

Journal of Electronic Imaging

JElectronicImaging.org

Segmenting images with complex textures by using hybrid algorithm

Yuxing Han
Chuanbin Lai
Bing Wang
Hui Gu



Yuxing Han, Chuanbin Lai, Bing Wang, Hui Gu, "Segmenting images with complex textures by using hybrid algorithm," *J. Electron. Imaging* **28**(1), 013030 (2019), doi: 10.1117/1.JEI.28.1.013030.

Segmenting images with complex textures by using hybrid algorithm

Yuxing Han,^{a,b,*} Chuanbin Lai,^a Bing Wang,^a and Hui Gu^{c,d}

^aShanghai University, School of Computer Engineering and Science, Shanghai, China

^bShanghai University, Shanghai Institute for Advanced Communication and Data Science, Shanghai, China

^cShanghai University, Material Genome Institute, Shanghai, China

^dShanghai University, School of Materials Science and Engineering, Shanghai, China

Abstract. Application of computer images processing technology to analyze materials microstructural images, particularly metallographic images, has received increasing attention. The metallographic images contain the mesoscopic information on structural relation and components of materials. Quantitative analysis of these images can help to correlate the materials structures to their performance and properties at various levels. There are two challengeable issues necessary to be resolved, i.e., automatic segmentation and classification of different microscopic structures in metallographic images. Since the metallographic images often contain complex textures, the segmentation of them is usually inaccurate with present methods. We propose a hybrid algorithm, which combines the Gaussian filter, the mean shift method, the FloodFill, the improved flow-based difference-of-Gaussians, and the clustering to resolve the issues. The experiment results and the comparative results show that our method is effective to segment and classify the microstructural elements in metallographic images with complex textures. © 2019 SPIE and IS&T [DOI: [10.1117/1.JEI.28.1.013030](https://doi.org/10.1117/1.JEI.28.1.013030)]

Keywords: image segmentation; materials microstructural images; metallographic images; machine learning; segmentation of microstructural elements.

Paper 180748 received Aug. 27, 2018; accepted for publication Jan. 10, 2019; published online Feb. 7, 2019.

1 Introduction

The properties of materials are determined by their microscopic structures, and the structures mainly depend on chemical composition and processing techniques of the materials. Therefore, the study of the quantitative relationship among structures, properties, composition, and processing of the materials is a primary interest for materials science.^{1,2} The materials microstructure images contain mesoscopic structural elements and components of the materials, thus, quantitative analysis of these images is one of the important means to study the characteristic behaviors of the materials structures.

Traditional quantitative analysis of the microstructural images is mainly performed manually and individually. This analysis mainly includes: identifying microstructural elements in a microstructural image; measuring characteristic parameters of the microstructure, e.g., number of grains, size, shape, etc.³ Three metallographic images are shown in Figs. 1(a)–1(c), and Figs. 1(d)–1(f) are human-marked microstructural elements corresponding to the three images. Since the microstructural elements in the images have different sizes and shapes with irregular distribution, measuring these microstructural elements manually is a difficult task with low precision. Therefore, applying computer image processing technology to analyze metallographic images more accurately and elaborately has become a research focus in recent years.^{4–7}

To perform quantitative analysis for metallographic images with computer technology, image segmentation is the primary step. Effective and accurate segmentation results

can help further measure the data of the microstructural elements and components. There have been several methods related to the segmentation of the metallographic images. Chen and Chen⁸ and Liu and Chen⁹ both improved the traditional watershed method to solve the oversegmentation problem in the segmentation of the metallographic images. Albuquerque et al.¹⁰ and Wei-Na and Zhang¹¹ applied the artificial neuronal network in the segmentation and quantification of microstructures in the metallographic images; Chen et al.¹² segmented the metallographic images with a modified fuzzy C-means (FCM) method, and the experiment results showed that their algorithm was robust to noise. However, all the images processed in the above mentioned research have high contrast between different components and simple textures. When dealing with metallographic images, which have complex textures, rich details, and noise, as illustrated in Figs. 1(a)–1(c), accurately segmenting the microstructural elements with the above methods is difficult.

There also have been works on the application of deep learning to process material images. Azimi et al.¹³ proposed a segmentation method based on fully convolutional networks (FCN)¹⁴ for microstructural classification in the images of low carbon steel and achieved good results. Kondo et al.¹⁵ used convolutional neural networks to predict the ionic conductivity in ceramics. However, the deep learning method requires a certain number of images of one kind of material with high resolution and cannot work well for materials with only one or two images. Therefore, in this paper, we focus on the accurate segmentation method for the metallographic images with complex textures, rich details, and scarce samples.

*Address all correspondence to Yuxing Han, E-mail: hanyuxing@gmail.com

1017-9909/2019/\$25.00 © 2019 SPIE and IS&T

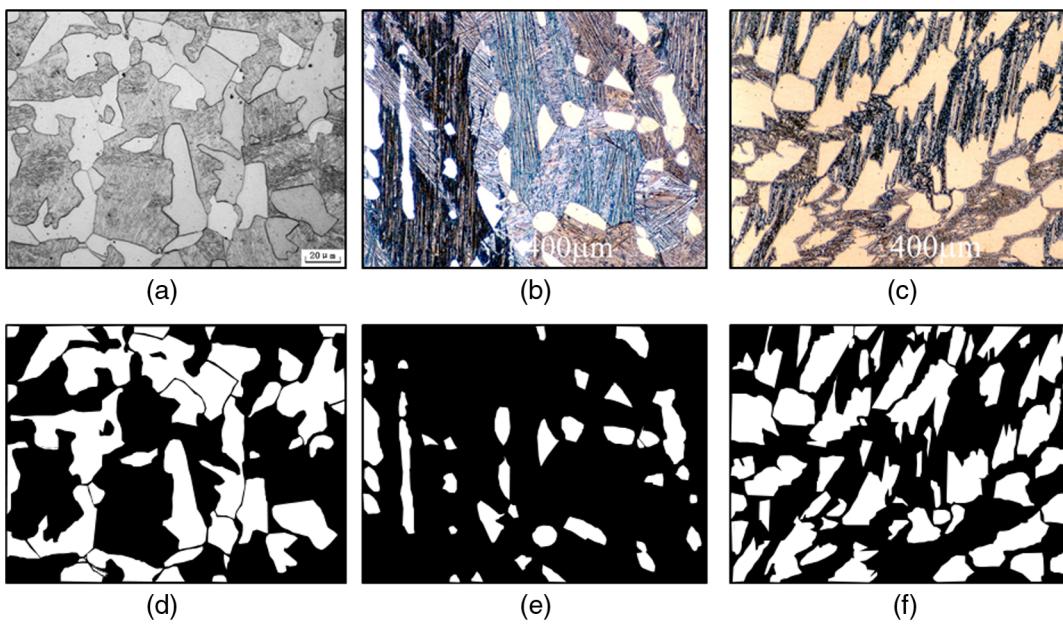


Fig. 1 Examples of the metallographic images. (a)–(c) Three metallographic images and (d)–(f) human-marked microstructural elements. Image (a) is a metallographic image of ferrite–martensite, images (b) and (c) are the metallographic images of Cu–Sn. The three images have complex textures and rich details.

This paper is structured as follows: Sec. 2 introduces the related work of segmentation. Section 3 details our approach for segmenting the metallographic images. In Sec. 4, two related experiments are reported and discussed. We make the conclusion in Sec. 5.

2 Related Work of Image Segmentation

Image segmentation is a classic research subject, and many methods have been developed for the last 40 years.^{16–19} All these methods can be classified into three main categories: region-based methods, edge-based methods, and machine learning-based methods.

1. Region-based methods divide the entire image into subregions directly. This kind of method mainly includes region growing,^{20,21} watershed,²² normalized cuts,²³ split and merge,^{20,21} mean shift,²⁴ graph-based segmentation,^{25–28} and so on. There are two basic ideas of these methods: one is starting from a single point, then gradually merging other points to regions; the other is starting from the whole image, then splitting the whole image into parts. The merging or splitting are both based on some representations of similarity, such as the color similarity between pixels. However, the metallographic images that have complex backgrounds usually can be oversegmented or undersegmented with these methods. The images tend to be oversegmented if there is diversity inside regions and easily undersegmented if there is similarity among regions.
2. Edge-based methods transform images to edge images and take these edges as the dividing lines among regions. Therefore, the segmentation result is influenced greatly by the effect of edge detection. There are many edge detection methods, such as Sobel operator, Prewitt operator, log operator, Canny,³¹ and

active contour model,^{32–34} etc. However, the edges in the metallographic images are often disconnected, and there are many false edges due to the complex textures. These defects of edge detection can heavily reduce the accuracy of segmentation.

3. Machine learning-based methods are divided into traditional machine learning-based methods and deep learning-based methods. Traditional machine learning-based methods are still region-based or edge-based methods essentially. This kind of method turns the segmentation problem into a classification problem and using features designed artificially. For example, a neural network and support vector machine were used to train a classifier to determine whether pixels belong to the edges.^{35,36} However, these methods are unable to completely eliminate the shortcomings in the region-based or edge-based methods. Deep learning-based segmentation methods have made great progress in recent years, such as FCN,¹⁴ SegNet,³⁷ DeepLab,^{38,39} etc. Deep learning-based methods can achieve the state-of-the-art results, however, the training of deep learning models requires a large amount of samples. Since the cost of metallographic images acquisition is high, the images of the same kind of material are scarce, and the number of labeled images is also small. Lack of data limits the effectiveness of deep learning-based methods in metallographic images segmentation.

In general, so far, there has been no a single, effective, and accurate segmentation method to process the metallographic images with disconnected edges, irregular microstructural elements, complex backgrounds, and scarce samples. According to the characteristics of the metallographic images, we propose a hybrid segmentation method, which combines mean shift, edge detection, and clustering.

The experimental results show that our approach can solve the oversegmentation problem properly and extract the microstructural elements from the metallographic images accurately.

3 Methods

The proposed approach consists of two major parts: one is the preliminary segmentation and the other one is the classification and further merging of the preliminary segmentation results.

3.1 Preliminary Segmentation of the Images

According to Sec. 2, the metallographic images are easily to be oversegmented or undersegmented with a single segmentation method. To obtain accurate segmentation results, a method combined with some submethods is proposed here.

3.1.1 Image preprocessing

One of the major challenges to segment the metallographic images is the obvious textures. The local region with the same type of textures should be segmented to one region. However, the obvious textures are easily mistaken for the edges, then the local regions tend to be oversegmented. Thus, blurring the textures is the primary task.

In this paper, the Gaussian filter is applied to blur the images. On the one hand, the Gaussian filter can blur the difference between pixels, which are close to each other in a spatial distance. On the other hand, the Gaussian filter can achieve a good denoising effect. The relation of the filtered value $I(x_0, y_0)$ and its neighbor pixels $I(x_i, y_j)$ is shown below.⁴⁰

$$I(x_0, y_0) = \sum_{i,j} I(x_i, y_j) G(x_i, y_j), \quad (1)$$

$$G(x_i, y_j) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x_i-x_0)^2+(y_j-y_0)^2}{2\sigma^2}}, \quad (2)$$

where $G(x_i, y_j)$ is the Gaussian kernel and σ determines the size of the Gaussian filter. The larger the size is, the smoother the processed image is. According to the 3σ principle (pauta criterion),⁴¹ the size dim of the filter can be computed by $dim = 1 + 2 \times \lceil 3\sigma \rceil$.

The result of the Gaussian filter is shown in Fig. 2(b), and the parameter dim is set to 7. Though the Gaussian filter can blur the textures, the true edges can also be blurred. To obtain accurate areas of different microstructural elements, an edge detection method is applied, and the details are discussed in Sec. 3.1.4.

3.1.2 Pixels classification

Due to the complex textures in the images, extracting an intact region is difficult. To obtain the regions, the pixels are classified in the proposed method. Once all pixels are classified into clusters, regions can be obtained according to the classification results.

The mean shift method is a robust approach toward feature space analysis,²⁴ which can be used to classify the pixels without specifying the number of clusters. Since the mean shift method works in the feature space, pixels need to be mapped into the feature space first. In our method, two kinds of features are extracted for each pixel p_i : one is the pixel co-ordinate $p_i^s = (x, y)$ in an image and the other one is the pixel value p_i^c .

For each pixel p_i , the mean shift method iteratively searches the corresponding prototype point $m_{i,k}$ of the pixel as in the following equation:²⁴

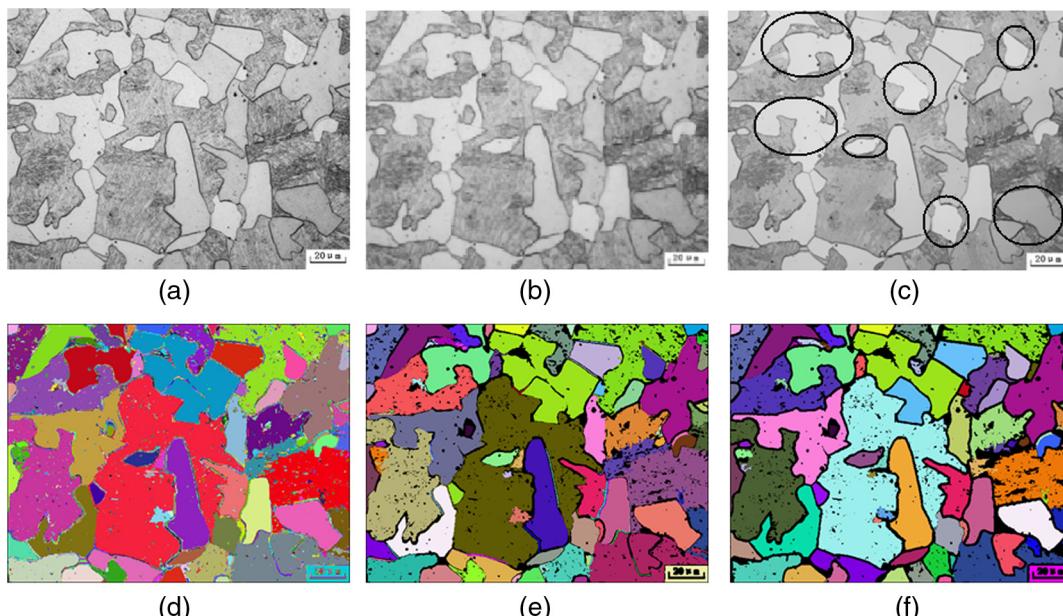


Fig. 2 Overview of the processing results in the first part of our approach. (a) The original metallographic image; (b) result with the Gaussian filter; (c) result with the mean shift method, the regions marked by black circles are the microstructural elements need to be extracted; (d) result with the FloodFill; (e) result by removing tiny regions; and (f) result of adding the edge detection.

$$m_{i,k+1}^s = \frac{\sum_{n=1}^N P_n^s g\left(\left\|\frac{p_n^c - m_{i,k}^c}{h_r}\right\|^2\right)}{\sum_{n=1}^N g\left(\left\|\frac{p_n^c - m_{i,k}^c}{h_r}\right\|^2\right)}, \quad (3)$$

where $m_{i,k}^c$ is the color value of $m_{i,k}$, which is the k 'th prototype point; $m_{i,k+1}^s$ is the co-ordinate of $m_{i,k+1}$, which is the $(k+1)$ 'th prototype point; p_n are pixels in a $h_s \times h_s$ square neighborhood centered on $m_{i,k}$; p_n^s and p_n^c are pixel p_n 's coordinate and color value, respectively; the operator $\| * \|$ is norm operator; h_r is a threshold; g is a kind of kernel function and plays a role as weight. In our approach, the uniform kernel is chosen for g given by

$$g(x) = \begin{cases} 1 & 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

The $(k+1)$ 'th prototype point $m_{i,k+1}$ is considered as the eventual prototype point, if $m_{i,k+1}$ meets one of the following conditions: (1) $m_{i,k+1}^s$ approximates to $m_{i,k}^s$ and (2) $m_{i,k+1}^c$ approximates to $m_{i,k}^c$. Otherwise, $m_{i,k+1}$ is considered as an intermediate result and used to calculate $m_{i,k+2}$ until k reaches a threshold. Large threshold can lead to the long running time. Through the experiments, when k is bigger than 5, the classification result has no obvious improvement. Hence, the threshold of k is set to 5 in our method. Then, each pixel p_i can find its final prototype point $m_{i,c}$, and p_i is classified into the cluster, which belongs to $m_{i,c}$. Specifically, the color value of p_i is changed to the color value of $m_{i,c}$.

There are two parameters, h_s and h_r , that can influence the segmentation results greatly. For example, in Fig. 2(c), the classification result is shown when the three parameters are set to $dim = 3$, $h_s = 15$, and $h_r = 20$. The parameters setting is discussed in Sec. 3.3.

3.1.3 Regions extraction

With the mean shift method, all the pixels are classified into different clusters. However, the regions in the images are still not extracted. In order to extract the regions, the FloodFill method⁴² is applied to merge similar pixels into regions.

First, each pixel is considered as a seed point unless it has been merged into a region. Then, from a seed point, its adjacent pixels are merged into a region of the seed point if these pixels' values are similar. The iteration keeps going until all the adjacent pixels have been merged or there are no similar

pixels any more. To better illustrating the result of FloodFill, the extracted regions are painted with random colors, as shown in Fig. 2(d).

Though most pixels can be merged into regions, there are still many tiny regions, which are regarded as noises need be removed, as shown in Fig. 2(d). Specifically, if the area of a region is less than a threshold $minArea$, then the region is removed. The result of removing the tiny regions is shown in Fig. 2(e).

3.1.4 Improved edge detection method

If the edge curves in the original image are too thick, they can be mistaken for the regions. Furthermore, though some neighbor regions are separated with edge curves, they can be wrongly merged. The reason is that the edge curves are blurred by the Gaussian filter. Two examples corresponding to the two situations are shown in Fig. 3.

To overcome the two kinds of bad results, an optional processing step that detects edge curves is adopted. There have been many edge detection methods, such as Canny³¹ and phase congruency.⁴³ However, the results of these methods include too many details that hinder extractions of regions, as shown in Figs. 4(a) and 4(b). Although some researchers can obtain good edge curves in an image based on machine learning, e.g., the method proposed by Martin et al.,³⁶ the edge curves are too thin and some edge curves are disconnected and missing, as illustrated in Fig. 4(c). To obtain accurate and significant edge curves of the regions, we proposed an improved method based on flow-based difference-of-Gaussians (FDoG)^{44,45} here.

FDog is a flow-driven anisotropic filtering framework, which can extract a set of coherent, smooth, and stylistic edge curves from an image. This method including two steps. First, the gradient $g(p_i)$ of each pixel p_i is computed, and the unit vector $t(p_i)$, which is perpendicular to $g(p_i)$, is also calculated. Then, the contrast along the direction of $t(p_i)$ is accumulated. If the accumulated contrast of pixel p_i is large, the pixel p_i is very likely in the edge. The accumulated contrast $H(p_i)$ of pixel p_i is computed by the following equation:⁴⁴

$$H(p_i) = \sum_s F(p_s) G_{\sigma_m}(p_s), \quad (5)$$

where p_s are pixels in the positive and negative direction of $t(p_i)$; $G_{\sigma_m}(p_s)$ is the Gaussian filter with σ_m ; subscript s is in

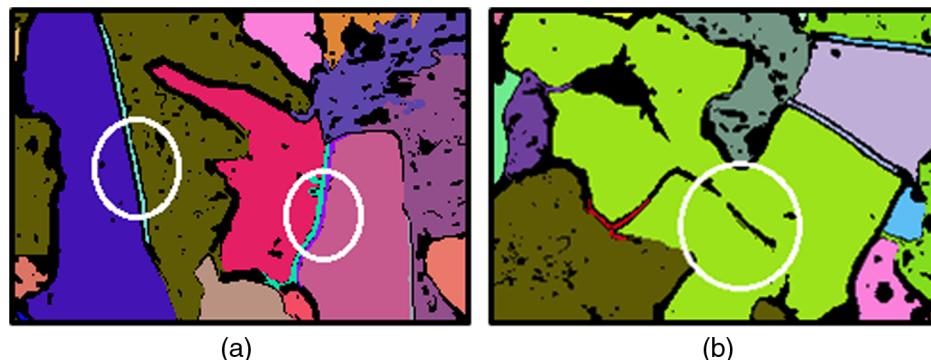


Fig. 3 Two main problems in merging pixels if there are significant edge curves. (a) The edge curves are mistaken for the regions and (b) the closed but disconnected regions are merged.

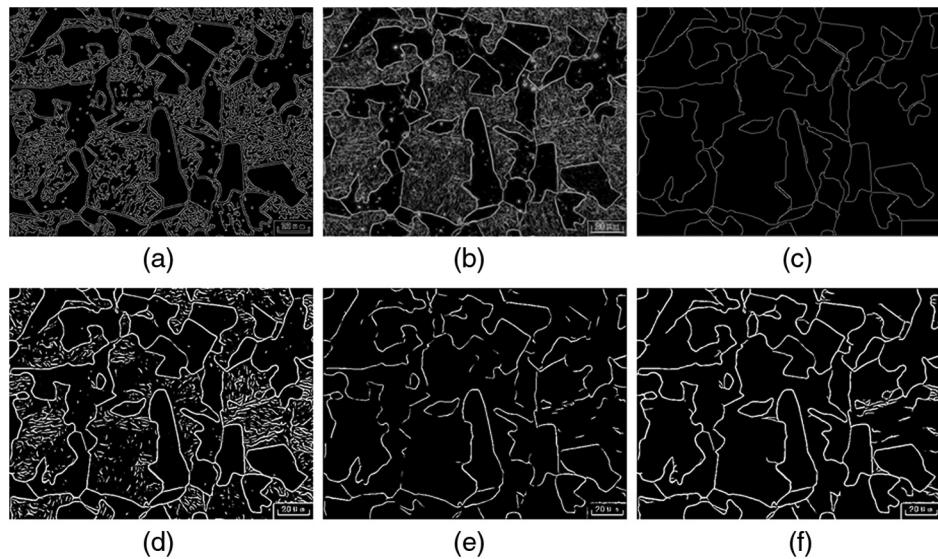


Fig. 4 Results of some edge detection methods. (a) The result of Canny method, (b) the result of phase congruency, (c) the result of the edge detection method based on machine learning, (d) the result of FDoG and $\rho_1 = 1.0$, (e) the result of FDoG and $\rho_2 = 0.95$, and (f) the result of the proposed method.

$[-\lfloor 3\sigma_m \rfloor, \lfloor 3\sigma_m \rfloor]$, as mentioned in Sec. 3.1.1, and negative s means negative direction of $t(p_i)$; $F(p_s)$ is the function to compute the contrast between the positive direction and negative direction of the gradient $g(p_s)$:⁴⁴

$$F(p_s) = \sum_o I(p_o) f(p_o), \quad (6)$$

where p_o are pixels in the positive and negative direction of $g(p_s)$; $I(p_o)$ is the grayscale value of pixel p_o ; o is the number of p_o ; $f(p_o)$ is the DoG(difference-of-Gaussians) filter:

$$f(p_o) = G_{\sigma_c}(p_o) - \rho \cdot G_{\sigma_s}(p_o), \quad (7)$$

where $G_{\sigma_c}(p_o)$ and $G_{\sigma_s}(p_o)$ are Gaussian filter with different σ . According to the research conducted by Kang et al.,⁴⁴ we set σ_s to $1.6\sigma_c$, which can make the shape of $f(p_o)$ closely resemble the shape of Laplacian-of-Gaussian. The range of o is also in $[-\lfloor 3\sigma_c \rfloor, \lfloor 3\sigma_c \rfloor]$. Parameter ρ controls the quantity of details detected, and the range of ρ is $[0.9, 1.0]$.⁴⁴

The result of the FDoG can be obtained by binary thresholding the $H(p_i)$ for all pixels. In the results of the FDoG, the larger the parameter ρ the more details in the results, and vice versa. Choosing a proper value for the parameter ρ is difficult. To overcome this problem, the double-threshold strategy is adopted to improve FDoG method.

For parameter ρ , a large value ρ_1 and a small value ρ_2 are selected. Through experiments, the method can achieve good result if ρ_1 and ρ_2 are set to 1 and 0.95, respectively. For other parameters, the default values are used, i.e., $\sigma_m = 3.0$ and $\sigma_c = 1.0$.⁴⁴ Then, two detection results under different parameters ρ can be obtained. Figure 4(d) is result with $\rho_1 = 1$ and Fig. 4(e) is result with $\rho_2 = 0.95$. Obviously, the false edge curves in Fig. 4(e) are much less than the false edge curves in Fig. 4(d). However, the edge curves in Fig. 4(e) are most disconnected. Therefore, the edge curves in Fig. 4(e) need to be connected to get the clear and accurate edge curves. For each pixel of the edge curves in the result with ρ_2 , if there are other pixels in the neighbor of the

corresponding pixel in the result with ρ_1 , then the other pixels are added to the result with ρ_2 . The result of the connection is shown in Fig. 4(f).

The above two problems in the FloodFill can be solved through making use of the edge curves. In the process of the FloodFill, the pixels that belong to the edge curves are excluded, i.e., let these pixels labeled as merged and cannot be merged into regions. After adding the improved edge detection method to the FloodFill, the preliminary segmentation result is obtained, as presented in Fig. 2(f).

3.2 Classification and Further Merging of the Preliminary Segmentation Results

3.2.1 Feature extraction for regions

Although some regions are far each other in the spatial distance, they still belong to the same class, such as the regions marked by black circle in Fig. 2(c). To complete the following quantitative analysis of the images, these regions need to be classified to find out the relationship between them. The general approach for classification involves feature extraction and feature selection (dimensionality reduction) to obtain feature vectors; then machine learning is used to construct a classification model with these vectors. Until now, a large number of feature extraction algorithms have been developed.⁴⁶⁻⁵⁵ Since many of the features are relevant, using all of these features is unnecessary and redundant. Taking into account the speed of the program and the effect of classification, one color feature F^m and one texture feature F^g are extracted from each region. The two features are detailed below:

Mean value F^m is the average grayscale value of all pixels in a region. F^m is computed as follows:

$$F^m = \frac{\sum_{i=1}^n p_i^c}{n}, \quad (8)$$

where p_i^c is the grayscale value of the pixel p_i and n is the number of the pixels in a region. F^m can describe the general

color information of a region. If textures of the image are simple and the contrast between regions is large, F^m is enough to distinguish the regions. However, most metallographic images have complex textures and rich details. The regions with different textures can have the similar F^m . Thus, the texture features need be applied to discriminate different regions.

Histogram of oriented gradients (HOG) F^g describes an image with a set of histograms of gradient orientations.⁴⁶ Here, for each region, a histogram is used to describe the region. Each bin in a histogram records the number of times a gradient orientation occurs in a region. The gradient is a vector which points the direction of the greatest change rate of the pixel's value. The gradient has two components: the gradient magnitude $M(x, y)$ and gradient orientation $O(x, y)$:

$$M(x, y) = \sqrt{\Delta x^2 + \Delta y^2}, \quad (9)$$

$$O(x, y) = \tan^{-1} \frac{\Delta y}{\Delta x}, \quad (10)$$

$$\begin{cases} \Delta x = I(x+1, y) - I(x-1, y) \\ \Delta y = I(x, y+1) - I(x, y-1) \end{cases} \quad (11)$$

where $I(x, y)$ is the grayscale value of the pixel (x, y) in the original image. The histogram is computed for each region and the number of bins is set to 9, which is proved to be the optimal parameter,⁴⁶ i.e., the 2π is divided into 9 intervals. In the statistics of gradient orientation, the gradient magnitude is used as weight. Furthermore, each bin is divided by the number of pixels in a region to eliminate the impact of

different region sizes. Figure 5 shows the HOG of two different regions.

From the above, the dimension of F^g is 9. To combine the two features F^g and F^m with equal weight, the dimension of each feature is supposed to be the same, i.e., one dimension. In our approach, the isometric feature mapping (Isomap)⁵⁶ is chosen to reduce the dimensionality of F^g . Isomap is a non-linear dimensionality reduction method and often used for computing a low-dimensional embedding of a set of high-dimensional data. Our experimental results also show that the Isomap can accurately reflect the essence of the high-dimensional data.

Through feature extraction and feature selection, a two-dimensional (2-D) feature vector F_i can be obtained for i 'th region, composed as the two essential cues:

$$F_i = [F_i^m, F_i^g]. \quad (12)$$

Furthermore, since the scales of the two features are different, all the features need to be normalized before they are used to classify the regions.

3.2.2 Classification and merging for the regions

As mentioned in Sec. 3.2.1, each region can be represented by a 2-D vector. Then, the following step is classifying the vectors into different clusters. Since these vectors are unlabeled, the unsupervised methods are adopted to construct the classifier. For two different situations, two kinds of clustering algorithms are employed, respectively: the k-means⁵⁷ for the case that the number of clusters is already known; and the density-based spatial clustering of applications with noise (DBSCAN)⁵⁸ for the opposite case.

K-means aims to classify m data objects into k clusters. The data objects belong to the same cluster if they belong to

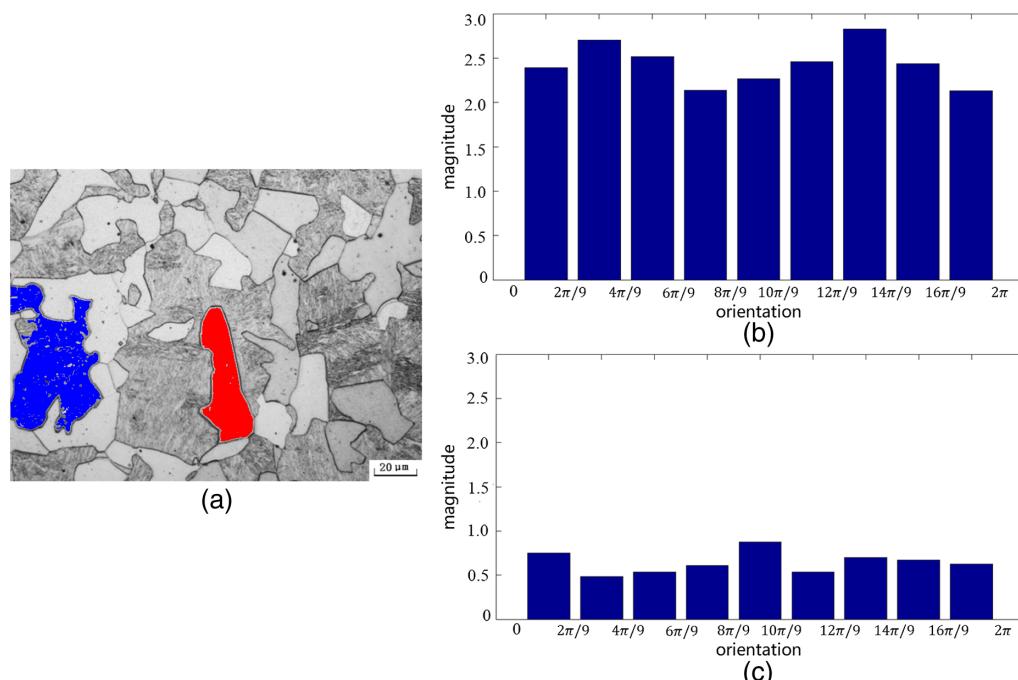


Fig. 5 HOG of two different regions. (a) The original image with two marked regions. The region painted by blue is marked as region 1, and the region painted by red is marked as region 2. (b) The HOG of region 1. (c) The HOG of region 2.

the same nearest center point. If the value of k is known in advance, the k-means can be applied for its interpretability, fast convergent rate, and nice clustering effect. However, in many cases, the value of k is unknown. Thus, the DBSCAN is used to classify the data without pointing the value of k .

DBSCAN is a density-based clustering algorithm: given a set of data, high-density regions can be discriminated from low-density regions with DBSCAN. In addition, DBSCAN can be used for arbitrarily shaped clusters and robust to noises. There are two parameters in the DBSCAN method: the neighborhood size Eps of a core point and the minimum number $minPts$ of points required in the neighborhood of a core point. The two parameters can affect the result of clustering, as shown in Fig. 6. The clustering result shown in Fig. 6(a) is better than other clustering results. If the Eps is too small or the $minPts$ is too large, the number of core points is small. Under this condition, the data are easy to be overclassified or mistaken for the noises, such as the results shown in Figs. 6(b) and 6(e). If the Eps is too large or the $minPts$ is too small, the number of core points is large. In this case, the data are easy to be underclassified or some noises cannot be detected, such as the results shown in Figs. 6(c) and 6(d). In the classification of the regions, there are some recommendation in Sec. 3.3 for the selection of the two parameters Eps and $minPts$.

With k-means or DBSCAN, all the regions can be labeled. Based on the labeled regions, the microstructural elements that belong to the same class can be merged if their shapes are connected and there is no edges between them. The merging results are presented in the columns (6) and (7) of Fig. 8.

3.3 Parameters Selection

In the proposed approach, the four parameters $[dim, h_r, h_c, minArea]$, which are introduced in Sec. 3.1, are keys for the preliminary segmentation. In general, for the selection of the parameters $[dim, h_r, h_c]$, it tends to choose large values on images that have rough textures and many small grains, such as Figs. 1(b) and 1(c), and choose small values on images that have dense and thin textures, such as Fig. 1(a). And for $minArea$, it tends to use small values on images that have many small grains and use large values on images, where most grains are large. In addition, for the range of the parameters, dim is commonly in $\{3, 5, 7, 9\}$, h_r and h_c are both in $[10, 50]$, and $minArea$ is less than the size of the image. Furthermore, the following

experiments show that if the characteristics in images are similar, such as Figs. 1(b) and 1(c), $[dim, h_r, h_c, minArea]$ can be set the same values. Thus, the same parameters setting can be applied for a certain type of images.

For the second part of our approach, if DBSCAN is chosen in the clustering step, the parameters Eps and $minPts$ need to be considered. Generally, the small Eps or large $minPts$ are used for the metallographic images with many small microstructural elements, as shown in Figs. 1(b) and 1(c); and large Eps or small $minPts$ are applied for the opposite case, as illustrated in Fig. 1(a). Since the feature vectors have been normalized, as mentioned in Sec. 3.2.1, the range of Eps is $(0, 1]$. And the range of $minPts$ is $[1, 30]$ in our approach.

4 Experimental Results and Discussions

The experimental results of our method are described in two subsections. First, the Sec. 4.1 compares the results of using various methods to segment the metallographic images. Then, the validation of our method in the quantitative analysis of the metallographic images is illustrated in Sec. 4.2. In addition, part of the metallographic images is obtained from the publicly available Dissemination of Information Technology for the Promotion of Materials Science (DoITPoMS) library.⁵⁹

4.1 Comparison between Various Segmentation Methods

To demonstrate the effectiveness of our approach, the proposed approach is tested on 10 metallographic images. Due to insufficient data, the deep learning-based segmentation methods are not adopted as comparison method. We compared our method with other region-based and machine learning-based methods, which also utilize texture features. The comparing methods are introduced in brief as follows:

- Hierarchical image segmentation (method 1):⁶⁰ This method extracts texture features around each pixel and uses machine learning method to get the probability that each pixel belong to the boundary. Then, the image is segmented based on oriented watershed transform and hierarchical region tree.
- Graph-based image segmentation (method 2):²⁷ An image segmentation method based on graph cut by using a graph-based representation of the images.

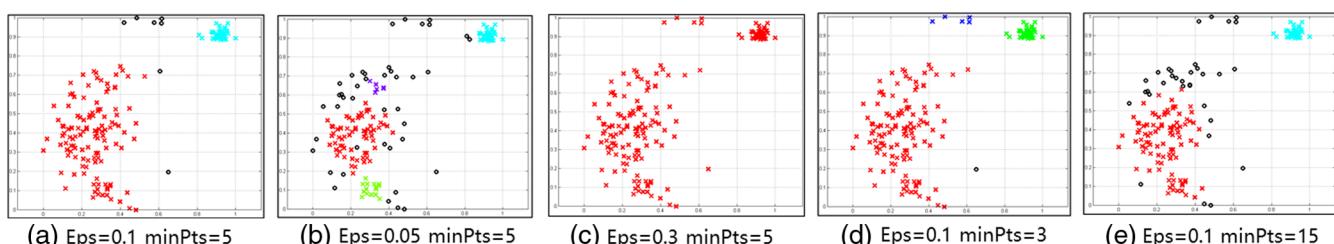


Fig. 6 (a)–(d) Clustering results of DBSCAN with different parameters. The data samples are the regions segmented from a metallographic image [the image (c) in Fig. 1] using our method. Each data point represents one region and is composed of two features extracted from the region, i.e., $[F_i^m, F_i^g]$ mentioned in Sec. 3.2.1. Since the features have been normalization, the range of the axes is $[0, 1]$. In these figures, the black circles represent noise, and the x-type points which have the same color belong to the same cluster.

- Marker-controlled watershed (method 3).²² The watershed transform is often applied to separate objects in an image, and there are many different algorithms to compute watersheds.^{61–64} In this experiment, the Marker-controlled watershed⁶⁴ is adopted.
- Fast generalized FCM (method 4).⁶⁵ A fast and robust FCM framework incorporated local spatial and gray information for image segmentation.

The 10 metallographic images are shown in column (1) of Fig. 7, of which are ferrite–martensite, pearlite–ferrite, troostite–martensite, four Cu–Sn, Cu–Zn, and pearlite–ferrite, respectively. And metallographic images (d) to (j) are obtained from the DoITPoMS library. As discussed in Sec. 3, our segmentation method mainly includes two parts, which are preliminary segmentation of the images and classification and further merging of the regions. First, the

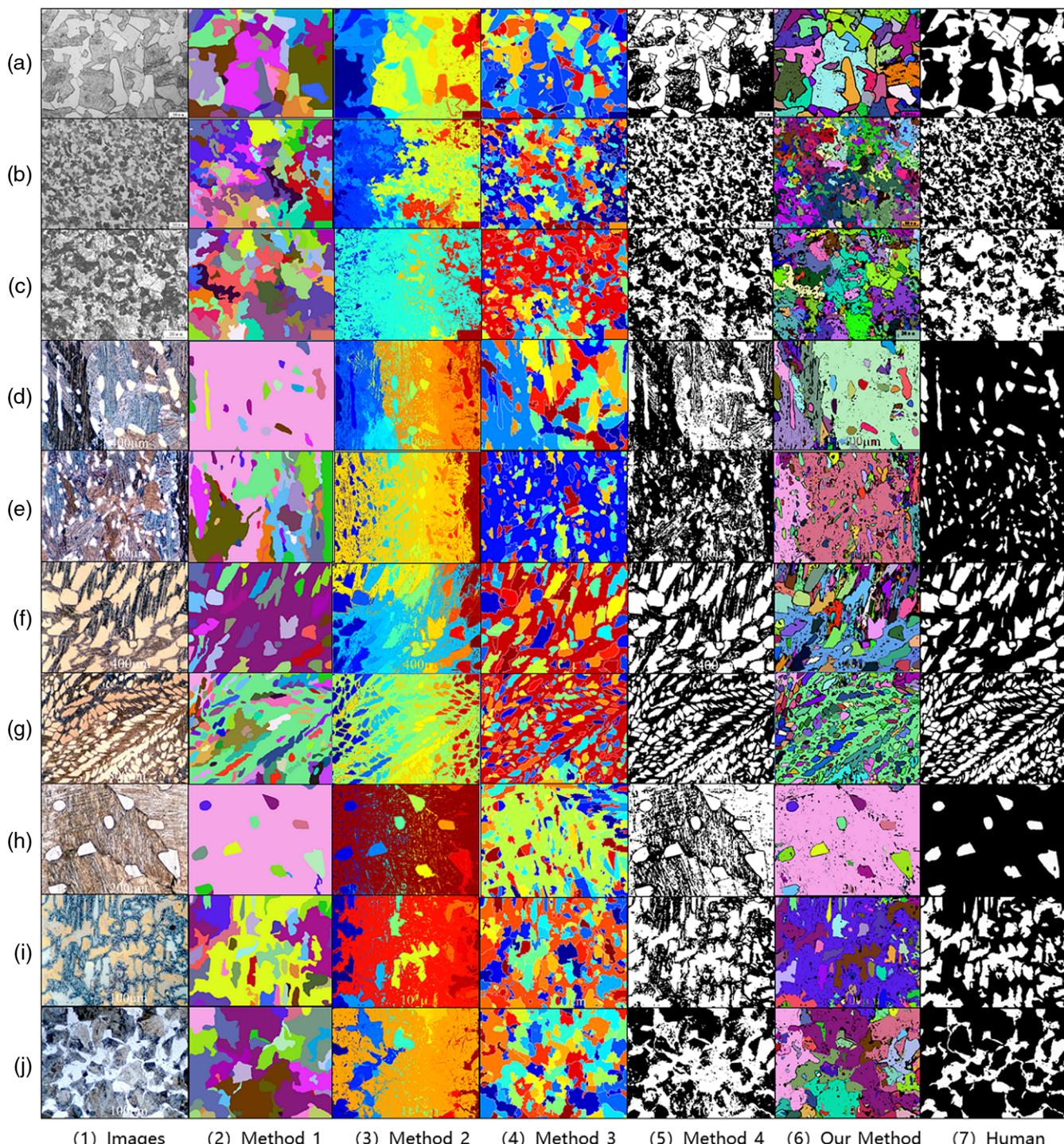


Fig. 7 Comparison of different segmentation methods. Images in column (1) are original images, images (d)–(j) are from the DoITPoMS library. Columns (2)–(5) show the segmentation results of method 1 to method 4, which are described in Sec. 4.1. And images in column (6) are the preliminary segmentation results of our method. Column (7) is the ground truth labeled by human.

preliminary segmentation of our method is compared with other four methods mentioned above, as shown in Fig. 7. The parameter settings of $[dim, h_s, h_c, minArea]$ for the 10 images are $[3, 15, 20, 100]$, $[5, 30, 30, 100]$, $[5, 30, 30, 100]$, $[7, 40, 30, 100]$, $[7, 40, 30, 50]$, $[7, 40, 30, 100]$, $[7, 40, 30, 50]$, $[7, 40, 40, 200]$, $[7, 40, 30, 100]$, and $[7, 40, 30, 100]$. Significantly, image (a) needs detection of the edges.

In Fig. 7, column (2) and column (3) are results of method 1 and method 2, and both of the two methods get the undersegmentation results on Figs. 7(b), 7(c), and 7(j). This proves that method 1 and method 2 are unable to achieve good results when the structural elements are connected.

Column (4) of Fig. 7 is results of method 3, and they are similar to our preliminary segmentation results, which are shown in column (6). However, method 3 is easy to bring much more oversegmentation than our preliminary segmentation method, such as results on Figs. 7(d) and 7(j), and this problem leads to more error segmentation. One of the disadvantages of method 4 is that the classes number of structural elements in image is need to be specified, and it is hard to further extract the regions from the clustering result since the regions are connected, as shown in column (5) of Fig. 7.

To quantitatively compare the results of different methods and validate the second part of our method, the results of the various methods in Fig. 7 are further classified and merged.

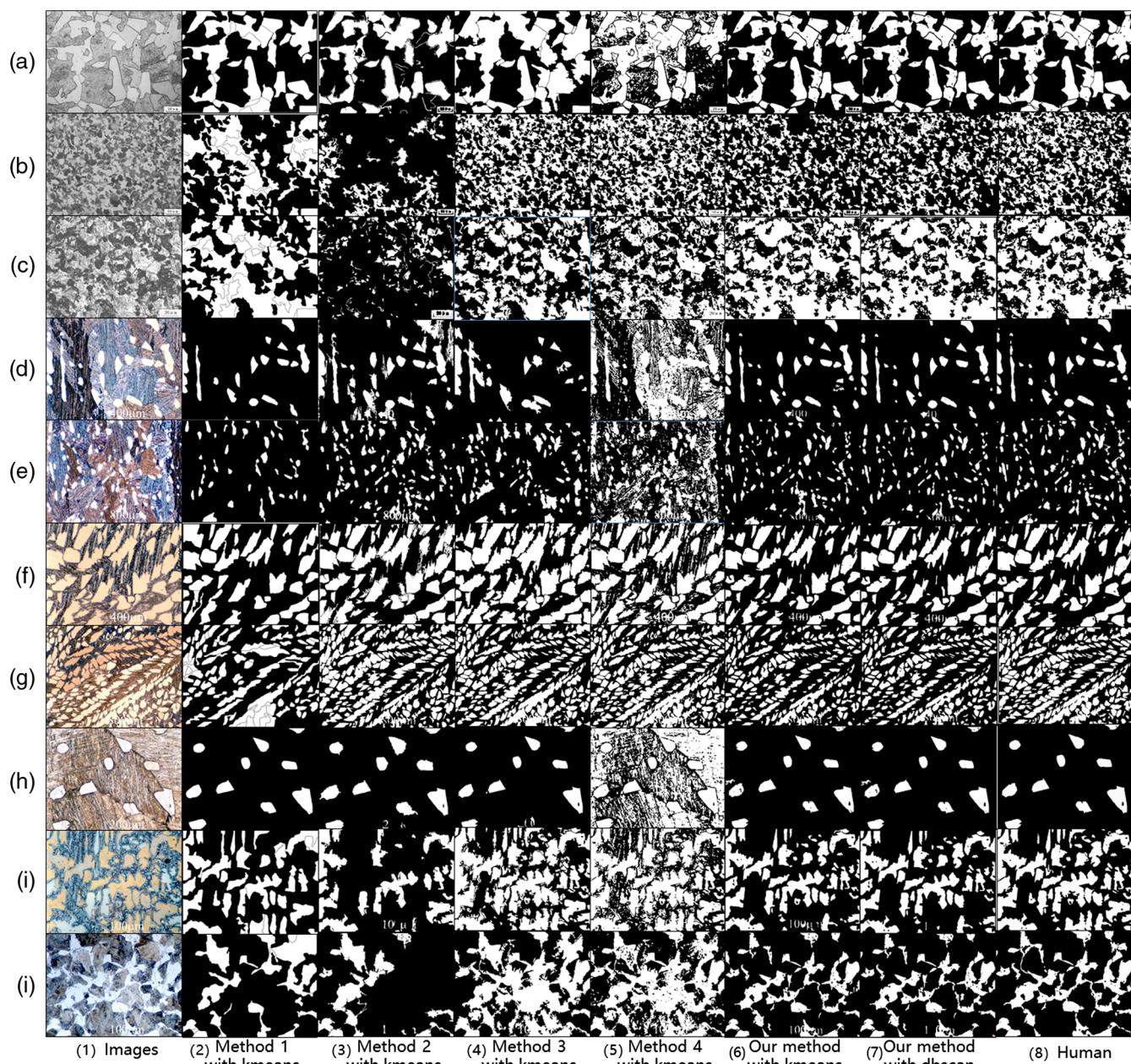


Fig. 8 Further classification and merging of the results of different segmentation methods. Images in column (1) are original images corresponding to row (1) of Fig. 7. Columns (2)–(5) show the further classification and merging of the results of method 1 to method 4, which correspond to columns (2)–(5) of Fig. 7, and k-means is adopted. The results of our method are column (6), which uses the k-means, and column (7) which applies the DBSCAN. Column (8) is the ground truth labeled by human.

For the classification, the k-means and features mentioned in Sec. 3.2 are adopted, and the parameter of k-means k is set to 2 for all images. The classification results are shown in columns (2) to (6) of Fig. 8. Since the method 4 is based on clustering, it does not need to be further merged. And to compare the classification effects of k-means and DBSCAN, DBSCAN is also applied to classify our preliminary segmentation results. The parameters of DBSCAN [$Eps, min\ Pts$] for the 10 preliminary segmentation results of our method are [0.1, 5], [0.05, 10], [0.1, 15], [0.1, 5], [0.1, 5], [0.1, 5], [0.1, 5], [0.1, 5], [0.1, 5], and [0.1, 15]. The result of DBSCAN is shown in column (7) of Fig. 8.

[0.1, 5], [0.1, 5], [0.1, 10], and [0.1, 15]. The result of DBSCAN is shown in column (7) of Fig. 8.

To evaluate the effects of various methods in the segmentation of the metallographic images, the precision–recall of classification results of different segmentation methods is calculated. For each method, the precision–recall of two classes is calculated, respectively, i.e., white class and black class. The precision–recall of various methods is illustrated in Table 1.

In general, the performance of our method with k-means exceeds other methods with k-means. Method 1 and method 2 are failed on Figs. 8(b) and 8(c) since the undersegmentation results, as shown in Fig. 7. In addition, due to the problems of method 1 on the edge detection, as mentioned in Sec. 3.1.4, some adjacent regions are connected wrongly and cause the incorrect classification results, such as the sub-figure row (a) column (2) in Fig. 8. Method 3 has much misclassification on Figs. 8(d) and 8(j). And the performance of method 4 on Figs. 8(a), 8(d) and 8(h) is poor. By comparing the column (6) of Fig. 7 and column (6) of Fig. 8, after processed by the second part of our method, the microstructural elements can be successfully segmented both in images with complex background [such as images (a), (d), and (e)] and images with simple background [such as images (b) and (c)]. In addition, the results of our method with k-means and DBSCAN are generally similar. Since the DBSCAN is able to eliminate some noises, such as the watermarks in graphs (a) and (b) of column (7) of Fig. 8, the performance of our method with DBSCAN is higher than our method with k-means.

There are some limitations in our method. First, several parameters need to be set manually. Although there are rules in the parameters setting as mentioned in Sec. 3.3, a method with no parameters can be more practical. Second, our method cannot segment some overlapped microstructures well. These problems are still need to be solved in our future works.

Table 1 The precision–recall of classification results of different segmentation methods.

Methods	Class	Precision	Recall	F-score
Method 1 with k-means	White	0.830	0.763	0.794
	Black	0.878	0.904	0.891
Method 2 with k-means	White	0.783	0.708	0.743
	Black	0.845	0.930	0.885
Method 3 with k-means	White	0.776	0.888	0.828
	Black	0.954	0.875	0.913
Method 4 with k-means	White	0.629	0.966	0.762
	Black	0.961	0.756	0.846
Our method with k-means	White	0.963	0.881	0.920
	Black	0.932	0.980	0.877
Our method with DBSCAN	White	0.974	0.883	0.926
	Black	0.938	0.982	0.959

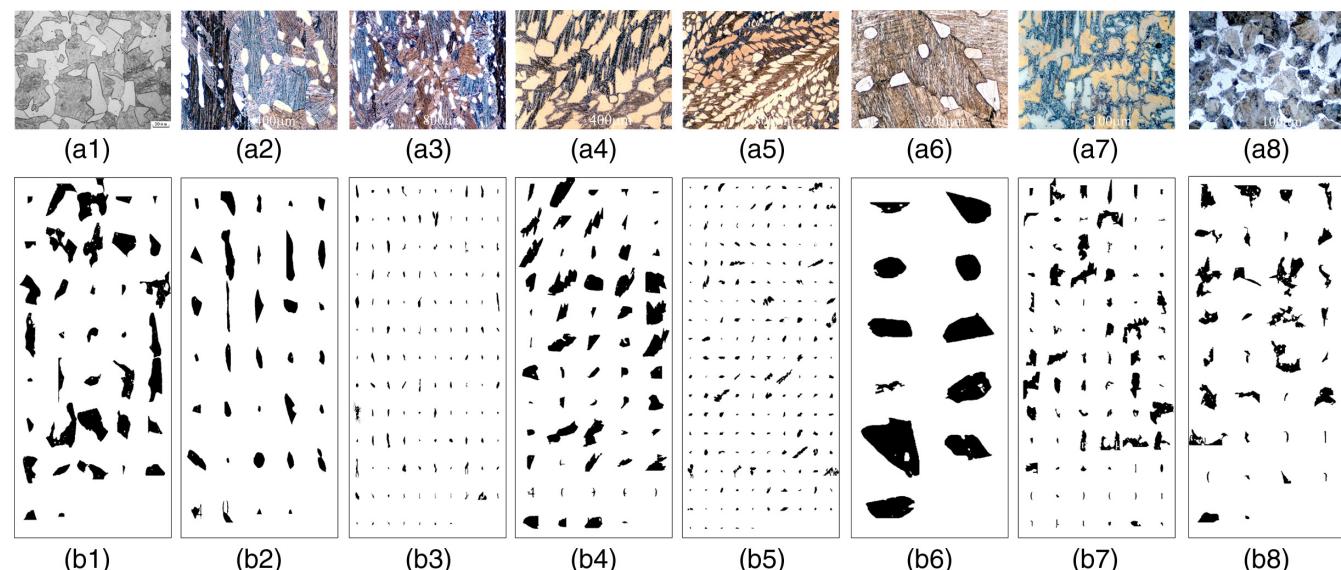


Fig. 9 The microstructural elements extracted from the segmentation results. To better show the results, the microstructural elements are painted to black. Images (a1)–(a8) are original images, which correspond to (a) and (d)–(j) of Fig. 7, and images (b1)–(b8) show the extracted microstructural elements. There is some distortion of the microstructural elements in (b1)–(b8) since the graphs are stretched to keep the same size.

Table 2 The precision–recall of our method in counting the number of the microstructural elements.

Images	Ground truth	Our method	Correct number	Precision	Recall
Image (a1)	40	37	37	1.00	0.925
Image (a2)	35	34	32	0.941	0.914
Image (a3)	119	127	113	0.889	0.949
Image (a4)	56	58	53	0.914	0.946
Image (a5)	179	185	167	0.903	0.933
Image (a6)	10	11	10	0.909	1.00
Image (a7)	73	78	69	0.885	0.945
Image (a8)	33	34	31	0.912	0.939
Mean value				0.919	0.944

4.2 Validating Our Method in the Quantitative Analysis

To evaluate the effects of our method in the quantitative analysis of the metallographic images, this experiment calculates the precision–recall of our method in counting the number of the microstructural elements, which are extracted from the segmentation results, as shown in Fig. 9. In this experiment, eight metallographic images (a1)–(a8) are used as the testing materials, which correspond to (a) and (d)–(j) of Fig. 7, and the elements extraction is based on the results of our method with DBSCAN. In order to better show the results, the microstructural elements are painted to black, as shown in Figs. 9(b1)–9(b8). The precision–recall of our method is illustrated in Table 2.

Table 2 shows the precision–recall of our method in counting the number of the microstructural elements. The ground truth is the number of the microstructural elements counted by manual in the original images. In general, the average precision on the eight metallographic images is 0.919 and the average recall is 0.944. As can be seen from Fig. 9, the error comes mainly from the watermarks and some tiny microstructural elements in the metallographic images. These problems are still remained to be improved.

5 Conclusion

The quantitative analysis of the metallographic images needs an intact and accurate segmentation of the images. However, the images often contain complex textures, which can greatly decrease the accuracy of the existing segmentation methods. In order to accurately segment the metallographic images with complex textures, this paper proposes an unsupervised segmentation method, i.e., GMFFC, combines the Gaussian filter, the mean shift method, the FloodFill, the improved FDG, feature extraction, and clustering methods. Two experiments were performed in Sec. 4, and the results demonstrate that: first, the features extracted from the images are effective for clustering; second, our method can achieve high precision in the segmentation and quantitative analysis of the metallographic images. In comparison with the existing methods, the contribution in this work includes the following:

- The oversegmentation and undersegmentation problems in the segmentation of the metallographic images with complex textures are solved.
- An improved edge detection method is proposed to make the segmentation results more accurate.
- The method is designed for the metallographic images with complex background, but it also maintains good performance for images with simple background.

Furthermore, we envision that the versatility of this method can make it useful for a large range of images.

In summary, as an accurate and efficient replacement of manual segmentation, the proposed method can be utilized for the quantitative analysis of the metallographic images. However, with further optimization in implementation, additional speed-up will be accomplished. To solve the limitations of our method mentioned in Secs. 4.1 and 4.2, i.e., the overmuch parameters and segmentation of overlapped microstructures, the end-to-end instance segmentation method based on deep learning with scarce samples is also one of our future research works.

Acknowledgments

This research is sponsored by the National Key Research and Development Program of China (Grant Nos. 2018YFB0704400 and 2018YFB0704402), Shanghai Pujiang Program (Grant No. 17PJ1402900), the National Science Foundation for Young Scientists of China (Grant No. 61603237), the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning. The authors declare that there are no relevant financial interests and potential conflicts of interests in the manuscript.

References

1. V. Korniy et al., “Quantitative analysis of metallographic images,” in *Int. Conf. Mod. Prob. Radio Eng., Telecommun., and Comput. Sci.*, IEEE, pp. 251–252 (2006).
2. D. Shi-Chao, L. Tiegen, and X. Zexin, “Extraction of enclosing image edge for metallographic structure based on canny and grayscale contour line,” *Opt. Precis. Eng.* **18**(10), 2314–2322 (2010).
3. O. Y. Povstyano et al., “Metallographic analysis and microstructural image processing of sandblasting nozzles produced by powder metallurgy methods,” *Powder Metall. Met. Ceram.* **54**(3–4), 234–240 (2015).
4. H. Zhao and M. Comer, “A hybrid Markov random field/marked point process model for analysis of materials images,” *IEEE Trans. Comput. Imaging* **2**(4), 395–407 (2016).
5. L. Duval et al., “Image processing for materials characterization: issues, challenges and opportunities,” in *IEEE Int. Conf. Image Process.*, pp. 4862–4866 (2014).
6. J. Hu, A. Susanto, and D. Koleva, “Image processing of cement-based materials in conditions of current flow for microstructural analysis,” in *Int. Conf. Audio, Language and Image Process. (ICALIP)*, IEEE, pp. 246–250 (2016).
7. D. Wang, “Research of metallographic analysis based on image processing technology,” *Foundry Technol.* **5**, 39 (2013).
8. Y. Chen and J. Chen, “A watershed segmentation algorithm based on ridge detection and rapid region merging,” in *IEEE Int. Conf. Signal Process., Commun. and Comput. (ICSPCC)*, IEEE, pp. 420–424 (2014).
9. J. Liu and J. Chen, “An improved iterative watershed according to ridge detection for segmentation of metallographic image,” *Metallogr. Image* **8**, 8–13 (2012).
10. V. H. C. De Albuquerque, J. M. R. Tavares, and P. C. Cortez, “Quantification of the microstructures of hypoeutectic white cast iron using mathematical morphology and an artificial neural network,” *Int. J. Microstruct. Mater. Prop.* **5**(1), 52–64 (2010).
11. H. E. Wei-Na and L. L. Zhang, “Study on artificial neuronal networks applied on microstructure segmentation from metallographic images,” *Electron. Des. Eng.* **3**, 49 (2013).
12. L. Chen et al., “Two-dimensional fuzzy clustering algorithm (2DFCM) for metallographic image segmentation based on spatial information,” in

- 2nd Int. Conf. Inf. Sci. and Control Eng. (ICISCE)*, IEEE, pp. 519–521 (2015).
13. S. M. Azimi et al., “Advanced steel microstructural classification by deep learning methods,” *Sci. Rep.* **8**(1), 2128 (2018).
 14. J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 3431–3440 (2015).
 15. R. Kondo et al., “Microstructure recognition using convolutional neural networks for prediction of ionic conductivity in ceramics,” *Acta Mater.* **141**, 29–38 (2017).
 16. P. Dhankhar and N. Sahu, “A review and research of edge detection techniques for image segmentation,” *Int. J. Comput. Sci. Mob. Comput.* **2**(7), 86–92 (2013).
 17. A. Bali and S. N. Singh, “A review on the strategies and techniques of image segmentation,” in *Fifth Int. Conf. Adv. Comput. and Commun. Technol. (ACCT)*, IEEE, pp. 113–120 (2015).
 18. T. Volkenandt, S. Freitag, and M. Rauscher, “Machine learning powered image segmentation,” *Microsc. Microanal.* **24**(S1), 520–521 (2018).
 19. H. Ajmal et al., “Convolutional neural network based image segmentation: a review,” *Proc. SPIE* **10649**, 106490N (2018).
 20. N. Sharma, M. Mishra, and M. Shrivastava, “Colour image segmentation techniques and issues: an approach,” *Int. J. Sci. Technol. Res.* **1**(4), 9–12 (2012).
 21. J. Ning et al., “Interactive image segmentation by maximal similarity based region merging,” *Pattern Recognit.* **43**(2), 445–456 (2010).
 22. J. B. Roerdink and A. Meijster, “The watershed transform: definitions, algorithms and parallelization strategies,” *Fundam. Inf.* **41**(1–2), 187–228 (2000).
 23. J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000).
 24. D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 603–619 (2002).
 25. A. K. Sinop and L. Grady, “A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm,” in *IEEE 11th Int. Conf. Comput. Vision*, IEEE, pp. 1–8 (2007).
 26. C. Rother, V. Kolmogorov, and A. Blake, “Grabcut: interactive foreground extraction using iterated graph cuts,” *ACM Trans. Graphics* **23**(3), 309–314 (2004).
 27. P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vision* **59**(2), 167–181 (2004).
 28. Y. Li et al., “Lazy snapping,” *ACM Trans. Graphics* **23**(3), 303–308 (2004).
 29. N. Senthilkumaran and R. Rajesh, “Image segmentation—a survey of soft computing approaches,” in *Int. Conf. Adv. Recent Technol. Commun. and Comput.*, IEEE, pp. 844–846 (2009).
 30. R. Muthukrishnan and M. Radha, “Edge detection techniques for image segmentation,” *Int. J. Comput. Sci. Inf. Technol.* **3**(6), 259 (2011).
 31. J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(6), 679–698 (1986).
 32. M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: active contour models,” *Int. J. Comput. Vision* **1**(4), 321–331 (1988).
 33. S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer, New York (2003).
 34. L. Chunming et al., “Minimization of region-scalable fitting energy for image segmentation,” *IEEE Trans. Image Process.* **17**(10), 1940–1949 (2008).
 35. H. Li et al., “Edge detection of noisy images based on cellular neural networks,” *Commun. Nonlinear Sci. Numer. Simul.* **16**(9), 3746–3759 (2011).
 36. D. R. Martin, C. C. Fowlkes, and J. Malik, “Learning to detect natural image boundaries using local brightness, color, and texture cues,” *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(5), 530–549 (2004).
 37. V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: a deep convolutional encoder-decoder architecture for image segmentation,” arXiv:1511.00561 (2015).
 38. L. C. Chen et al., “Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(4), 834–848 (2018).
 39. L. C. Chen et al., “Rethinking atrous convolution for semantic image segmentation,” arXiv:1706.05587 (2017).
 40. K. Ito, “Gaussian filter for nonlinear filtering problems,” in *Proc. 39th IEEE Conf. Decis. and Control*, IEEE, Vol. 2, pp. 1218–1223 (2000).
 41. C. Hua et al., “Performance reliability estimation method based on adaptive failure threshold,” *Mech. Syst. Sig. Process.* **36**(2), 505–519 (2013).
 42. J. Lee and H. Kang, “Flood fill mean shift: a robust segmentation algorithm,” *Int. J. Control Autom. Syst.* **8**(6), 1313–1319 (2010).
 43. P. Kovesi, “Phase congruency detects corners and edges,” in *Aust. Pattern Recognit. Soc. Conf.: DICTA* (2003).
 44. H. Kang, S. Lee, and C. K. Chui, “Coherent line drawing,” in *Proc. 5th Int. Symp. Non-Photorealistic Anim. and Rendering*, ACM, pp. 43–50 (2007).
 45. J. E. Kyprianidis and J. Döllner, “Image abstraction by structure adaptive filtering,” in *Proc. EG UK Theory and Pract. Comput. Graphics*, pp. 51–58 (2008).
 46. N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, IEEE, Vol. 1, pp. 886–893 (2005).
 47. H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: speeded up robust features,” *Lect. Notes Comput. Sci.* **3951**, 404–417 (2006).
 48. R. M. Haralick et al., “Textural features for image classification,” *IEEE Trans. Syst. Man Cybern. SMC-3*(6), 610–621 (1973).
 49. T. Ojala, M. Pietikainen, and T. Maenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 971–987 (2002).
 50. F. Tajeripour, E. Kabir, and A. Sheikhi, “Fabric defect detection using modified local binary patterns,” *EURASIP J. Adv. Signal Process.* **2008**, 783898 (2007).
 51. M.-K. Hu, “Visual pattern recognition by moment invariants,” *IRE Trans. Inf. Theory* **8**(2), 179–187 (1962).
 52. N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979).
 53. A. Khotanzad and Y. H. Hong, “Invariant image recognition by Zernike moments,” *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(5), 489–497 (1990).
 54. W. Lievers and A. Pilkey, “An evaluation of global thresholding techniques for the automatic image segmentation of automotive aluminum sheet alloys,” *Mater. Sci. Eng. A* **381**(1–2), 134–142 (2004).
 55. M. Varma and A. Zisserman, “A statistical approach to texture classification from single images,” *Int. J. Comput. Vision* **62**(1–2), 61–81 (2005).
 56. J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science* **290**(5500), 2319–2323 (2000).
 57. T. Kanungo et al., “An efficient K-means clustering algorithm: analysis and implementation,” *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 881–892 (2002).
 58. M. Ester et al., “A density-based algorithm for discovering clusters in large spatial databases with noise,” *KDD Proc.* **96**(34), 226–231 (1996).
 59. Z. Barber, J. Leake, and T. Clyne, “The DoITPoMS project—a web-based initiative for teaching and learning materials science,” *J. Mater. Educ.* **29**(1–2), 7–16 (2007).
 60. P. Arbelaez et al., “Contour detection and hierarchical image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(5), 898–916 (2011).
 61. H. T. Nguyen, M. Worring, and R. Van Den Boomgaard, “Watersnakes: energy-driven watershed segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(3), 330–342 (2003).
 62. H. Ng et al., “Medical image segmentation using K-means clustering and improved watershed algorithm,” in *IEEE Southwest Symp. Image Anal. and Interpret.*, IEEE, pp. 61–65 (2006).
 63. K. Parvati, P. Rao, and M. M. Das, “Image segmentation using grayscale morphology and marker-controlled watershed transformation,” *Discrete Dyn. Nat. Soc.* **2008**, 1–8 (2008).
 64. I. The MathWorks, “Marker-controlled watershed segmentation,” <https://www.mathworks.com/help/images/marker-controlled-watershed-segmentation.html> (17 December 2018).
 65. W. Cai, S. Chen, and D. Zhang, “Fast and robust fuzzy C-means clustering algorithms incorporating local information for image segmentation,” *Pattern Recognit.* **40**(3), 825–838 (2007).

Yuxing Han is working in Shanghai University. He also is a research fellow in Tokyo Institute of Technology, Japan, from 2013. His research interests include image processing for materials' microstructure, molecular robotics, computer vision, and robot vision.

Chuanbin Lai is a graduate student of computer engineering and science, Shanghai University. His main research interests include image processing for materials' microstructure, computer vision, and machine learning.

Bing Wang is a professor of computer engineering and science, Shanghai University. Her main research interests include mathematical modelling of epidemics with complex network theory, design of robust network structure, and network synchronizability.

Hui Gu is a professor of materials' microstructures, School of Materials Science and Engineering, and Materials Genome Institute, Shanghai University. His research interest lies in hierarchical phase-microstructures-interface relationship of advanced materials and focuses on development of fully quantitative methods of materials microstructures.