_S(x,y)$ and vertical difference $\Delta G_y(x,y) = G_y \oplus W_S(x,y)$ of each pixel $I(x,y)$. The gradient $g(x,y)$ and the gradient direction $\theta_g(x,y)$ of $I(x,y)$ can be computed as follows:

$$g(x,y) = (\Delta G_x^2(x,y) + \Delta G_y^2(x,y))^{1/2}$$

and

$$\theta_g(x,y) = \frac{\pi}{2} + \tan^{-1}\frac{\Delta G_y(x,y)}{\Delta G_x(x,y)}. \quad (5)$$

Then, by quantizing θ_g and counting the pixels with the corresponding gradient θ_g greater than a predefined threshold, a histogram of θ_g, denoted as H_{dir}, can be constructed. $H_{dir}(\theta_g)$ is relatively uniform for images without strong orientation but is peaky for highly directional images. Hence, the degree of directionality relates to the sharpness of peaks. The directionality T_{dir} is obtained as follows:

$$T_{dir} = 1 - r \times n_p \sum_{p}^{n_p} \sum_{\theta_g \in w_p} (\theta_g - \theta_p)^2 H_{dir}(\theta_g), \quad (6)$$

where n_p is the number of peaks, θ_p is the position of the pth peak, w_p is the range of the angles attributed to the pth peak, and r is a normalizing factor related to quantizing levels of θ_g.

Adaptable thresholding detector

The general principle of partition is that the data in an identical group should be very similar, but those in distinct groups should vary tremendously. Variance (or standard deviation) is usually used to define the difference among data in a group. Hence, Otsu's method,[25] Ng's method,[24] and the MCVT method[20] all classify data according to the within-class variances, which should be as small as possible. Besides, through the number of data in each group, Otsu's and Ng's methods integrate the variances of data in all the divided groups into one variance to describe the discrepancy of the data within a class. Otsu's method applies the variance of data and the number of data within a class to decide the optimal threshold. Hou et al.[20] found that the threshold obtained by OTM tends to draw closer to the cluster with a larger variance or a larger number of data. Hence, Tsai et al.[30] proposed an adaptable threshold decision method (ATDM) to remedy these drawbacks.

Let $x_{\min}$ and $x_{\max}$ be the minimal and maximal data in a data set, which will be divided into G groups according to the distribution of the data values. In this case, $G-1$ thresholds $t_1, t_2, \ldots, t_{G-1}$ must be specified, so that all the data of the gth group are in the interval between t_{g-1} and t_g. Let $x_{g,i}$ be the ith smallest data value in the gth group, and $n_{g,i}$ be the number of data of which the values are equal to $x_{g,i}$ in the gth group. Given any threshold $T = (t_1, t_2, \ldots, t_{G-1})$, the group interval of the gth group, $R_g(T)$, will be

$$R_g(T) = \begin{cases} t_1 - x_{\min}, & \text{if } g = 1, \\ t_g - t_{g-1}, & \text{if } 1 < g < G, \quad \text{and} \\ x_{\max} - t_{G-1}, & \text{if } g = G. \end{cases} \tag{7}$$

The percentage $P_g(T)$ of the data quantity in the gth group to the entire data set is

$$P_g(T) = \frac{\sum_{i=1}^{R_g(T)} n_{g,i}}{\sum_{g=1}^{G} \sum_{i=1}^{R_g(T)} n_{g,i}}. \tag{8}$$

The average data value M_g of the gth group is

$$M_g = \frac{\sum_{i=1}^{R_g(T)} n_{g,i} x_{g,i}}{\sum_{i=1}^{R_g(T)} n_{g,i}}. \tag{9}$$

The standard deviation $Std_g(T)$ of the data values in the gth group is:

$$Std_g(T) = \sqrt{\frac{\sum_{i=1}^{R_g(T)} \sum_{j=1}^{n_{g,i}} (x_{g,j} - M_g)^2}{\sum_{i=1}^{R_g(T)} n_{g,i}}}. \tag{10}$$

ATDM will select the optimal threshold via R_g, P_g, and Std_g by testing every possible threshold T, where R_g is the difference of the maximal and minimal values in the gth group. When given any threshold $T = (t_1, t_2, \ldots, t_{G-1})$, ATDM computes the optimal thresholds T^* by the following formula:

$$T^* = \mathrm{ARG}\left(\mathop{\mathrm{MIN}}_{T}\left(\sum_{g=1}^{G} \frac{P_g(T) Std_g(T)^{r_1}}{R_g(T)^{r_2}}\right)\right). \tag{11}$$

Here, r_1 and r_2 are two given constants describing the relations among R_g, P_g, and Std_g. Setting different values to r_1 and r_2 produces different thresholds, and it also produces different segmentation results. Thus, it is essential to assign the most suitable values to r_1 and r_2 for selection of the most suitable thresholds.

To partition the same data set, different thresholds could be used according to the requirement of applications. For a special application, the images generally have similar properties (characteristics). Hence, one can take the accumulated historic data of the application to train the parameters r_1 and r_2 to be most appropriate for the threshold decision.

Segmentation errors

Misclassification error (MCE),[28] relative foreground area error (RAE),[28] and relative distance error (RDE)[32] are three commonly used segmentation error measures. In this article, these three measures MCE, RAE, and RDE will be adopted to evaluate the performance of a segmentation method. For a two-class segmentation problem, MCE can be described as

$$\mathrm{MCE} = 1 - \frac{|B_O \cap B_T| + |F_O \cap F_T|}{|B_O| + |F_O|}, \tag{12}$$

where B_O and F_O are the background and foreground pixels assigned by experts, respectively, B_T and F_T are the background and foreground pixels in the segmented image, respectively, and $|A|$ represents the number of pixels in set A. The definition of RAE is

$$\mathrm{RAE} = \begin{cases} \dfrac{A_R - A_T}{A_R}, & \text{if } A_T < A_R, \\[2ex] \dfrac{A_T - A_R}{A_T}, & \text{if } A_T \geq A_R, \end{cases} \tag{13}$$

where A_R is the area of the ground-truth object and A_T is the area of the segmented object.

Let $e_1, e_2, \ldots, e_{n_e}$ be the pixels on the extracted contour E and $t_1, t_2, \ldots, t_{n_t}$ be the pixels on the target contour (probably drawn by an expert) T, where n_e and n_t are the numbers of pixels on E and on T, respectively. To check whether the pixels on E are close to the pixels on T, for each pixel, RE computes the distance d_{e_i}:

$$d_{e_i} = \mathrm{MIN}\{\mathrm{Distance}(e_i, t_j) | j = 1, 2, \ldots, n_t\}, \tag{14}$$

where Distance(e_i, t_j) represents the Euclidean distance between e_i and t_j. To detect some pixels on T without being

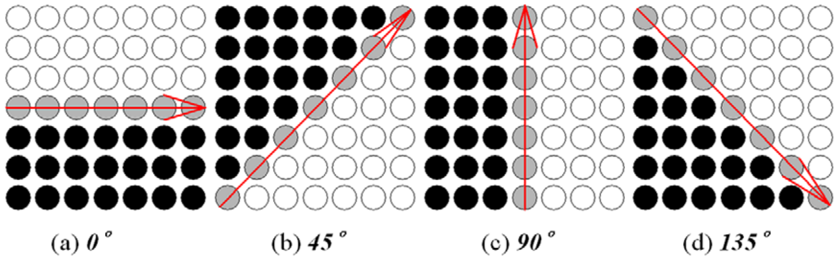

Figure 2. Four θ_l-partitions of W.

mapped to proper pixels on E, RE also calculates the distance d_{t_j}:

$$d_{t_j} = \mathrm{MIN}\{\mathrm{Distance}(e_i, t_j) | i = 1, 2, \ldots, n_e\}. \quad (15)$$

The relative difference error RDE is defined as follows:

$$\mathrm{RDE} = \frac{\sqrt{\frac{\sum_{i=1}^{n_e} d_{e_i}^2}{n_e}} + \sqrt{\frac{\sum_{i=1}^{n_t} d_{t_i}^2}{n_t}}}{2}. \quad (16)$$

FBS METHOD

The FBS method contains three stages: feature extraction, feature-based segmentation, and region merging. The feature extraction stage is to extract image features, suppress noise, and highlight the edges on an image for the following segmentation and analysis. The feature-based segmentation stage is to separate the regions on the image according to the edges. Finally, the region merging stage is to merge adjacent regions with similar texture features into one region.

Feature extraction stage

In this stage, the FBS method first extracts seven GLCM features, three Tamura features, and a gradient feature on each pixel of an image I_o, and then selects three of these features, which can definitely highlight the boundaries of objects, to describe the textures of the objects on I_o. Let $I_0(i, j)$ be the intensity of the pixel located at the coordinates (i, j) on I_0, and we call W_G a corresponding window of $I_0(i, j)$, where W_G consists of $m_G \times m_G$ pixels on I_0 and $I_0(i, j)$ is the central pixel of W_G. W_G can be regarded as an image. The FBS method computes the co-occurrence matrix CM of W_G by the following program segment:

```
For i = 1 to m_G
  For j = 1 to m_G
    if j < m_G then CM(M_G(i, j), M_G(i, j + 1)) + +
    if i < m_G and j < m_G then CM(M_G(i, j), M_G(i + 1, j + 1)) + +
    if i < m_G then CM(M_G(i, j), M_G(i + 1, j)) + +
    if i > 1 and j < m_G then CM(M_G(i, j), M_G(i − 1, j + 1)) + +
```

and then divides each element in CM by $(4 \times m_G \times m_G - 6 \times m_G - 2)$ to constrain the element to be in the interval [0, 1]. From CM, the FBS method can compute the features of the energy, entropy, contrast, inverse difference moment, mean, variance, and maximum probability of the GLCM for $I_0(i, j)$.

Similarly, let W_T, consisting of $m_T \times m_T$ pixels, be another corresponding window of $I_0(i, j)$. W_T is considered to be an image too. Then, the FBS method computes the contrast, coarseness, and directionality of Tamura for $I_0(i, j)$ from W_T.

The gradient magnitude of a pixel can describe the strength of an edge at the pixel. Therefore, the FBS method also computes the gradients of all the pixels in I_0. Since in a texture-based segmentation system the object surface is rugged, the traditional gradient operators, such as Sobel[12,13] and Laplacian gradient operators,[2] are not suitable for computation of the contour gradients of objects with complex texture patterns. This article hence proposes a texture-based gradient operator to compute the gradient feature of a pixel. The gradient feature can not only enrich the object contour but also suppress the noise contour.

The FBS method considers that the direction of an edge at one pixel is close to one of 0°, 45°, 90°, and 135°. Let W_g be a corresponding window of $I_o(i, j)$, where W_g consists of $m_g \times m_g$ pixels. To estimate the direction of the edge at $I_0(i, j)$, W_g is divided into two equal regions according to the four different directions 0°, 45°, 90°, and 135°. Figure 2 shows the four different partitions with $m_g = 7$. The black and white dots signify two different regions, the black region and the white region. We call the partitions in Fig. 2 θ-partitions for $\theta = 0°, 45°, 90°$, and 135°, respectively. For each θ-partition, the average gray levels c_b and c_w of the pixels on black and white regions and $d_\theta = |c_b - c_w|$ are calculated. The gradient feature of $I_o(i, j)$ is defined as $\mathrm{MAX}_{\theta=0°,45°,90°,135°}(d_\theta)$.

The FBS method extracts seven GLCM features, three Tamura features, and a gradient feature for each pixel on I_o. Hence, each pixel possesses 11 feature values. The method transforms the ith feature value of each pixel in I_o into $(\frac{f - f_m}{f_M - f_m}) \times 255$, where f is the ith feature value of the pixel, and f_m and f_M are the minimum and maximum of the ith feature in I_o. Then, the ith new feature values of all the pixels can comprise a gray-level image I_i for $i = 1$ to 11, where I_i and I_o have the same image size. Figure 3 shows the 11 images I_1 to I_{11} of one example image I_o. We call these images the feature images of I_0. The experimental results show that for most images the features inverse difference moment of the GLCM, contrast of Tamura, and gradient can highlight the object boundary, so this method will take these three features to describe the textures of an image. Let I_i, I_c, and

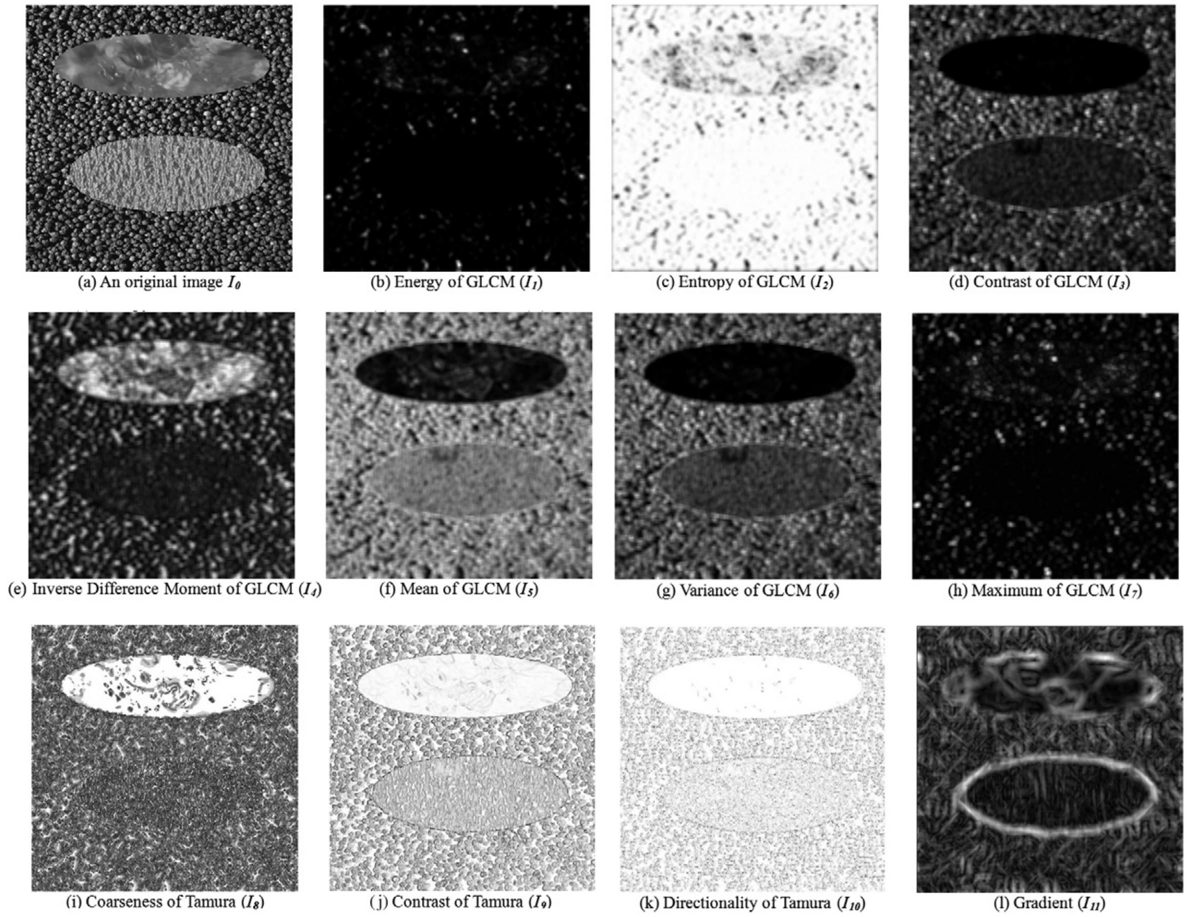

Figure 3. An example after feature extraction.

I_g be three feature images respectively describing the inverse difference moment of the GLCM, the contrast of Tamura, and the gradient of I_0.

Since the gray levels of the pixels in an object on I_i, I_c, and I_g are in disarray, a mean filter[12,13] is used to remove noise on objects. The mean filter is simple, intuitive, and easy to implement to reduce the amount of intensity variation between one pixel and its neighbors, and to reduce noise in an image. The idea of a mean filter is simply to replace the gray level of each pixel $I(x, y)$ in an image I with the mean of the pixel gray levels of the corresponding window of $I(x, y)$. A 3×3 square kernel is used in the mean filter. After being processed by the mean filter, I_i, I_c, and I_g are transformed into another three images I'_i, I'_c, and I'_g.

To highlight the boundaries of objects, the FBS method takes the texture-based gradient operator to compute the gradient features of the pixels in I'_i and I'_c. After that, I'_i and I'_c are changed into I'_{ig} and I'_{cg}. Figure 4 demonstrates the I'_{ig}, I'_{cg}, and I'_g of the images in Fig. 3(e), (j), and (l).

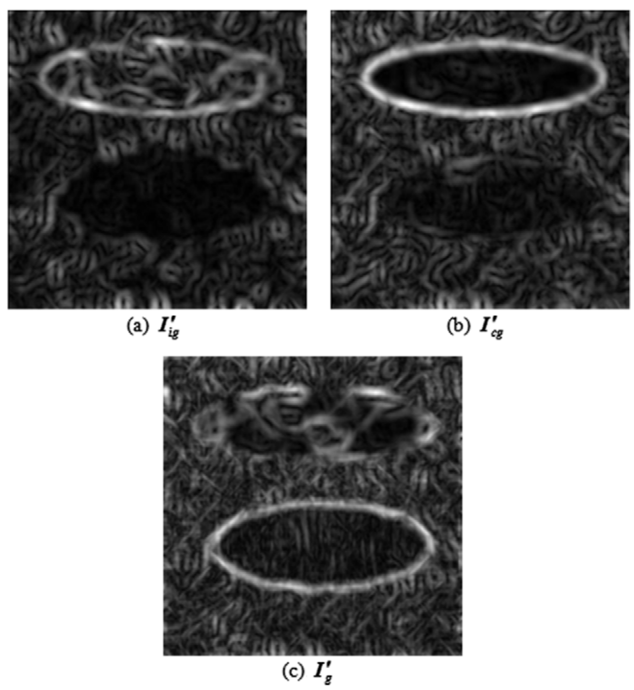

Figure 4. I'_{ig}, I'_{cg}, and I'_g of the images in Fig. 3(e), (j), and (l).

Feature-based segmentation stage

I'_{ig}, I'_{cg}, and I'_g describes the gradient intensities of the pixels on I_0. This stage is to identify the boundaries of the objects on I_0 through I'_{ig}, I'_{cg}, and I'_g. This stage contains five approaches: feature combination, run-length enhancement, adaptable threshold detection, boundary repair, and region detection. The feature combination approach is to integrate

I'_{ig}, I'_{cg}, and I'_g into one image via a geometric mean operation. The run-length enhancement approach is not only to strengthen the boundaries but also to connect the disconnected boundaries of objects. The adaptable threshold detection approach is to isolate the objects from the image background via the ATDM.[30] The boundary repair approach is to mend the broken boundaries of objects. The region detection approach is to thin out the boundaries of objects to the thickness of one pixel.

In the feature extraction stage, three features, inverse difference moment of GLCM, contrast of Tamura, and gradient, are picked out to portray the textures of I_0, which are respectively characterized by I_i, I_c, and I_g. I'_{ig}, I'_{cg}, and I'_g depict the intensities of the pixels located at the boundaries of objects. The feature combination approach integrates them into one image I_G by using the geometric mean:

$$I_G(x, y) = \sqrt[4]{I'_{ig}(x, y) I'_{cg}(x, y) (I'_g(x, y))^2}. \tag{17}$$

Figure 5(a) shows the I_G obtained by combining I'_{ig}, I'_{cg}, and I'_g in Fig. 4.

There may be some noise with high gradient intensity or some disconnected object contours on I_G, such as the gradient indicated by the red arrows in Fig. 5(a). To enhance the object contours and suppress the gradient of noise, run-length enhancement is used. In microscopic vision, one can imagine that an object contour is connected with a lot of tiny straight line segments. In the run-length enhancement, the direction of one line segment is considered to be one of 0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, and 157.5°. Let $W_r(x, y)$ be a corresponding window of $I_G(x, y)$, where $I_G(x, y)$ is the central pixel of $W_r(x, y)$ consisting of $m_r \times m_r$ pixels. Let L_θ be a line segment that cuts across W_r as well as passing through $I_G(x, y)$, and r_θ be the mean of the gray levels of the pixels that are inside W_r and located on L_θ. The run-length $R(x, y)$ of $I_G(x, y)$ is

$$R(x, y) = \underset{\theta=0°,22.5°,45°,67.5°,90°,112.5°,135°,157.5°}{\mathrm{MAX}} (r_\theta). \tag{18}$$

Then, it computes $I_r(x, y) = \frac{R(x,y)-\min_r}{\max_r-\min_r} \times 255$, where $\min_r$ and $\max_r$ are the maximal and minimal values of all $R(x, y)$s. Hence, after running the run-length enhancement, I_G is changed into I_r. Fig. 5(b) is the I_r of I_G in Fig. 5(a).

In order to thin down the object contours, the texture-based gradient operator is used to compute the gradient features of the pixels in I_r. Let $I_{rg}(x, y)$ be the gradient feature of the pixel $I_r(x, y)$. Then, the FBS method subtracts I_{rg} from I_r to generate a new image I_s as follows:

$$I_s(x, y) = \begin{cases} 0, & \text{if } I_r(x, y) < I_{rg}(x, y), \\ I_r(x, y) - I_{rg}(x, y), & \text{otherwise.} \end{cases} \tag{19}$$

Fig. 5(c) and (d) display the I_{rg} and I_s of I_r in Fig. 5(b).

Next, the adaptable threshold detection approach adopts the ATDM[30] to isolate the candidate object contour pixels from I_s. In this approach, two thresholds Th_s and Th_g are given by the ATDM via I_s and I'_g respectively. After this approach, I_s is transformed into a binary image I_{sb}, and I'_g

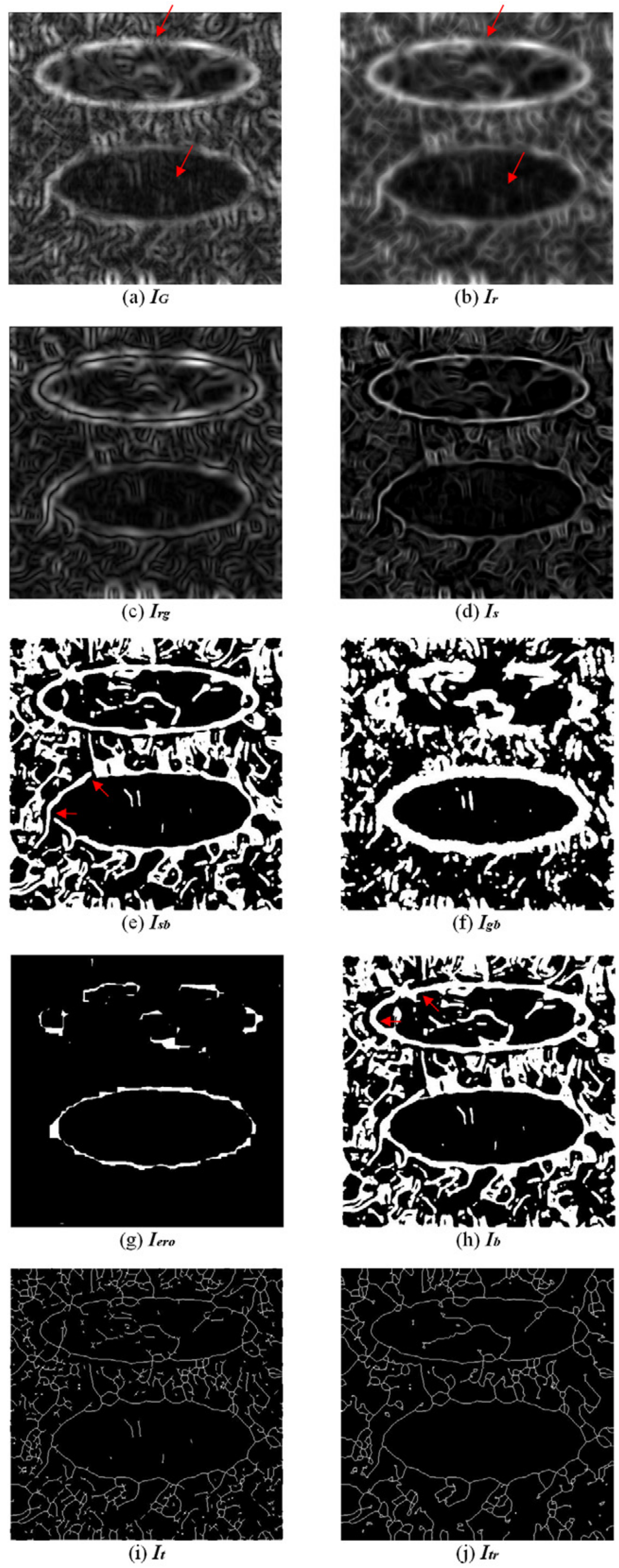

Figure 5. An example through the feature-based segmentation stage.

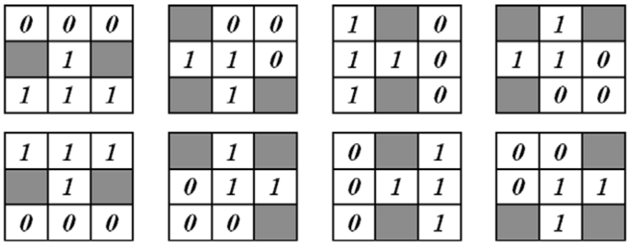

Figure 6. The eight structuring elements for thinning.

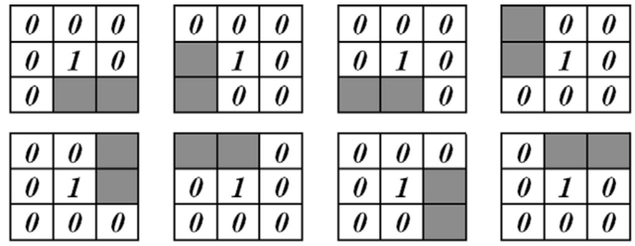

Figure 7. The eight structuring elements for trimming spurs.

into another binary image I_{gb}, by the following formula:

$$I_{sb}(x, y) = 1 \quad \text{if } I_s(x, y) \geq Th_s\text{; otherwise, } I_{sb}(x, y) = 0\text{, and}$$
$$I_{gb}(x, y) = 1 \quad \text{if } I'_g(x, y) \geq Th_g\text{; otherwise, } I_{gb}(x, y) = 0.$$

Fig. 5(e) and (f) demonstrate I_{sb} and I_{gb}, where the pixels with value 1 are white and the pixels with value 0 are black. The white pixels stand for the possible object contour pixels.

Then, the binary morphological erosion operator $\odot$[22] is used to erode I_{gb} and generate a binary image I_{ero} based on a structuring element B:

$$I_{ero} = I_{gb} \odot B = \{I_{gb}(x, y) | B_{xy} \subseteq I_{gb}\}, \qquad (20)$$

where B consists of $m_B \times m_B$ pixels and each pixel in B is 1. Fig. 5(g) is the I_{ero} of I_{gb} in Fig. 5(f) with $m_B = 15$.

The boundary repair approach then combines I_{sb} and I_{ero} into one binary image I_b as follows:

$$I_b(x, y) = I_{sb}(x, y) \vee I_{ero}(x, y), \qquad (21)$$

where $\vee$ is the OR logic operator. Fig. 5(h) is the I_b after combining I_{sb} and I_{gb} in Fig. 5(e) and (g), where the red arrow indicates that I_b provides better object contours than I_{sb} and I_{gb}.

Afterward, the region detection approach takes the HMTS algorithm[12,13] to thin down the edges to the thickness of one pixel. Let each pixel $I_b(x, y)$ in I_b correspond to a 3×3 window $W_t(x, y)$, where $I_b(x, y)$ is the central pixel of $W_t(x, y)$. The HMTS algorithm compares $W_t(x, y)$ with each of the eight structuring elements shown in Figure 6, where the gray pixels stand for the don't-care pixels (a don't-care pixel may be a 1-bit pixel or a 0-bit pixel). We say that $W_t(x, y)$ is matched if $W_t(x, y)$ is completely the same as one of the eight structuring elements, regardless of the don't-care pixels. When $W_t(x, y)$ is matched, the $I_b(x, y)$ is changed into 0. The HMTS algorithm is performed to cut off the redundant-edge pixels, so that the edges have a thickness of only one pixel. The algorithm repeats this procedure until no more thinning is required. Fig. 5(i) displays the result after running the thinning operation on Fig. 5(h).

Since an object contour may be disconnected in I_t, in this approach the FBS method then connects the two closest line end points by a straight line if the distance between the two line end points is less than e pixels. Finally, the spur trimming algorithm[12,13] is employed to remove the spurs. The procedure of the spur trimming algorithm is exactly the same as that of the HMTS algorithm except for the eight structuring elements in Fig. 6, which are replaced by the eight structuring elements in Figure 7. Let I_{tr} be the binary contour image that has been processed by the spur trimming algorithm on I_t. Fig. 5(j) is the I_{tr} of I_t in Fig. 5(i).

Region merging stage. After running the feature-based segmentation stage, I_0 is divided into many regions indicated by I_{tr}. The region merging stage will fuse adjacent regions with similar texture patterns into one. This stage contains three approaches: small region merging, similar region merging, and contour smoothing. In the small region merging approach, a small region will be merged with the region that is most similar to and neighbors the small region. In the similar region merging approach, two adjacent regions with similar textures will be integrated into one region. The contour smoothing approach is to smooth the object contour.

Let R_s and R be the regions in I_0 where R adjoins R_s and the number of pixels in R_s is less than a given threshold Th_A. The difference *Diff* of R_s and R is defined as

$$Diff = \sqrt{(\sigma_{R_s} - \sigma_R)^2 + (\mu_{R_s} - \mu_R)^2}, \qquad (22)$$

where σ_{R_s} and σ_R are the standard deviations of the pixel gray levels in R_s and in R; μ_{R_s} and μ_R are the averages of the pixel gray levels in R_s and in R. The small region merging approach will join R_s to R if *Diff* is smaller than the difference between R_s and each other region adjoining R_s. Let I_{bm} be the binary image after running the small region merging approach on I_{tr}. Figure 8(a) displays the I_{bm} after merging the small regions on I_{tr} in Fig. 5(j) with $Th_A = 300$.

Next, the similar region merging approach is repeated to combine each two adjacent regions if both regions have similar textures until the difference between each two adjacent regions in I_0 is greater than a given threshold Th_d. Let I_{bM} be the binary image obtained by running the similar region merging approach on I_{bm}. Fig. 8(b) demonstrates the I_{bM} of I_{bm} in Fig. 8(a) with $Th_d = 900$.

The object contour on I_{bM} is often rugged. The contour smoothing approach is to smooth the object contours. Let (x_i, y_i) be the coordinates of the ith pixel located on the contour of one object on I_{bM}. The object contour can be smoothed by replacing the coordinates (x_i, y_i) with (x'_i, y'_i),

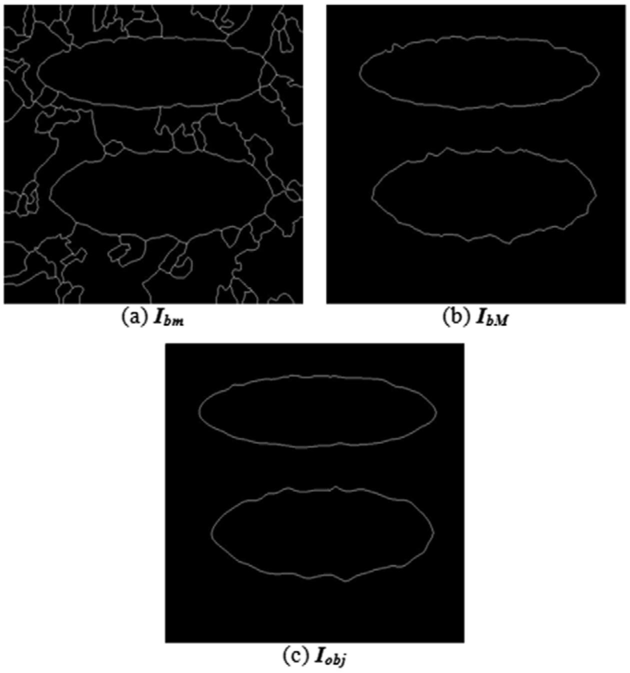

Figure 8. An example through the region merging stage.

where

$$x'_i = \begin{cases} \dfrac{\sum_{j=i-k}^{i+k} x_j}{K}, & \text{if } (i-k) \geq 1 \text{ and } (i+k) \leq n, \\ \dfrac{\sum_{j=i-k}^{n-i} x_j}{K}, & \text{if } (i+k) > n, \\ \dfrac{\sum_{j=1}^{i+k} x_j}{K}, & \text{if } (i-k) < 1, \end{cases} \tag{23}$$

$$y'_i = \begin{cases} \dfrac{\sum_{j=i-k}^{i+k} y_j}{K}, & \text{if } (i-k) \geq 1 \text{ and } (i+k) \leq n, \\ \dfrac{\sum_{j=i-k}^{n-i} y_j}{K}, & \text{if } (i+k) > n, \text{ and} \\ \dfrac{\sum_{j=1}^{i+k} y_j}{K}, & \text{if } (i-k) < 1, \end{cases}$$

$$\text{for } K = \begin{cases} 2k+1, & \text{if } (i-k) \geq 1 \text{ and } (i+k) \leq n, \\ k+1+(n-i), & \text{if } (i+k) > n, \text{ and} \\ k+1+(i-1), & \text{if } (i-k) < 1. \end{cases}$$

Here, n is the number of pixels on this contour. Let I_{obj} be the image generated by smoothing I_{bM}. I_{obj} indicates the obtained object contours on I_0. Fig. 8(c) demonstrates the I_{obj} obtained by smoothing I_{bM} in Fig. 8(b); I_{obj} points out the contours of the objects on I_0 in Fig. 3(a), where $k = 4$.

GENETIC ALGORITHM

Table I shows the parameters that will significantly affect the performance of the FBS method. In this article, a genetic-based parameter selector (GBPS)[26] is employed to determine the most suitable values of m_G, m_T, m_g, m_r, r_1, r_2, m_B, Th_A, Th_d, and k. The GBPS concatenates ten binary substrings s_G, s_T, s_g, s_r, s_1, s_2, s_B, s_A, s_d, and s_k, respectively comprised of n_G, n_T, n_g, n_r, n_1, n_2, n_B, n_A, n_d, and n_k binary bits, to represent a chromosome Ch. m_G, m_T, m_g, m_r, r_1, r_2, m_B, Th_A, Th_d, and k can be encoded as $m_G = 2 \times n'_G + 1$, $m_T = 2 \times n'_T + 1$, $m_g = 2 \times n'_g + 1$, $m_r = 2 \times n'_r + 1$, $r_1 = 0.1 \times n'_1 + 0.1$, $r_2 = 0.1 \times n'_2 + 0.1$, $m_B = 2 \times n'_B + 1$, $Th_A = 20 \times n'_A$, $Th_d = 30 \times n'_d$, and $k = n'_k + 1$, where n'_G, n'_T, n'_g, n'_r, n'_1, n'_2, n'_B, n'_A, n'_d, and n'_k are the numbers of 1-bits in s_G, s_T, s_g, s_r, s_1, s_2, s_B, s_A, s_d, and s_k, respectively.

The GBPS uses the accumulated historic data to decide the most appropriate values of m_G, m_T, m_g, m_r, r_1, r_2, m_B, Th_A, Th_d, and k via a genetic algorithm. When given a Ch, a set of m_G, m_T, m_g, m_r, r_1, r_2, m_B, Th_A, Th_d, and k can be calculated; then the FBS method can be adopted to segment objects based on the m_G, m_T, m_g, m_r, r_1, r_2, m_B, Th_A, Th_d, and k via the accumulated historic data. After that, the segmentation error MCE, RAE, or RDE can be computed by comparing the object contours obtained by the FBS method with the ground truth drawn by certain experts. The GBPS then uses the obtained segmentation error to measure the fitness of Ch.

Initially, the GBPS creates N chromosomes at random, each chromosome comprising of $n_G + n_T + n_g + n_r + n_1 + n_2 + n_B + n_A + n_d + n_k$ binary bits. To develop the best solution, the genetic algorithm repeatedly executes the three operations mutation, crossover, and selection, until the fitnesses of the reserved chromosomes are similar to one another.

In the mutation operation, for each of the N reserved chromosomes, the GBPS uses a random number generator to specify one bit b for each of s_G, s_T, s_g, s_r, s_1, s_2, s_B, s_A, s_d, and s_k. After that, b is replaced by $\neg b$ to generate a new chromosome, where $\neg$ stands for the operation "NOT".

In the crossover operation, similarly, a random number generator is used to designate N' pairs of chromosomes from the N reserved chromosomes. Let $Ch[i..j]$ be the substring consisting of the ith to jth bits in Ch, $Set = \{0, n_G, n_T, n_g, n_r, n_1, n_2, n_B, n_A, n_d, n_k\}$ be an ordered set, and e_i be the ith element in Set. For each chromosome pair (Ch_1, Ch_2), the genetic algorithm concatenates

$$\bigotimes_{i=1}^{10} \left(Ch_1 \left[\left(1 + \sum_{j=0}^{i-1} e_j\right) .. \left(\sum_{j=0}^{i-1} e_j + \left\lfloor \frac{e_i}{2} \right\rfloor\right) \right] \otimes Ch_2 \left[\left(\sum_{j=0}^{i-1} e_j + \left\lfloor \frac{e_i}{2} \right\rfloor + 1\right) .. \sum_{j=0}^{i} e_j \right] \right)$$

into a new chromosome, and concatenates

$$\bigotimes_{i=1}^{10} \left(Ch_2 \left[\left(1 + \sum_{j=0}^{i-1} e_j\right) .. \left(\sum_{j=0}^{i-1} e_j + \left\lfloor \frac{e_i}{2} \right\rfloor\right) \right] \otimes Ch_1 \left[\left(\sum_{j=0}^{i-1} e_j + \left\lfloor \frac{e_i}{2} \right\rfloor + 1\right) .. \sum_{j=0}^{i} e_j \right] \right)$$

Table I. The parameters used in the FBS method.

Parameter	Role	Affected
m_G	Window size used in GLCM	The size of the texture pattern and image resolution
m_T	Window size used in Tamura	The size of the texture pattern and image resolution
m_g	Window size for computation of the gradient feature	The size of the texture pattern and image resolution
m_r	Window size used in run-length enhancement	The size of the texture pattern and image resolution
r_1, r_2	Two given constants for describing the relations among R_g, P_g, and Std_g in the ATDM	The shape of the histogram distribution of the image
m_B	Window size of the structured element used in the morphological erosion operator	The resolution of the image
Th_A	The maximal area of noise	The size of the object
Th_d	The minimal area of an object	The size of the object
K	The range for smoothing the obtained contour	Texture pattern

into another new chromosome, where $\otimes$ represents the concatenation operation.

In the selection operation, according to the fitness, N optimal chromosomes are selected from the N chromosomes reserved in the previous iteration, the N chromosomes created in the mutation operation, and the $2 \times N'$ chromosomes created in the crossover operation. The three operations mutation, crossover and selection need to be continuously operated until the fitnesses of the reserved N chromosomes are close to one another or the number of iterations equals the given maximal number of generations.

EXPERIMENTS

The purpose of this subsection is to investigate the performance of the FBS method by using experiments. In the first experiment, four synthesized images (SI1, SI2, SI3, and SI4 of 512×512 pixels) and two natural scene images (NSI1 of 512×512 pixels and NSI2 of 1024×1024 pixels) downloaded from[1] are used as the test images.

First, images SI1 and NSI1 are randomly selected to train the best parameters $m_G = 9$, $m_T = 11$, $m_g = 7$, $m_r = 13$, r_1, r_2, $m_B = 15$, $Th_A = 300$, $Th_d = 900$, and $k = 4$ via the GBPS, where $N = 10$, $N' = 10$, $|m_G| = 20$, $|m_T| = 20$, $|m_g| = 20$, $|m_r| = 20$, $|r_1| = 40$, $|r_2| = 40$, $|m_B| = 20$, $|Th_A| = 50$, $|Th_d| = 50$, and $|k| = 20$. The GBPS uses the RDE as the measure of fitness of Ch based on the m_G, m_T, m_g, m_r, r_1, r_2, m_B, Th_A, Th_d, and k encoded by Ch. Then, the FBS method separates objects from the six test images based on $m_G = 9$, $m_T = 11$, $m_g = 7$, $m_r = 13$, r_1, r_2, $m_B = 15$, $Th_A = 300$, $Th_d = 900$, and $k = 4$. Figure 9 shows the six test images and the segmentation results obtained by the FBS method and the CSGV-based image segmentation method, and Table II demonstrates the obtained segmentation errors, based on $m_G = 9$, $m_T = 3$, $m_r = 15$, $r_1 = 4.5$, $r_2 = 4.5$, $m_B = 15$, and $k = 4$.

The experimental results illustrate that the FBS method can give more precise and smoother object contours than the CSGV-based image segmentation method. From the images in Fig. 9, one can clearly observe that the FBS method provides lower over-segmentation than the CSGV-based image segmentation method.

Table II. The segmentation errors of the first experiment.

Image	Method	MCE	RAE	RDE
SI1	FBS	0.0125	0.0125	1.6541
	CSGV	0.0211	0.0186	2.9775
SI2	FBS	0.0162	0. 0156	1.6734
	CSGV	0.0224	0.0152	2.0976
SI3	FBS	0.0197	0.0096	1.5381
	CSGV	0.0295	0.0091	2.2813
SI4	FBS	0.0215	0.0150	2.3030
	CSGV	0.0247	0.0129	2.4417
NSI1	FBS	0.0185	0.0075	2.6024
	CSGV	0.3035	0.3060	10.6226
NSI2	FBS	0.0556	0.0523	27.5204
	CSGV	0.2712	0.2685	36.0494

In experiment 2, the FBS method is tested on natural texture mosaics from Prague. Here, 20 benchmark images, downloaded from the Texture Mosaics Database (http://mosaic.utia.cas.cz),[17] are used as test images. First, two of the 20 test images are randomly selected to train the best parameters $m_G = 19$, $m_T = 3$, $m_g = 7$, $m_r = 15$, $r_1 = 2.0$, $r_2 = 0.9$, $m_B = 15$, $Th_A = 280$, $Th_d = 260$, and $k = 4$ via the GBPS, where $N = 10$, $N' = 10$, $|m_G| = 20$, $|m_T| = 20$, $|m_g| = 20$, $|m_r| = 20$, $|r_1| = 40$, $|r_2| = 40$, $|m_B| = 20$, $|Th_A| = 50$, $|Th_d| = 50$, and $|k| = 20$. The GBPS similarly uses the RDE as the measure of fitness of Ch based on the m_G, m_T, m_g, m_r, r_1, r_2, m_B, Th_A, Th_d, and k encoded by Ch. Then, the FBS method separates objects from the 20 test images based on $m_G = 19$, $m_T = 3$, $m_g = 7$, $m_r = 15$, $r_1 = 2.0$, $r_2 = 0.9$, $m_B = 15$, $Th_A = 280$, $Th_d = 260$, and $k = 4$.

In this experiment, five segmentation algorithms, AR3D-GM,[16] GMRF-GM,[15] JSEG,[10] Blobworld,[3] and EDISON,[6] are also used to extract objects from the 20 test images. The region-based performance criteria[19] are used to evaluate the segmentation results as well. The region-based performance criteria mutually compare ground truth (GT) image regions with the corresponding machine segmented regions (MS). They are the *correct, over-segmentation, under-segmentation, missed*, and *noise* criteria. *Correct* represents that over 70% of

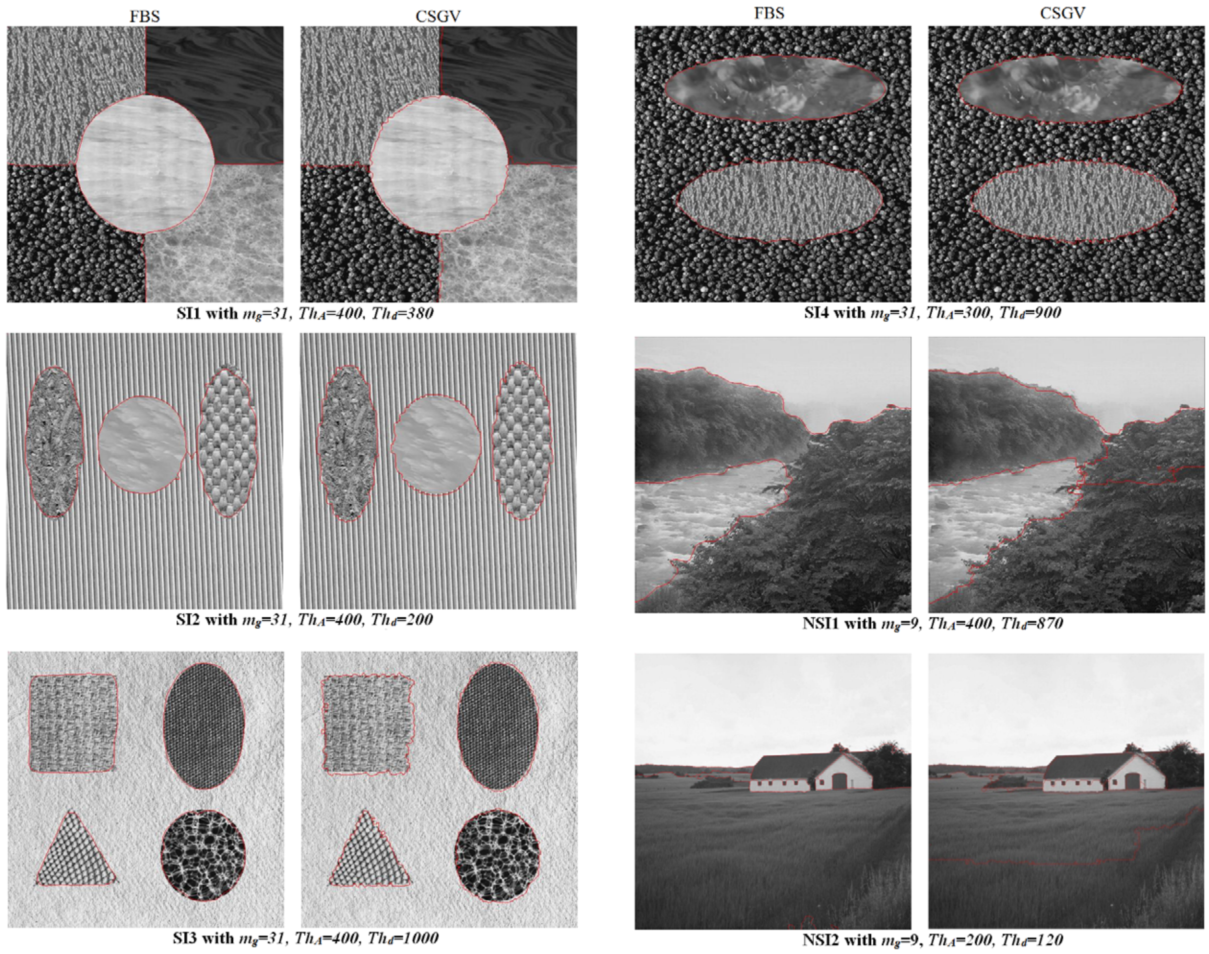

Figure 9. The segmentation results obtained by the FBS method and the CSGV-based image segmentation method.

GT (ground truth) region pixels are correctly assigned, *over-segmentation* means that over 70% of GT pixels are assigned to a union of regions, *under-segmentation* means that over 70% of GT pixels from a classified region belong to a union of GT regions, *missed* means GT in none of the previous categories, and *noise* means MS in none of the previous categories.

Table III demonstrates the segmentation measures CS (correct segmentation), OS (over-segmentation), US (under-segmentation), ME (missed error), and NE (noise error)[17] obtained by the FBS method, AR3D-GM, GMRF-GM, JSEG, Blobworld, and EDISON in extracting objects from the test images. The experimental results show that the FBS method is much better than the other methods in severing the objects from the test images.

Table III. The results of the second experiment.

	FBS	AR3D-GM	GMRF-GM	JSEG	Blobworld	EDISON
CS	62.80	37.42	31.93	27.47	21.01	12.68
OS	21.05	59.53	53.27	38.62	7.33	86.91
US	3.95	8.86	11.24	5.04	9.30	0.00
ME	1.90	12.55	14.97	35.00	59.55	2.48
NE	6.35	13.14	16.91	35.50	61.68	4.68

CONCLUSIONS

This article proposes the FBS method, which can effectively isolate objects with similar texture patterns from a gray-level image. The FBS method takes three features — inverse difference moment of GLCM, contrast of Tamura, and gradient — to describe the textures of an image, and integrates the three features into one by a geometric mean. In addition, a texture-based gradient operation is presented to compute the pixel gradients where an object consists of similar texture patterns; run-length enhancement is offered to strengthen the boundaries of objects and suppress the boundary of noise; GDW enhancement is provided to suppress the gradient of the noise contour but highlight the gradient of the object contour. Moreover, the GBPS is employed to obtain the optimal parameters used in the FBS method to cut off objects with similar texture patterns from various images. The experimental results show that the FBS method is superior to many leading segmentation methods that have been developed to extract objects with similar texture patterns.

ACKNOWLEDGMENTS

This research was supported by NSC grant 101-2221-E-005-016.

REFERENCES

[1] P. Brodatz, *Texture: A Photographic Album for Artists and Designers* (Dover, New York, 1966).

[2] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," IEEE Trans. Commun. **31**, 532–540 (1983).

[3] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," *Proc. 3rd Int'l Conf. on Visual Information Systems* (1999), pp. 509–517.

[4] V. Castelli and L. D. Bergman, *Image Databases: Search and Retrieval of Digital Imagery* (John Wiley & Sons, 2002).

[5] J. M. G. Chamizo, A. F. Guilló, and J. A. López, "Image labelling in real conditions," Int. J. Syst. Cybern. **34**, 1587–1597 (2005).

[6] C. Christoudias, B. Georgescu, and P. Meer, 'Synergism in low level vision,' *Proc. 16th Int'l Conf. on Pattern Recognition*, 4 (2002) pp. 150–155.

[7] R. W. Conners and C. A. Harlow, "Toward a structural textural analyzer based on statistical methods," Comput. Graph. Image Process. **12**, 224–256 (1980).

[8] I. Daubechies, "Orthonormal bases of compactly supported wavelets," Commun. Pure Appl. Math. **41**, 909–996 (1988).

[9] I. Daubechies, "The wavelet transform time-frequency localization and signal analysis," IEEE Trans. Inform. Theory **36**, 961–1005 (1990).

[10] Y. Deng and B. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," IEEE Trans. Pattern Anal. Mach. Intell. **23**, 800–810 (2001).

[11] S. R. Fountain and T. N. Tan, "Efficient rotation invariant texture features for content-based image retrieval," Pattern Recognit. **31**, 1725–1732 (1998).

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing* (Addison-Wesley, 1992).

[13] R. C. Gonzalez and R. E. Woods, *Digital Image Processing* (Prentice-Hall, Englewood Cliffs, NJ, 2002).

[14] M. Gorkani and R. Picard, "Texture orientation for softening photos at a glance," *Proc. IEEE Conf. on Pattern Recognition* (1994), pp. 459–464.

[15] M. Haindl and S. Mikeš, "Model-based texture segmentation," *Proc. Image Analysis and Recognition*, Lect. Notes Comput. Sci. (2004), pp. 306–313.

[16] M. Haindl and S. Mikeš, "Colour texture segmentation using modelling approach," *Proc. 3th International Conference on Advances in Pattern Recognition* (2005), pp. 484–491.

[17] M. Haindl and S. Mikeš, "Texture segmentation benchmark," *Proc. 19th Int'l Conf. on Pattern Recognition* (2008), pp. 1–4.

[18] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Texture features for image classification," IEEE Trans. Syst. Man Cybern. **SMC-3**, 610–621 (1973).

[19] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher, "An experimental comparison of range image segmentation algorithms," IEEE Trans. Pattern Anal. Mach. Intell. **18**, 673–689 (1996).

[20] Z. Hou, Q. Hu, and W. L. Nowinski, "On minimum variance thresholding," Pattern Recognit. Lett. **27**, 1732–1743 (2006).

[21] P. W. Huang, S. K. Dai, and P. L. Lin, "Texture image retrieval and image segmentation using composite sub-band gradient vectors," J. Vis. Commun. Image Represent. **17**, 947–957 (2006).

[22] H. C. Lin, C. Y. Chiu, and S. N. Yang, "Finding textures by textual descriptions, visual examples, and relevance feedbacks," Pattern Recognit. Lett. **24**, 2255–2267 (2003).

[23] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," Int. J. Comput. Vision **43**, 7–27 (2001).

[24] H. F. Ng, "Automatic thresholding for defect detection," Pattern Recognit. Lett. **27**, 1644–1649 (2006).

[25] N. Otsu, "A threshold selection method from gray-level histogram," IEEE Trans. Syst. Man Cybern. **SMC-9**, 62–66 (1979).

[26] P. Y. Pai, C. C. Chang, Y. K. Chan, M. H. Tsai, and S. W. Guo, "An image segmentation-based thresholding method," J. Imaging Sci. Technol., accepted to be appeared.

[27] M. C. Roh, T. Y. Kim, J. Park, and S. W. Lee, "Accurate object contour tracking based on boundary edge selection," Pattern Recognit. **40**, 931–943 (2007).

[28] M. Sezgin and B Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," Electron. Imaging **13**, 146–165 (2004).

[29] H. Tamura, S. Mori, and T. Yamawaki, "Texture features corresponding to visual perception," IEEE Trans. Syst., Man Cybern. **8**, 460–473 (1978).

[30] M. H. Tsai, M. H. Wang, T. Y. Chang, P. Y. Pai, and Y. K. Chan, "An adaptable threshold decision method," *Proc. 5th Intl Conf. on Information Assurance and Security* (2009), pp. 45–48.

[31] L. Wolf, X. Huang, L. Martin, and D. N. Metaxas, "Patch-based texture edges and segmentation," Proc. European Conf. Comput. Vision **2**, 481–493 (2006).

[32] S. F. Yang-Mao, Y. K. Chan, and Y. P. Chu, "Edge enhancement nucleus and cytoplasm contour detector of cervical smear images," IEEE Trans. Syst., Man Cybern. **38**, 353–366 (2008).

[33] D. Ziou, "The influence of edge direction on the estimation of edge contrast and orientation," Pattern Recognit. **34**, 855–863 (2001).