

Multiscale Adaptive Edge Detector for Images Based on a Novel Standard Deviation Map

Yunhao Cui¹, Yi An¹, *Member, IEEE*, Wei Sun¹, Huosheng Hu², *Senior Member, IEEE*, and Xueguan Song¹

Abstract—Edge detection plays an important role in many applications, such as industrial inspection and automatic driving. However, it is difficult to effectively distinguish between faint edges and noise, which may result in losing effective edges or generating spurious edges. This will reduce the accuracy of edge detection. In addition, some parameters need to be set artificially. In the case of the fixed parameters, the overall performance of edge detection on different images is not high. The adaptivity of edge detection needs to be improved further. To solve these problems, this article proposes a multiscale adaptive edge detector for images. First, multiscale pyramid images are constructed from an input image to provide multiscale features for edge detection. At each scale, a gradient map and a novel standard deviation map are calculated based on the gradients and the statistical characteristics of the local gradient differences, respectively, to accurately distinguish the edges from the background and noise. By using these two feature maps, candidate edges are adaptively identified from the image by using pixel-by-pixel detection. Then, candidate edges at different scales are thinned and fused together based on a novel voting mechanism. Finally, a binarized edge map is obtained by using adaptive hysteresis linking. These steps make the proposed edge detector accurate and adaptive. Experiments demonstrate that the proposed edge detector achieves good performance, which is beneficial to measurement applications.

Index Terms—Edge detection, gradient map, image denoising, image processing, multiscale pyramid.

I. INTRODUCTION

EDGES carry vital feature information, which corresponds to sharp variations of pixel intensities in an image [1]. The essence of edge detection is to identify and locate the edge pixels from a large amount of redundant information. Edge detection plays a very important role in many measurement applications, such as defect detection, noncontact dimension measurement, intelligent manufacturing, and intelligent driving [2]–[7].

Manuscript received April 4, 2021; revised May 8, 2021; accepted May 17, 2021. Date of publication May 26, 2021; date of current version June 8, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61673083, in part by the Science and Technology Major Project of Shanxi Province under Grant 20191101014, and in part by the State Key Laboratory of Mining Equipment and Intelligent Manufacturing. The Associate Editor coordinating the review process was Chi-Hung Hwang. (*Corresponding author: Yi An.*)

Yunhao Cui, Wei Sun, and Xueguan Song are with the School of Mechanical Engineering, Dalian University of Technology, Dalian 116023, China (e-mail: cyhlovegl@mail.dlut.edu.cn; sunwei@dlut.edu.cn; sxg@dlut.edu.cn).

Yi An is with the School of Control Science and Engineering, Dalian University of Technology, Dalian 116023, China (e-mail: anyi@dlut.edu.cn).

Huosheng Hu is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: hhu@essex.ac.uk).

Digital Object Identifier 10.1109/TIM.2021.3083888

Existing edge detectors can be classified into two kinds: supervised edge detectors and unsupervised edge detectors. Although the supervised edge detectors have made important research progress in recent years, their training processes rely on a large number of well-labeled images [8], [9]. The labeling process is time-consuming and expensive. In addition, some parameters need to be set artificially in the process of training. These characteristics limit the performance of the supervised edge detectors.

Compared with the supervised edge detectors, unsupervised edge detectors adopt inherent image characteristics to detect edges, without the need for artificially labeled images. This greatly reduces the costs of application and improves the universality. The unsupervised edge detectors mainly include differential-based edge detectors (DBEDs) [10]–[13], active contour edge detectors [14], [15] and statistical edge detectors [16], [17].

Among the existing edge detectors, the DBEDs are widely used in practical engineering, since the differentiation of intensity is closer to the essence of edges. As a pioneer of the DBEDs, Canny [10] cascades a series of key technologies, including the Gaussian filter, gradient map calculation, nonmaximum suppression and hysteresis linking, which have been widely used by the subsequent edge detectors. The Gaussian filter can suppress noise pixels and improve the accuracy of edge detection. Gradient map calculation is used to distinguish the edges from the background. Nonmaximum suppression is used to thin the edges, which can keep the unique response to a single edge. Hysteresis linking is beneficial to improve the accuracy of edge localization and keep good edge continuity.

However, there exist some problems which will reduce the accuracy and adaptivity of edge detectors. First, it is difficult to effectively distinguish between faint edges and noise pixels, which may result in losing effective edges or generating spurious edges. This will reduce the accuracy of edge detection. Second, the Gaussian filter may blur the edges when filtering out noise, which will degrade the localization precision and the resolution of edges. Especially for the faint edges, it will cause serious edge missing. Third, the adaptivity is reduced because some parameters need to be set artificially and the overall performance of edge detection on different images will be degraded in the case of the fixed parameters. These problems limit the performance of edge detectors in measurement applications.

To solve these problems, we propose a multiscale adaptive edge detector (MAED). The main steps are as follows:

1) multiscale pyramid images are constructed by down sampling an input image to provide multiscale features for edge detection; 2) at each scale, a gradient feature map and a novel standard deviation feature map are calculated based on the gradients and the statistical characteristics of the local gradient differences, respectively, to accurately distinguish edge pixels from the background and noise; 3) the candidate edges are adaptively identified from the image by using pixel-by-pixel detection based on these two feature maps, and they are thinned by using nonmaximal suppression; 4) these thinned candidate edges at different scales are fused together based on a voting mechanism; and 5) adaptive hysteresis linking is developed to obtain the final binarized edge map and improve the adaptivity of our method. Experiments show that our method achieves good performance, which is beneficial to measurement applications.

The remainder of this article is organized as follows. Section II reviews the related works. Section III describes the proposed method. Section IV shows the experimental results and analysis. The conclusions are drawn in Section V.

II. RELATED WORKS

Some researchers have carried out related studies and made certain progress to improve the accuracy and adaptivity of edge detection.

A. Methods for Accuracy Improvement

The Gaussian filter plays an important role in Canny and many subsequent edge detectors. However, it can cause edge blurring while reducing noise. This will reduce the accuracy of edge detection. Many researchers have proposed improvement methods to solve this problem.

Perona and Malik [18] adopted a nonlinear anisotropic diffusion method instead of the Gaussian filter by smoothing an image only in the direction orthogonal to the gradient. Tomasi and Manduchi [19] proposed a bilateral filter which considers both the spatial relationship between pixels and the intensity difference. The bilateral filter can effectively improve the edge preservation effect when reducing noise. However, compared with the Gaussian filter, more parameters need to be set artificially. Mafi *et al.* [20] presented a new filter that embeds the switching median filter and the fixed weighted mean filter to keep edge details when filtering the high-intensity impulse noise. Qiu *et al.* [21] derived an explicit edge-preserving image filter based on a local linear model and an unbiased risk estimate. Paras and Vipin [22] presented an image denoising method based on wavelet transforms, which can preserve edges.

Some researchers adopted the anisotropic Gaussian kernels for image denoising [23]–[25]. Compared with the traditional isotropic Gaussian kernel, these improved algorithms have achieved some improvements in preventing edge missing. However, they increase the computational complexity and introduce more parameters to be set artificially. In recent years, other kinds of methods have also made some progresses in improving the accuracy of edge detection. For example, Zhang *et al.* [26] proposed a clustering based super-pixel

method for edge detection and achieved good performance. Cihan and Cuneyt [27] developed a popular detector edge drawing for extracting continuous contours by calculating a set of anchor pixels and connecting them based on the gradient direction. James and Steven [28] systematically analyzed the influences of different scales in edge detection.

B. Methods for Adaptivity Improvement

To improve the adaptivity of edge detection, Wang *et al.* [29] proposed an adaptive fourth-order partial differential for image denoising. Yitzhaky and Peli [30] adopted the Chi-square test to determine the best edge detector parameters from candidate hysteresis thresholds. Medina-Carnicer *et al.* [31], [32] made a series of studies in automatically determining the hysteresis thresholds. In [31], a criterion is provided to reduce the number of initial candidates for choosing hysteresis thresholds. In [32], the hysteresis process is transformed into a unimodal thresholding problem. Han *et al.* [33] proposed an adaptive method to obtain the dual thresholds based on the Otsu method. However, they ignored the limitations of the Otsu method, which results in the inaccurate setting of the low threshold. For two types of images with different edge information, Rong *et al.* [34] proposed two corresponding adaptive threshold selection methods based on the mean and standard deviation of the image gradient amplitude. Shi *et al.* [35] proposed an adaptive edge detector based on the morphological filtering.

C. Discussion

By analyzing the above-mentioned methods, we can find that although many contributions have been made in improving the accuracy and adaptivity of edge detection, there still exist some problems as follows. Some methods for improving the accuracy introduce more parameters that need to be set artificially, which reduces the adaptivity of edge detection. Some methods provide candidate values to narrow the range of parameters selection, which is insufficient to improve the adaptivity. Furthermore, the parameters obtained by some adaptive methods are not accurate enough. In summary, further studies for improving the accuracy and adaptivity of edge detection at the same time still need to be done.

III. MULTISCALE ADAPTIVE EDGE DETECTOR

The framework of our method (MAED) is shown in Fig. 1. Our method mainly consists four steps: constructing multiscale pyramid images, candidate edge detection, multiscale edge merging, and adaptive hysteresis linking.

A. Constructing Multiscale Pyramid Images

To make better use of the multiscale characteristics of the image and improve the integrity and antinoise ability of edge detection, multiscale pyramid images are constructed.

The size of an input gray level image I_0 is $s_0 = w \times h$. As the scale level increases, the image size is reduced by half. The size of the image I_l on the scale level l is

$$s_l = w/2^l \times h/2^l, \quad l \in \{0, 1, \dots, n\}. \quad (1)$$

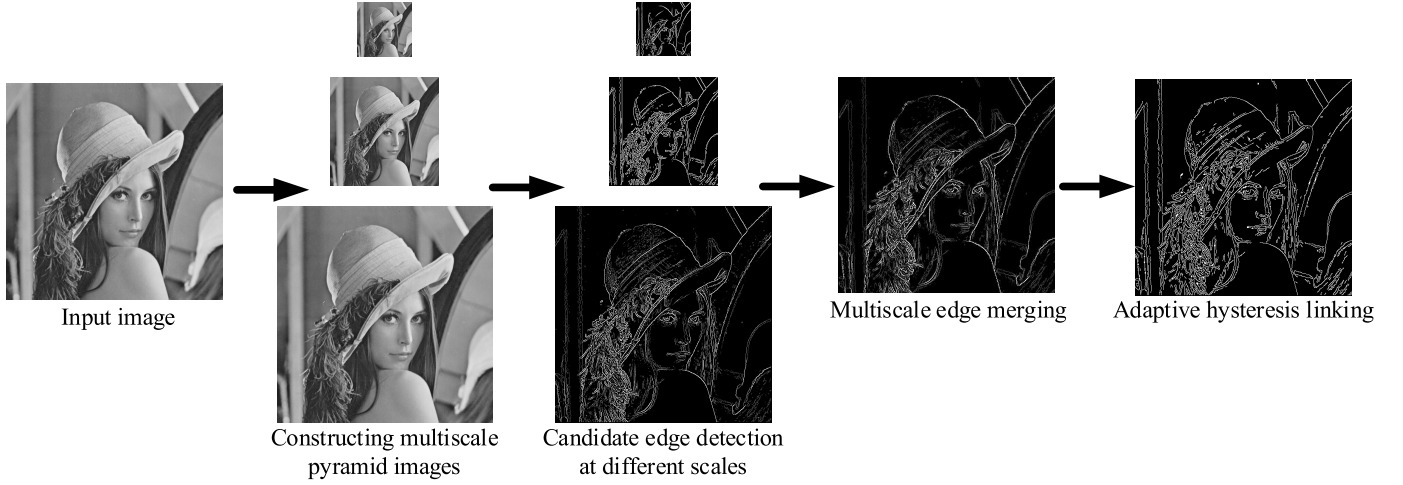


Fig. 1. Framework of the proposed method.

In this study, we adopt the Gaussian kernel as the pyramid generation kernel to blur the image I_l , because the Gaussian kernel will not introduce other noises and its calculation is simple and convenient. Multiscale pyramid images are denoted as $L = \{I_l | 0 \leq l \leq n\}$.

B. Candidate Edge Detection

The candidate edge detection is performed on each image I_l . It mainly includes the following steps.

1) *Calculating the Gradient Map*: The gradient represents the variances of image gray levels. By adopting the gradient map, the background with small gray variances can be effectively distinguished from edge pixels and noise pixels with sharp gray variances.

As a classical first-order differential operator, the Sobel operator is used by many edge detectors due to its simplicity and good effect. Therefore, we adopt the Sobel operator to calculate the gradient map by using a pair of convolution masks S_x and S_y . For example, the 3×3 convolution masks are obtained by

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (2)$$

For each image I_l , the gradient maps G_{lx} and G_{ly} along the x - and y -directions are obtained by

$$G_{lx} = S_x * I_l \quad (3)$$

$$G_{ly} = S_y * I_l \quad (4)$$

where $*$ represents the convolution operation. The gradient magnitude $G_l(i, j)$ of each pixel $I_l(i, j)$ is calculated by

$$G_l(i, j) = \sqrt{G_{lx}^2(i, j) + G_{ly}^2(i, j)}. \quad (5)$$

The gradient direction $\theta_l(i, j)$ of each pixel $I_l(i, j)$ is obtained by

$$\theta_l(i, j) = \arctan(G_{ly}(i, j)/G_{lx}(i, j)). \quad (6)$$

Then, we obtain the gradient map G_l .

2) *Calculating the Standard Deviation Map*: According to the gradient, the background with small gray variances can be effectively distinguished from edges and noise with sharp gray variances. However, it is difficult to effectively distinguish between edges and noise using the gradient map, which may result in losing effective edges or generating spurious edges. This will reduce the accuracy of edge detection. To solve this problem, we develop the standard deviation map based on the statistical characteristics of the local gradient differences. It is easy to identify edges from noise based on the standard deviations of the gradient differences in the local neighborhoods.

The gradient difference $d_l^{ij}(m, n)$ in the local neighborhood $N_l(i, j) = \{G_l(i + m, j + n) | -k_l \leq m, n \leq k_l\}$ of the pixel $G_l(i, j)$ on the gradient map is calculated by

$$d_l^{ij}(m, n) = |G_l(i + m, j + n) - G_l(i, j)|. \quad (7)$$

The mean $M_l(i, j)$ of the gradient differences in the local neighborhood $N_l(i, j)$ of the pixel $G_l(i, j)$ is computed as

$$M_l(i, j) = \sum_{m=-k_l}^{k_l} \sum_{n=-k_l}^{k_l} d_l^{ij}(m, n) / ((2k_l + 1)^2 - 1). \quad (8)$$

The standard deviation $S_l(i, j)$ of the gradient differences in the local neighborhood $N_l(i, j)$ of the pixel $G_l(i, j)$ is computed as

$$S_l(i, j) = \sqrt{\frac{\sum_{m=-k_l}^{k_l} \sum_{n=-k_l}^{k_l} (d_l^{ij}(m, n) - M_l(i, j))^2}{(2k_l + 1)^2 - 1}}. \quad (9)$$

In order to show the contributions of the standard deviation map, we compare it with the gradient map. Fig. 2(a) shows the gradient map. As can be observed, the background and foreground can be well distinguished. However, there are many noise pixels, which are great obstacles for edge extraction. As marked in the partial enlarged drawing of the gradient map, the value of the noise pixel at coordinate (261, 49) in the red box is 227, which is very close to the value 255 of the adjacent edge pixel in the blue box. In summary, it is very difficult to distinguish edge pixels and noise pixels because of the small

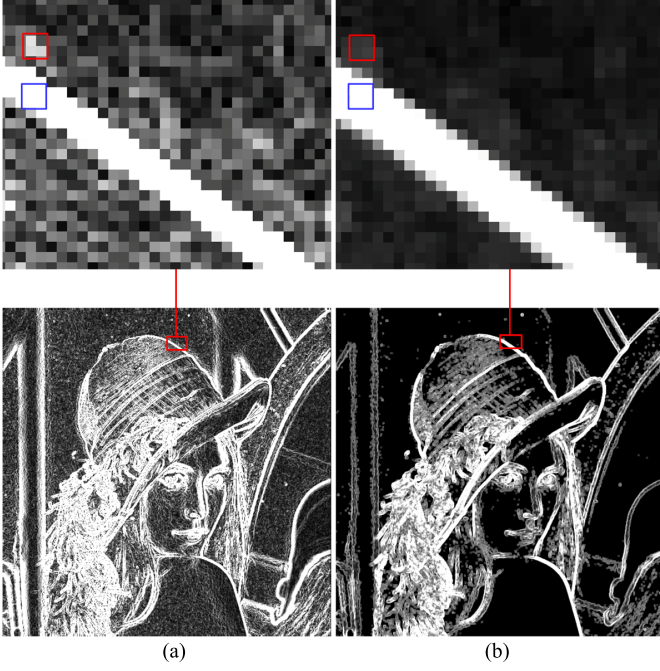


Fig. 2. Comparison of the (a) gradient map and (b) standard deviation map.

difference between them on the gradient map. This will cause the loss of edge pixels or the generation of spurious edges.

Fig. 2(b) shows the standard deviation map. As can be observed, most of the noise pixels have been effectively suppressed. This can avoid the influence of noise pixels on edge detection. As marked in the partial enlarged drawing of the standard deviation map, the value of the noise pixel at the same coordinate (261, 49) in the red box is 55, which is much lower than the value 255 of the adjacent edge pixel in the blue box. In summary, it is easy to distinguish edge pixels and noise pixels on the standard deviation map because of the huge difference between them. This is critical to improve the accuracy of edge detection.

The window sizes of the convolution masks and the local neighborhood have important influences on the gradient map and the standard deviation map, respectively. For all the edge detection methods, the setting of the window size is a common and difficult problem. When the small window size is used, more details are reserved. However, it is easy to be disturbed by noise and produce spurious edges. When the large window size is used, the detected edge map contains fewer spurious edges. However, some details will be lost. In this article, we give the following suggestions to select the appropriate window size according to different images and tasks. If the image has the simple background or we need to extract more details, it is recommended to use the small window size, i.e., 3×3 . If the image has the complex background or we want to extract important contours and ignore details, it is recommended to use the large window size, i.e., 7×7 . The gradient map and the standard deviation map follow the same selection rules.

3) *Pixel-by-Pixel Edge Detection*: To effectively extract edge pixels from the image, we comprehensively adopt the

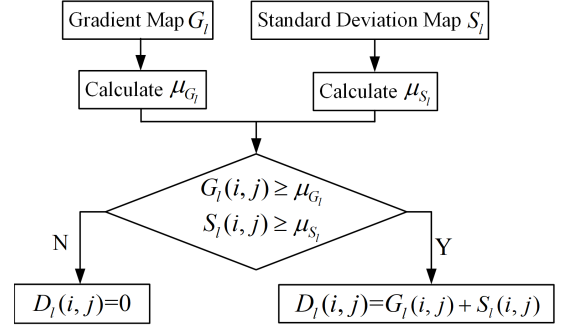


Fig. 3. Pixel-by-pixel detection.

statistical characteristics of the gradient map and the standard deviation map to perform pixel-by-pixel detection, as shown in Fig. 3.

The mean μ_{G_l} in the gradient map and the mean μ_{S_l} in the standard deviation map are calculated by

$$\mu_{G_l} = \sum_{i=0}^{w/2^l-1} \sum_{j=0}^{h/2^l-1} G_l(i, j) / (w/2^l \times h/2^l) \quad (10)$$

$$\mu_{S_l} = \sum_{i=0}^{w/2^l-1} \sum_{j=0}^{h/2^l-1} S_l(i, j) / (w/2^l \times h/2^l). \quad (11)$$

Now, we create a candidate edge map D_l with the size of $w/2^l \times h/2^l$. For each pixel $D_l(i, j)$, if its corresponding pixels $G_l(i, j)$ and $S_l(i, j)$ on the gradient map and the standard deviation map meet the following conditions:

$$\begin{cases} G_l(i, j) \geq \mu_{G_l} \\ S_l(i, j) \geq \mu_{S_l} \end{cases} \quad (12)$$

it is identified as a candidate edge pixel. Its value is set as

$$D_l(i, j) = G_l(i, j) + S_l(i, j) \quad (13)$$

where $G_l(i, j)$ and $S_l(i, j)$ are normalized to $[0, 255]$. If not, it is identified as noise. Its value is set as

$$D_l(i, j) = 0. \quad (14)$$

Compared with the edge pixels and noise pixels, the background pixels have smaller changes in grayscale. Therefore, their gradients are much lower than the mean in the gradient map. As a result, the background pixels can be effectively removed by using the mean of the gradients as the threshold, as shown in Fig. 4(a). The gradients of the edge pixels and the noise pixels are similar in the gradient map. Therefore, they cannot be effectively distinguished by the gradients, as shown in Fig. 4(a). Compared with the edge pixels, the noise pixels have smaller standard deviations. Therefore, their standard deviations are much lower than the mean in the standard deviation map. As a result, the noise pixels can be effectively removed by using the mean of the standard deviations as the threshold, as shown in Fig. 4(b). Therefore, by using the means of both the gradients and standard deviations as the thresholds, the background pixels and noise pixels can be effectively removed.

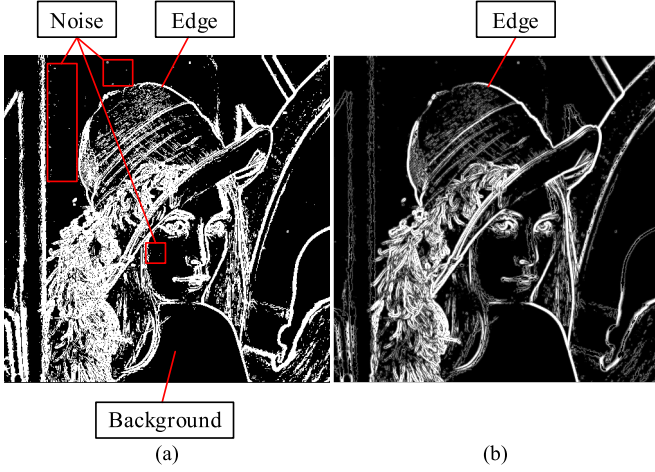


Fig. 4. Results of (a) removing the background based on the mean of gradients and (b) removing the background and noise based on both the means of the gradients and standard deviations.

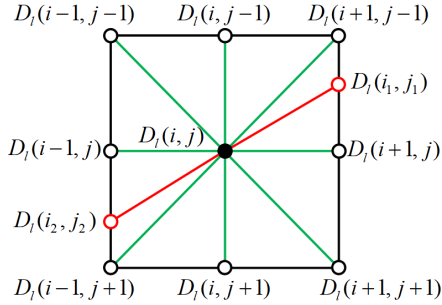


Fig. 5. Nonmaximum suppression.

4) *Nonmaximal Suppression*: To make the candidate edges thinner, we apply nonmaximal suppression to thin the candidate edges to ensure that each edge is one-pixel width. Nonmaximum suppression is to compare a candidate edge pixel $D_l(i, j)$ with its adjacent pixels along the gradient direction to determine whether $D_l(i, j)$ is a local maximum. If it is the local maximum, its value is kept unchanged; if not, its value is set to zero in the candidate edge map D_l .

As shown in Fig. 5, assume $D_l(i, j)$ is a candidate edge pixel, we take its gradient direction $\theta_l(i, j) \in [0, \pi/4]$ represented by the red line as an example, the value of a neighboring pixel $D_l(i_1, j_1)$ is calculated by

$$D_l(i_1, j_1) = D_l(i+1, j) + (D_l(i+1, j-1) - D_l(i+1, j))\tan\theta_l(i, j). \quad (15)$$

The value of the other neighboring pixel $D_l(i_2, j_2)$ is computed by

$$D_l(i_2, j_2) = D_l(i-1, j) + (D_l(i-1, j+1) - D_l(i-1, j))\tan\theta_l(i, j). \quad (16)$$

If the pixel $D_l(i, j)$ meets

$$D_l(i, j) < D_l(i_1, j_1) \text{ or } D_l(i, j) < D_l(i_2, j_2) \quad (17)$$

it is suppressed as

$$D_l(i, j) = 0. \quad (18)$$

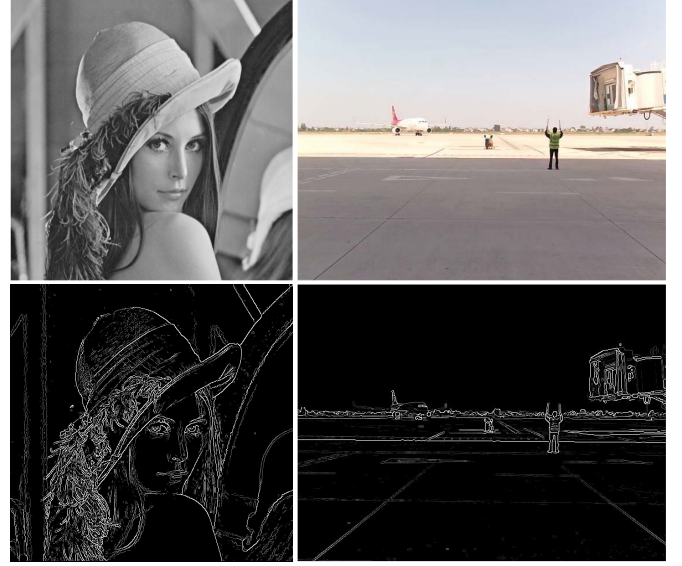


Fig. 6. Candidate edge maps D_l of two images and their original images.

If $D_l(i, j)$ is a local maximum, its value is kept unchanged. Other gradient directions can be calculated in a similar way.

Through candidate edge detection, the noise pixels are suppressed, and the width of the candidate edges is thinned on each scale level. The candidate edge maps D_l of two images on the scale level $l = 0$ are shown in Fig. 6.

C. Multiscale Edge Merging

The thinned candidate edge maps at different scales are fused to get the merged edge map in this step. A fusion feature map F with the size $w \times h$ is constructed based on the gradient map and the standard deviation map. The value of $F(i, j)$ is computed by

$$F(i, j) = G_0(i, j) + S_0(i, j). \quad (19)$$

A multiscale merged edge map M with the size $w \times h$ is also created. On each scale level l , if $D_l(i, j)$ is a candidate edge pixel, its corresponding pixel $M(i \times 2^l, j \times 2^l)$ and l adjacent pixels of $M(i \times 2^l, j \times 2^l)$ along the direction perpendicular to the gradient direction are voted in the multiscale merged edge map M .

For each pixel $M(i, j)$, its votes are counted. If the number of the votes is larger than or equal to $\lceil (n+1)/2 \rceil$, $M(i, j)$ is considered as a candidate edge pixel, where $\lceil \cdot \rceil$ represents the operation of round up and n is the number of scale levels. And, its value is given by

$$M(i, j) = F(i, j). \quad (20)$$

By utilizing the advantages of multiscale features, the thinned candidate edge maps at different scales are merged based on a novel voting mechanism. It makes the fused edge map robust to noise and have good edge continuity.

D. Adaptive Hysteresis Linking

To improve the continuity of the edges, hysteresis linking is adopted to further process the merged edge map M . Hysteresis

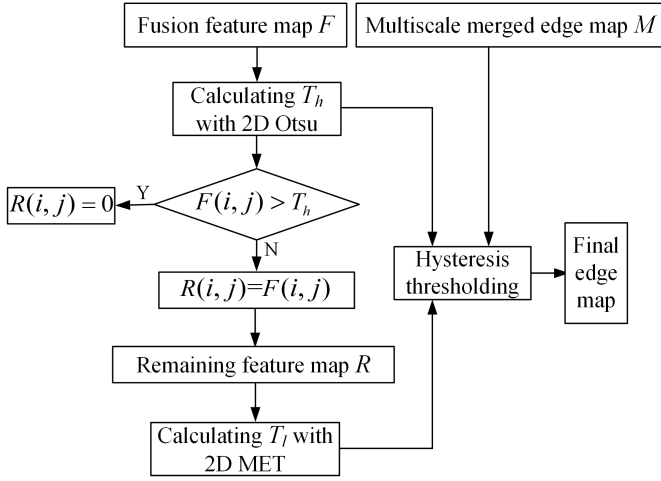


Fig. 7. Adaptive hysteresis linking.

linking regards the pixels whose intensities are greater than the high threshold as edge pixels, and the pixels whose intensities are less than the low threshold as nonedge pixels. For the pixels whose intensities are between the high threshold and the low threshold, if they are located in the local neighborhoods of edge pixels, they will also be regarded as edge pixels.

The traditional hysteresis linking method needs to set high and low thresholds artificially, which reduces the adaptivity of edge detection. To solve this problem, we propose an adaptive hysteresis linking method, which can improve the adaptivity of edge detection, as shown in Fig. 7.

1) *Calculating the High Threshold*: The 2-D Otsu considers both the gray value distribution of pixels and the average gray value distribution of their neighboring pixels [36]. It calculates the optimal threshold by maximizing the between-class variance. The high threshold T_h of the fusion feature map F is calculated by

$$T_h = k \times \text{Otsu}(F) \quad (21)$$

where Otsu means the 2-D Otsu method and $k \in (0, 1)$ is the adjustment factor. The default value of k is 0.8.

2) *Thresholding With T_h* : T_h is adopted to threshold the fusion feature map F . We create a remaining feature map R . The value of the pixel $R(i, j)$ is calculated by

$$\begin{cases} R(i, j) = 0, & \text{if } F(i, j) > T_h \\ R(i, j) = F(i, j), & \text{if } F(i, j) \leq T_h. \end{cases} \quad (22)$$

3) *Calculating the Low Threshold*: In this step, we need to obtain the low threshold T_l of the remaining feature map R . However, after the strong edges are removed, the remaining edges are very blurred, which are difficult to be identified from the remaining feature map R . According to Yi *et al.*'s [37] research, the Otsu methods do not perform well in dealing with blurring edges, and usually miss some edge pixels. The 2-D maximum entropy threshold (MET) method [38] has a better effect on processing blurring edges. The 2-D MET method applies the theory of information entropy to obtain the optimal threshold, which maximizes the sum of the

information entropy of the target and background. The 2-D MET method considers both the gray distribution information of pixels and the spatial correlation information among pixels. The low threshold T_l is calculated by

$$T_l = \text{MET}(R) \quad (23)$$

where MET means the 2-D MET method.

4) *Hysteresis Thresholding*: To obtain the final edge map E , the thresholds T_h and T_l are used to process the multiscale merged edge map M by hysteresis thresholding as follows. If $M(i, j) > T_h$, $E(i, j)$ is recognized as an edge pixel and its value is set as $E(i, j) = 255$. If $M(i, j) < T_l$, $E(i, j)$ is recognized as a nonedge pixel and its value is set as $E(i, j) = 0$. If $T_l < M(i, j) < T_h$, we further observe whether there are edge pixels in its local neighborhood. If there are edge pixels in its neighborhood, $E(i, j)$ is recognized as an edge pixel and its value is set as $E(i, j) = 255$. Otherwise, $E(i, j)$ is recognized as a nonedge pixel and its value is set as $E(i, j) = 0$. As a result, we obtain the final edge map E .

IV. EXPERIMENTS AND PERFORMANCE EVALUATION

To show the performance of our method, we conduct the experiments in terms of the accuracy and adaptivity of edge detection. Our method is compared with Canny [10], Han *et al.* [33], Morphology [35], Edge Drawing [27], Rong *et al.* [34], and superpixel-based edge detector (SBED) [26]. These six methods are very typical and representative for edge detection. The test images are selected from the actually collected images and the BSDS500 dataset [39]. We select 40 images from the BSDS500 dataset, as shown in Fig. 8. The rule for selecting the test images is that different types and sizes of objects, different complexity levels of the background, and different light conditions should be included as far as possible. In addition, some test images should contain faint edges. These rules can reflect the real performance of the edge detectors comprehensively.

A. Performance Metrics

We choose Pratt's figure of merit (FOM) and F-Measure as the performance metrics to evaluate the edge detectors by judging the similarity between the detected edge maps and the ground truth.

1) *FOM*: FOM [40] is a classical evaluation metric and is widely used to evaluate many state-of-the-art edge detectors [23], [41], [42]. It considers the loss of true edge pixels, the localization errors, and spurious edge pixels. FOM is calculated by

$$\text{FOM} = (1/\max(N_I, N_D)) \sum_{k=1}^{N_D} (1/(1 + \gamma d_k^2)) \quad (24)$$

where N_I is the number of the edge pixels in the ground truth, N_D is the number of the edge pixels in the detected edge map, d_k is the distance between the k th detected edge pixel and its corresponding edge pixel in the ground truth, and $\gamma = 1/9$ is a scaling constant as described in [40]. FOM varies from 0 to 1, where 1 represents a perfect match between the detected edge map and the ground truth.

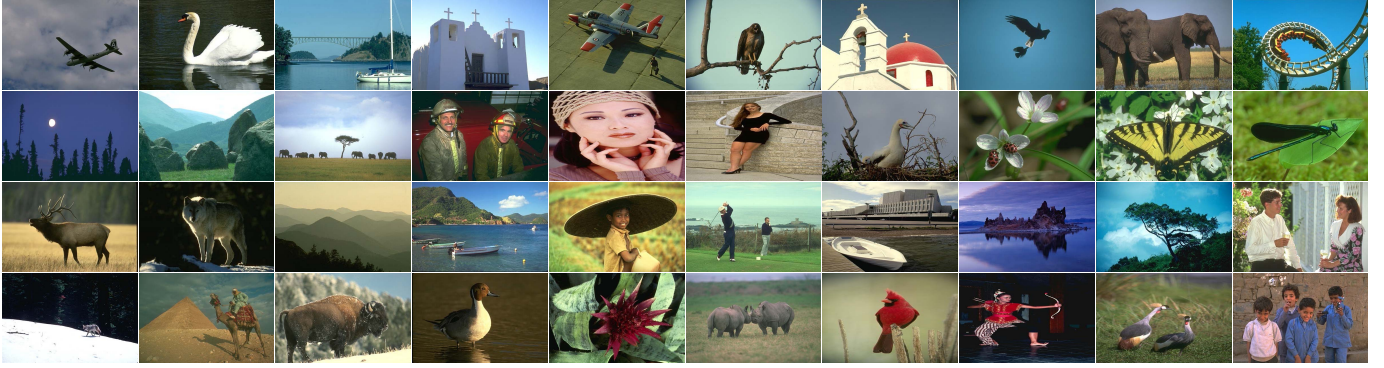


Fig. 8. Forty test images from the BSDS500 dataset.

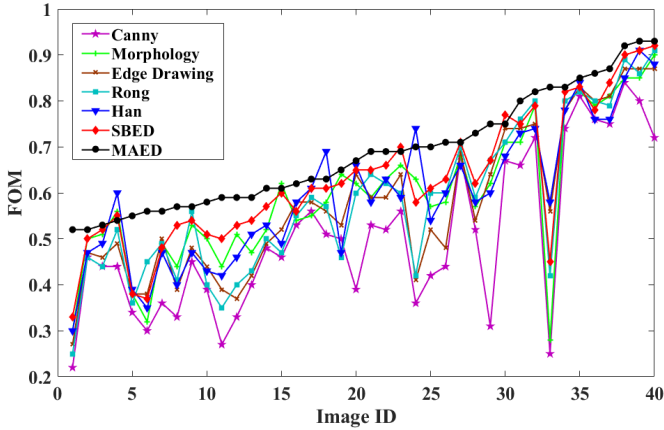


Fig. 9. FOM results of 40 images in the accuracy experiments.

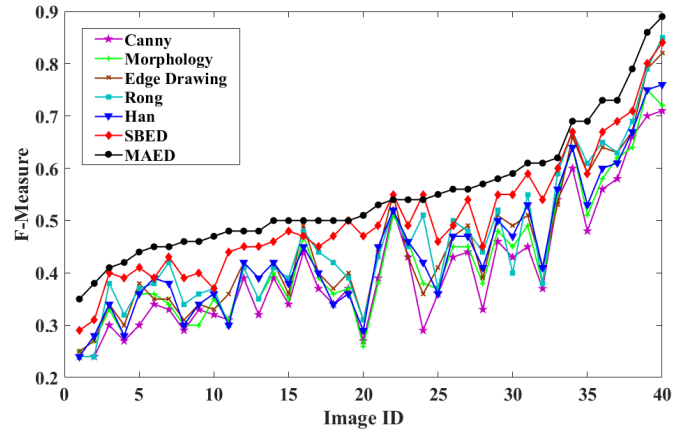


Fig. 10. F-Measure results of 40 images in the accuracy experiments.

2) *F-Measure*: *F-Measure* is the harmonic mean of the precision and recall. It is widely used to evaluate the performance of edge detectors [43], [44]. For each edge pixel in the detected edge map, if it matches the edge pixel in the ground truth, it will be marked as a true positive pixel. If not match, it will be marked as a false positive pixel. For each edge pixel in the ground truth, if it does not match the edge pixel in the detected edge map, it will be marked as a false negative pixel. The precision P and recall R are calculated by

$$P = TP / (TP + FP) \quad (25)$$

$$R = TP / (TP + FN) \quad (26)$$

where TP is the number of the true positive pixels, FP is the number of the false positive pixels, and FN is the number of the false negative pixels. *F-Measure* is calculated by

$$F_{\text{Measure}} = 2PR / (P + R). \quad (27)$$

FOM and *F-Measure* can effectively evaluate the accuracy of edge detection. *FOM* reflects the detection rate of edge pixels, the suppression effect of spurious edges, and the localization accuracy of edge pixels. *F-Measure* reflects the detection rate of the effective edge pixels and the suppression effect of spurious edges.

B. Accuracy Experiments

The optimal parameters are set for each image processed by each edge detector. Figs. 9 and 10 show the *FOM* and *F-Measure* results of 40 images with seven edge detectors, respectively. As can be observed, the performance of our method for each image is always close to the best among seven edge detectors. The performance of Canny is not very good, because it is difficult to effectively distinguish faint edges and noise pixels based on the gradient map, and the setting of parameters is easily be interfered by human factors. The performance of Morphology, Edge Drawing, Han, Rong, and SBED is better than that of Canny. The accuracy of SBED is the closest to that of our method. However, the edges identified by SBED are wider and the localization is not accurate enough. Besides, we can find that the images with good metrics usually contain less faint edges and noise, while the images with bad metrics usually contain more faint edges and noise. Even for the images which are difficult to detect the edges, our method can still achieve better edge detection performance in most cases.

Table I gives the means of *FOM* and *F-Measure* for each edge detector in the accuracy experiments. As can be observed, our method is higher than the other edge detectors for both the means of *FOM* and *F-Measure*. According to the meaning of *FOM* and *F-Measure*, the edge maps obtained by our method

TABLE I
MEANS OF FOM AND F-MEASURE IN THE ACCURACY EXPERIMENTS (%)

	Canny	Morphology	Edge Drawing	Rong	Han	SBED	MAED
FOM	50.6	59.7	57.7	58.2	59.6	62.7	68.6
F-Measure	40.6	42.5	44.8	45.9	44.0	50.6	55.1

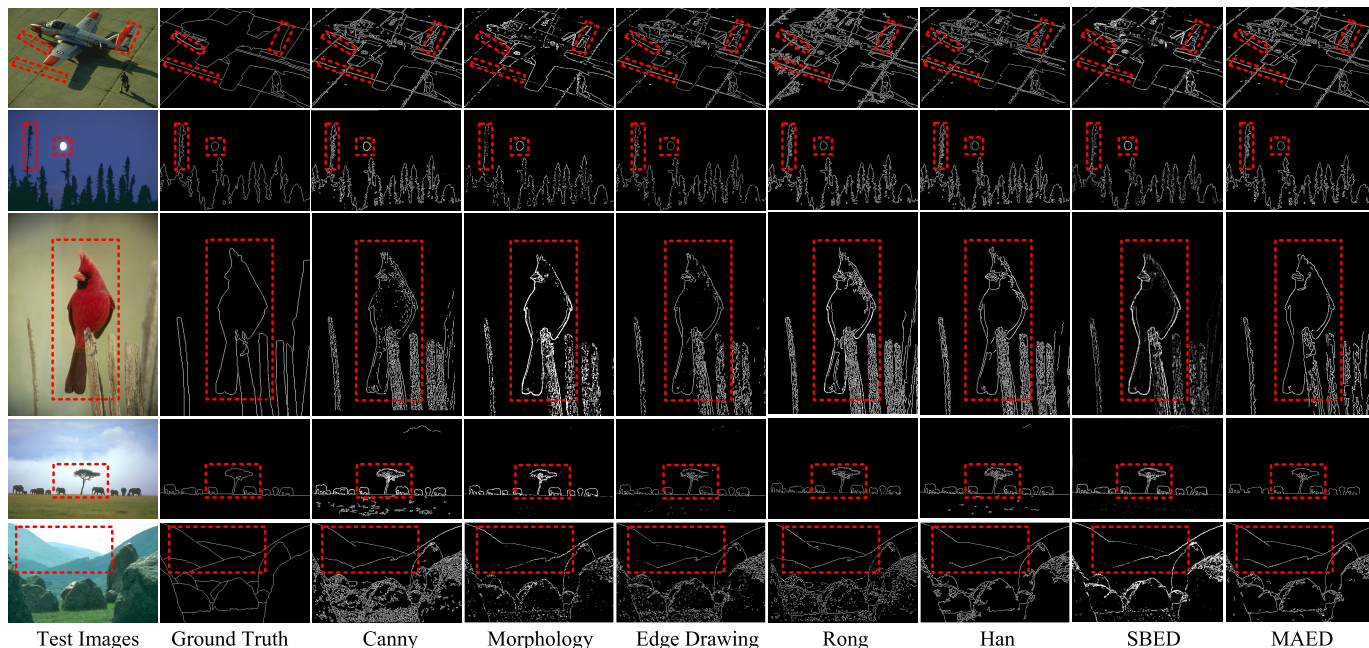


Fig. 11. Edge detection results of five typical test images with the seven edge detectors in the accuracy experiments.

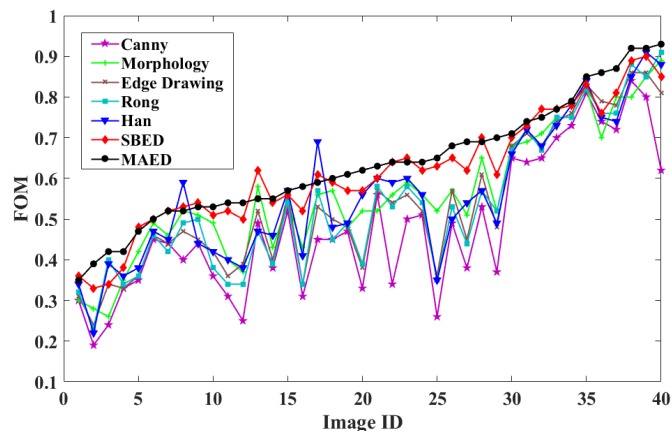


Fig. 12. FOM results of 40 images in the adaptivity experiments.

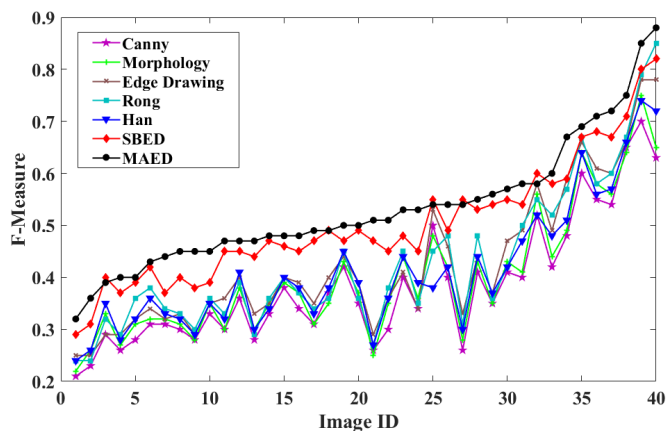


Fig. 13. F-Measure results of 40 images in the adaptivity experiments.

have more effective edge pixels, higher localization accuracy, and fewer spurious edge pixels. This shows that our method has the higher accuracy of edge detection.

Fig. 11 shows the edge detection results of five typical test images with seven edge detectors. The first column is the images selected from BSDS500. These images have different types of objects with different light and shade, weak edges, and complex backgrounds. They can be used to illustrate the performance of each edge detector. The second column is the ground truth. The third–ninth columns are the results

of Canny, Morphology, Edge Drawing, Han, Rong, SBED, and our method. As can be observed, Canny has more noise, which reduces the accuracy. Morphology, Edge Drawing, and Han lose some faint edges, resulting in lower edge detection accuracy. For most images, the edge maps obtained by Rong have good continuity and less loss of faint edges. However, they contain many spurious edges. The performance of SBED is the closest to our method. However, the edges detected by SBED are wider and some weak edges are lost. Compared with other edge detectors, our method achieves the best tradeoff

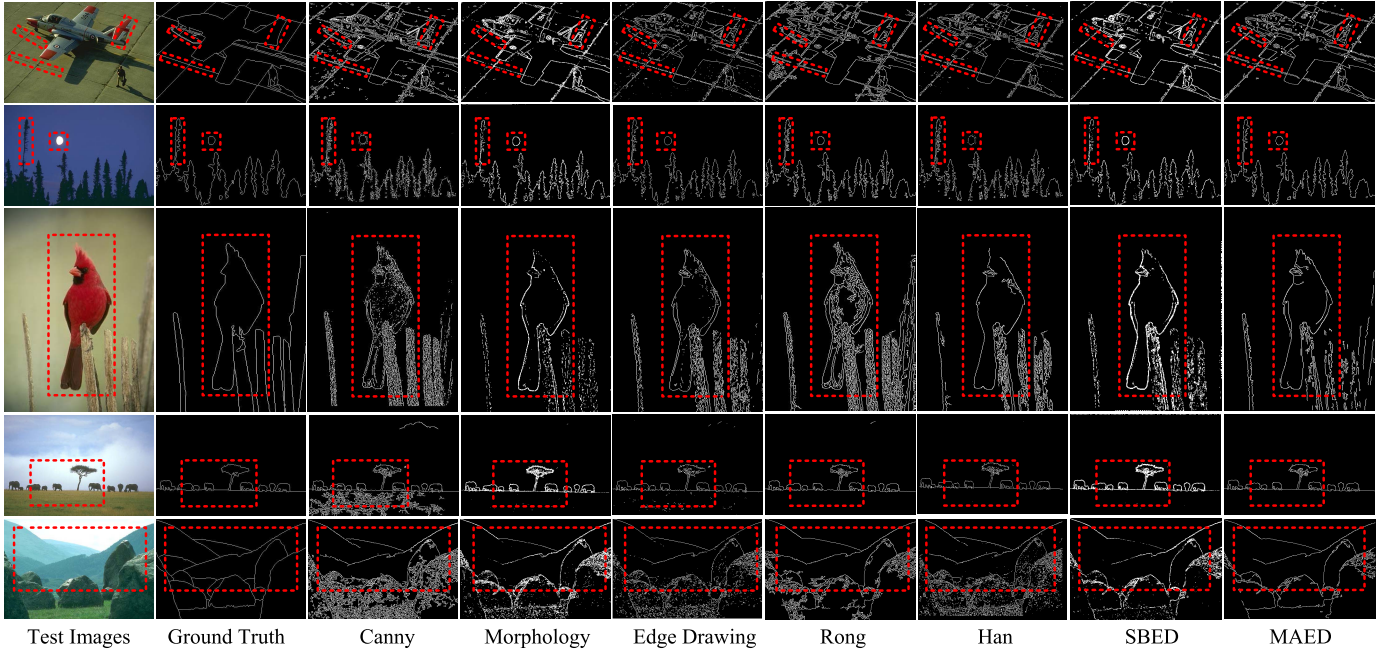


Fig. 14. Edge detection results of five typical test images with the seven edge detectors in the adaptivity experiments.

TABLE II
FIXED PARAMETERS FOR THE EDGE DETECTORS

Algorithms	Parameters
Canny [7]	W -size=5x5, $\sigma=1.2$, T -high=125, T -low=60
Morphology [18]	W -size=3x3, $\alpha = 0.3$, $\beta = 0.7$
Edge Drawing [28]	W -size=3x3, $\sigma=1.2$, $T_{grad} = 20$, $T_{anch} = 6$, $i=2$
Han [34]	W -size=5x5, $\sigma=1.2$
Rong [35]	W -size=5x5, $\sigma=1.2$
SBED [27]	$NSP=1500$, $CF=10$, G -low=0.1, G -high=0.8
MAED	W -size=3x3, $k=0.8$

between reducing the loss of faint edges and preventing the generation of spurious edges. The proposed method is the closest to the ground truth and has the higher accuracy.

As marked in the images of the first row of Fig. 11, our method can detect the edges of the aircraft tail, tow bar, and the stop line completely, and generate less spurious edges. Canny, Morphology, Edge Drawing, Han, and SBED lose some faint edges. Rong can detect most of the edges, but it generates too many spurious edges. As marked in the fifth row, only our method can detect the edges of mountains more completely and clearly with fewer spurious edges.

C. Adaptivity Experiments

The adaptivity of edge detectors is mainly reflected in two aspects: one is the number of parameters to be set artificially, and the other is the performance of edge detectors for different types of images in a set of fixed parameters. The fewer the parameters that need to be set artificially, the higher the adaptivity. The better the edge detection effect for different types of images, the higher the adaptivity in the case of the fixed parameters.

In the adaptivity experiments, we also use the 40 images in Fig. 8 as test images for quantitative analysis. We choose a set of fixed parameters for each edge detector to evaluate its adaptivity for different images. The parameters which are most commonly used in the accuracy experiments are chosen as the fixed parameters, as presented in Table II. As can be observed, the number of the parameters that our method needs to set is relatively small in all the seven edge detectors.

Figs. 12 and 13 show the FOM and F-Measure results of 40 images with seven edge detectors in the case of the fixed parameters. As can be observed, the performance of our method for all the test images is always close to the best among seven edge detectors. The performance of Canny is not very good, since it needs to set some parameters artificially in the Gaussian filtering and hysteresis linking. It can only achieve good results on a part of images, and the overall performance needs to be improved in the case of the fixed parameters. The adaptivity of Morphology, Edge Drawing, Han, Rong, and SBED is better than that of Canny. The performance of SBED is the closest to our method. However, the number of parameters that need to be set artificially in this method is still more than our method. And, the edges identified by SBED are wider, which affect the accuracy of localization.

Table III gives the means of FOM and F-Measure for each edge detector in the adaptivity experiments. As can be observed, our method is higher than the other edge detectors for both the means of FOM and F-Measure. According to the meaning of FOM and F-Measure, our method has a better performance in the case of the fixed parameters. This demonstrates that our method has the higher adaptivity of edge detection.

Fig. 14 shows the edge detection results of five typical test images with seven edge detectors in the case of the fixed parameters. The first column is the images selected from

TABLE III
MEANS OF FOM AND F-MEASURE IN THE ADAPTIVITY EXPERIMENTS (%)

	Canny	Morphology	Edge Drawing	Rong	Han	SBED	MAED
FOM	48.3	56.0	53.6	53.5	55.8	61.5	63.7
F-Measure	38.4	40.1	42.7	42.7	41.2	50.2	53.3

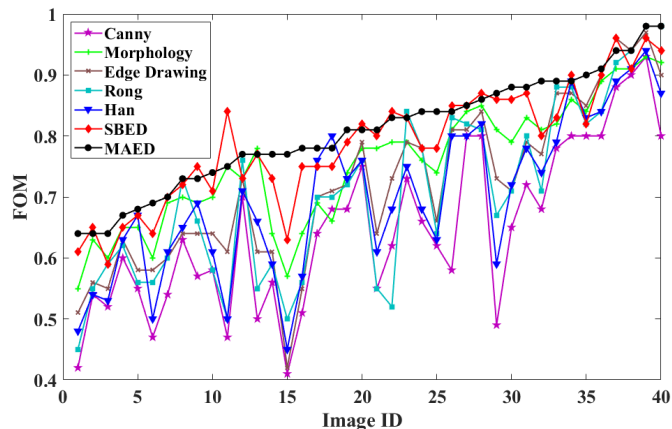


Fig. 15. FOM results of 40 images in the resolution experiments.

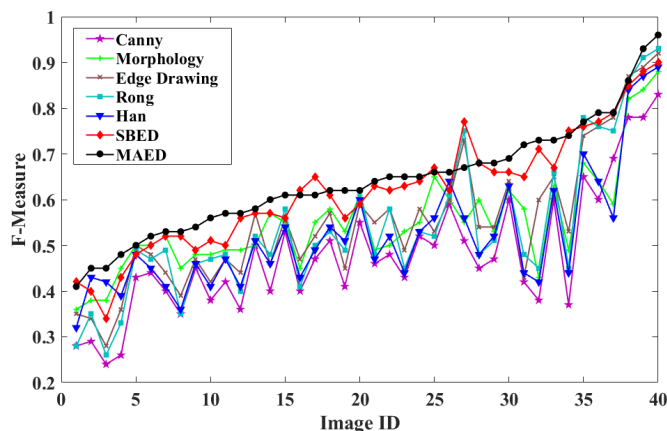


Fig. 16. F-Measure results of 40 images in the resolution experiments.

BSDS500. These images include different types of objects, different light conditions, complex backgrounds, and faint edges. They can be used to fully illustrate the performance of each edge detector. The second column is the ground truth. The third–ninth columns are the results of edge detection with Canny, Morphology, Edge Drawing, Han, Rong, SBED, and our method. As can be observed, Canny and Rong contain more spurious edges. Morphology, Edge Drawing, Han, and SBED lose some edge pixels in the faint edges, resulting in poor edge continuity and lower edge detection accuracy. Compared with the other edge detectors, our method achieves the best tradeoff between reducing the loss of faint edges and preventing the generation of spurious edges. Our method can still achieve better edge detection effect, which proves that our method has higher adaptivity in the case of the fixed parameters.

As marked in the second row of Fig. 14, only our method can detect the moon and the treetops in the dark more

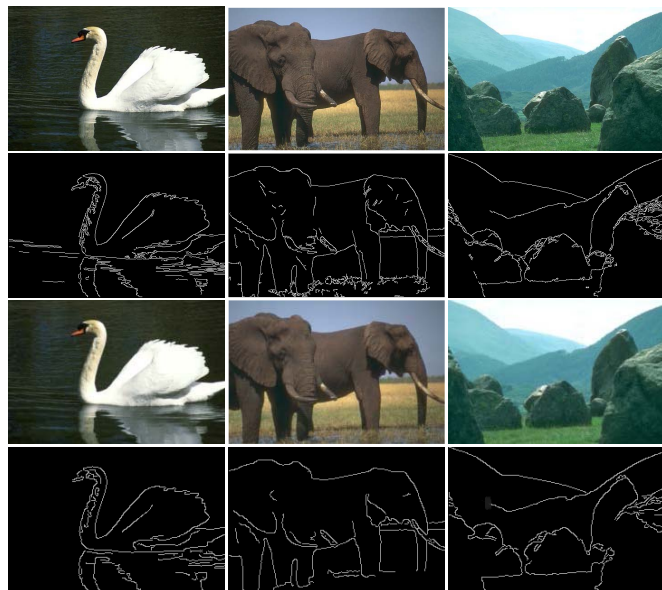


Fig. 17. Edge detection results of three images at different resolutions with our method in the case of the fixed parameters.

completely and clearly with fewer spurious edges. As marked in the third row, our method can detect the edges of the bird completely and generate less spurious edges. Morphology, Han, and SBED lose some faint edges. Canny, Edge Drawing, and Rong can detect most of the edges, but it generates too many spurious edges.

In summary, our method needs to set fewer parameters, and has better edge detection effect for all the test images in the case of the fixed parameters. Therefore, it has the higher adaptivity.

D. Resolution Experiments

In this experiment, we select 40 test images as shown in Fig. 8 and reduce their resolution. Figs. 15 and 16 show the FOM and F-Measure results in the case of the fixed parameters at the reduced resolution. As can be observed, the performance of our method for each image is always close to the best among seven edge detectors. Table IV gives the means of FOM and F-Measure for each edge detector in the resolution experiments. As can be observed, our method is higher than the other edge detectors for both the means of FOM and F-Measure.

Compared with the results shown in Figs. 12 and 13, it is proved that our method can achieve good edge detection effects at different resolutions. The main reason is that our method effectively utilizes the advantages of multiscale features. This makes the fused edge map robust to noise and have good edge continuity. Further, it demonstrates that our method has the higher adaptivity.

TABLE IV
MEANS OF FOM AND F-MEASURE IN THE RESOLUTION EXPERIMENTS (%)

	Canny	Morphology	Edge Drawing	Rong	Han	SBED	MAED
FOM	64.8	75.3	71.8	70.4	70.0	78.7	80.8
F-Measure	48.0	55.1	55.2	53.8	52.2	61.8	64.1

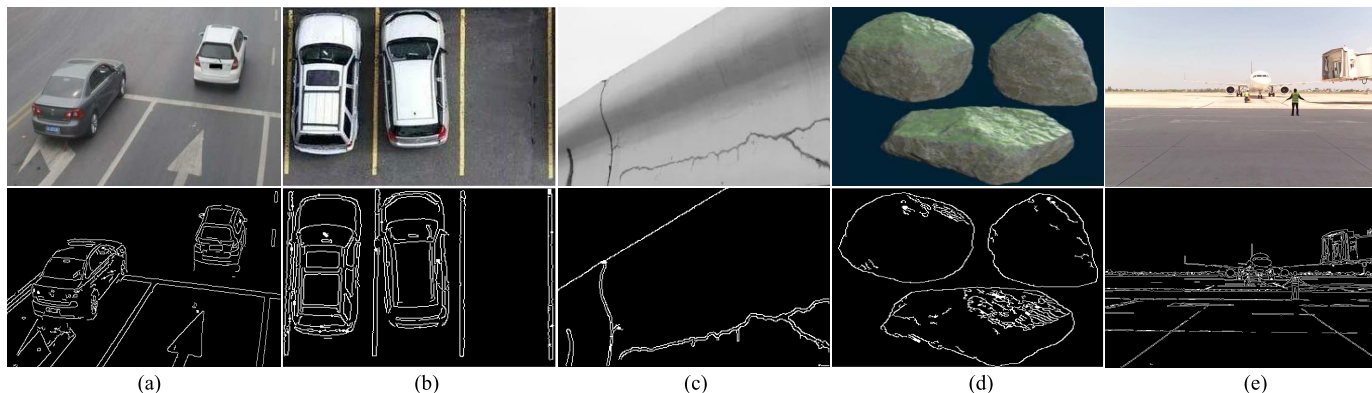


Fig. 18. Five typical measurement applications of edge detection with our method. (a) Automatic traffic monitoring. (b) Automatic parking monitoring. (c) Bridge crack detecting. (d) Ore fragmentation computing. (e) Aircraft parking monitoring.

Fig. 17 shows the edge detection results of three images at different resolutions with our method in the case of the fixed parameters. The first row shows the test images at the resolution 481×321 , whose edge maps are shown in the second row. The third row shows the test images at the reduced resolution 241×161 , whose edge maps are shown in the fourth row. As can be observed, the edge maps detected from the images at the high resolution contain more detailed features, but are susceptible to noise. The edge maps detected from the images at the low resolution are less affected by noise, but lose some details. This shows the influence of different image resolutions on edge detection.

In summary, because our method can make better use of the multiscale characteristics of the images, so it has better edge detection effect for the test images at different resolutions.

E. Application Experiments

This experiment is performed to show the contributions of the proposed method in measurement applications. As shown in Fig. 18, in the field of automatic traffic monitoring, our method can accurately detect traffic indicator lines and vehicle contours, which helps to monitor the traffic situation and effectively judge whether there are vehicles violating traffic regulations. In the field of automatic parking monitoring, our method can accurately detect vehicle contours and parking lines, which is beneficial to monitor the parking condition. In the field of bridge crack detecting, our method can effectively detect the cracks on the bridge, in order to provide support for the daily maintenance of the bridge. In the field of ore fragmentation computing, our method can accurately detect the edges of ores, which is helpful to the automatic calculation of the ore sizes. In the field of aircraft parking monitoring, the aircraft contours and stop lines can be effectively extracted by our method, so as

to monitor the aircraft parking status and provides support for aircraft automatic parking. In summary, our method plays an important role in many measurement applications.

F. Discussion

From the experimental results, we can see that our method has high accuracy and good adaptivity. The reasons that our method can achieve good performance are mainly due to the following aspects.

- 1) The standard deviation map contains the statistical characteristics of local gradient differences, which is helpful to distinguish edges from noise and beneficial to extract edges more accurately.
- 2) The pixel-by-pixel detection makes full use of different features in the gradient map and the standard deviation map, which helps to accurately extract edges from the image.
- 3) Multiscale edge merging effectively utilizes the advantages of multiscale features based on the novel voting mechanism, which makes the fused edge map robust to noise and have good edge continuity.
- 4) Adaptive hysteresis linking automatically obtains dual thresholds based on the 2-D Otsu and 2-D maximum entropy, which helps to improve the adaptivity effectively.

V. CONCLUSION

In this article, we propose a novel unsupervised MAED. The main contributions of our edge detector include construction of the standard deviation map, pixel-by-pixel detection, multiscale edge merging, and adaptive hysteresis linking. The standard deviation map, which is proposed for the first time, can effectively distinguish the edges from noise based on the statistical characteristics of the local gradient differences.

Pixel-by-pixel detection can accurately extract the edges from the image based on the gradient map and the standard deviation map. Multiscale edge merging utilizes the advantages of multiscale features based on the voting mechanism, which makes the fused edge map robust to noise and have good edge continuity. Adaptive hysteresis linking sets the high and low thresholds automatically, which is helpful to improve the adaptivity. These contributions make our method have high accuracy and good adaptivity. Both the actually collected images and benchmark dataset are deployed to show the performance of our method. Experiments show that our method outperforms state-of-the-art edge detectors in terms of the accuracy and adaptivity, which is important to measurement applications.

Although our method has achieved good performance in edge detection, it cannot be directly applied to a color image, and needs to transform the color image into a gray image first. Color information is helpful to edge detection. We will make the further research in the future work.

REFERENCES

- [1] B. Gardiner, S. A. Coleman, and B. W. Scotney, "Multiscale edge detection using a finite element framework for hexagonal pixel-based images," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1849–1861, Apr. 2016.
- [2] T. Qiu, Y. Yan, and G. Lu, "An autoadaptive edge-detection algorithm for flame and fire image processing," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 5, pp. 1486–1493, May 2012.
- [3] X. Gong, H. Su, D. Xu, J. Zhang, L. Zhang, and Z. Zhang, "Visual defect inspection for deep-aperture components with coarse-to-fine contour extraction," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 6, pp. 3262–3274, Jun. 2020.
- [4] Y. Gan and Q. Zhao, "An effective defect inspection method for LCD using active contour model," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 9, pp. 2438–2445, Sep. 2013.
- [5] D.-M. Tsai and Y.-C. Hsieh, "Machine vision-based positioning and inspection using expectation-maximization technique," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 11, pp. 2858–2868, Nov. 2017.
- [6] T. Jiang, H. Cui, X. Cheng, and W. Tian, "A measurement method for robot peg-in-hole prealignment based on combined two-level visual sensors," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.
- [7] Z. Wang, G. Cheng, and J. Zheng, "Road edge detection in all weather and illumination via driving video mining," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 2, pp. 232–243, Jun. 2019.
- [8] S. Xie and Z. Tu, "Holistically-nested edge detection," *Int. J. Comput. Vis.*, vol. 125, nos. 1–3, pp. 3–18, Dec. 2017.
- [9] Z. Yu, C. Feng, M.-Y. Liu, and S. Ramalingam, "CASENet: Deep category-aware semantic edge detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1761–1770.
- [10] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [11] R. J. Qian and T. S. Huang, "Optimal edge detection in two-dimensional images," *IEEE Trans. Image Process.*, vol. 5, no. 7, pp. 1215–1220, Jul. 1996.
- [12] S. K. G. Shirazi and R. Safabakhsh, "Omnidirectional edge detection," *Comput. Vis. Image Understand.*, vol. 113, no. 4, pp. 556–564, Apr. 2009.
- [13] J. Cao, L. Chen, M. Wang, and Y. Tian, "Implementing a parallel image edge detection algorithm based on the otsu-canny operator on the Hadoop platform," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–12, May 2018.
- [14] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes-active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1987.
- [15] P. Brigger, J. Hoeg, and M. Unser, "B-spline snakes: A flexible tool for parametric contour detection," *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1484–1496, Sep. 2000.
- [16] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu, "Statistical edge detection: Learning and evaluating edge cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 1, pp. 57–74, Jan. 2003.
- [17] R. R. Rakesh, P. Chaudhuri, and C. A. Murthy, "Thresholding in edge detection: A statistical approach," *IEEE Trans. Image Process.*, vol. 13, no. 7, pp. 927–936, Jul. 2004.
- [18] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.
- [19] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 1998, pp. 839–846.
- [20] M. Mafi, H. Rajaei, M. Cabrerizo, and M. Adjouadi, "A robust edge detection approach in the presence of high impulse noise intensity through switching adaptive median and fixed weighted mean filtering," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5475–5490, Nov. 2018.
- [21] T. Qiu, A. Wang, N. Yu, and A. Song, "LLSURE: Local linear SURE-based edge-preserving image filtering," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 80–90, Jan. 2013.
- [22] J. Jain and V. Tyagi, "An adaptive edge-preserving image denoising technique using patch-based weighted-SVD filtering in wavelet domain," *Multimedia Tools Appl.*, vol. 76, no. 2, pp. 1659–1679, Jan. 2017.
- [23] W. Zhang, Y. Zhao, T. P. Breckon, and L. Chen, "Noise robust image edge detection based upon the automatic anisotropic Gaussian kernels," *Pattern Recognit.*, vol. 63, pp. 193–205, Mar. 2017.
- [24] E. Galvanin, G. Vale, and A. Poz, "The canny detector with edge region focusing using an anisotropic diffusion process," *Pattern Recognit. Image Anal.*, vol. 16, no. 4, pp. 614–621, 2006.
- [25] J. Geusebroek, A. W. M. Smeulders, and J. van de Weijer, "Fast anisotropic gauss filtering," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 938–943, Aug. 2003.
- [26] H. Zhang, C. Wu, L. Zhang, and H. Zheng, "A novel centroid update approach for clustering-based superpixel methods and superpixel-based edge detection," 2019, *arXiv:1910.08439*. [Online]. Available: <http://arxiv.org/abs/1910.08439>
- [27] C. Topal and C. Akinlar, "Edge drawing: A combined real-time edge and segment detector," *J. Vis. Commun. Image Represent.*, vol. 23, no. 6, pp. 862–872, Aug. 2012.
- [28] J. H. Elder and S. W. Zucker, "Scale space localization, blur, and contour-based image coding," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1996, pp. 27–34.
- [29] N. Wang, Y. Chen, L. Yao, Q. Zhang, L. Jia, and Z. Gui, "Image smoothing via adaptive fourth-order partial differential equation model," *J. Eng.*, vol. 2019, no. 11, pp. 8198–8206, Nov. 2019.
- [30] Y. Yitzhaky and E. Peli, "A method for objective edge detection evaluation and detector parameter selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 1027–1033, Aug. 2003.
- [31] R. Medina-Carnicer, F. J. Madrid-Cuevas, A. Carmona-Poyato, and R. Muñoz-Salinas, "On candidates selection for hysteresis thresholds in edge detection," *Pattern Recognit.*, vol. 42, no. 7, pp. 1284–1296, Jul. 2009.
- [32] R. Medina-Carnicer, F. J. Madrid-Cuevas, R. Muñoz-Salinas, and A. Carmona-Poyato, "Solving the process of hysteresis without determining the optimal thresholds," *Pattern Recognit.*, vol. 43, no. 4, pp. 1224–1232, Apr. 2010.
- [33] D. D. Han, T. C. Zhang, and J. Zhang, "Research and implementation of an improved canny edge detection algorithm," *Key Eng. Mater.*, vol. 572, pp. 566–569, Sep. 2013.
- [34] L. Xuan and Z. Hong, "An improved canny edge detection algorithm," in *Proc. 8th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Nov. 2017, pp. 577–582.
- [35] W. Shi, X. Zhu, and G. Zhu, "Adaptive edge detection algorithm of MRI image based on morphology," *Chin. J. Sci. Instrum.*, vol. 34, no. 2, pp. 408–414, 2013.
- [36] L. Jianzhuang, L. Wenqing, and T. Yupeng, "Automatic thresholding of gray-level pictures using two-dimension otsu method," in *Proc. Int. Conf. Circuits Syst.*, 1991, pp. 325–327.
- [37] S. Yi, G. Zhang, J. He, and S. Li, "Maximum entropy image segmentation based on maximum interclass variance," *Comput. Eng. Sci.*, vol. 40, no. 10, pp. 1874–1881, Oct. 2018.
- [38] A. Abutaleb, "Automatic thresholding of gray-level pictures using two-dimensional entropy," *Comput. Vis. Graph. Image Proc.*, vol. 47, pp. 22–32, Jul. 1989.
- [39] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
- [40] I. E. Abdou and W. K. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," *Proc. IEEE*, vol. 67, no. 5, pp. 753–763, May 1979.

- [41] G. Sun, Q. Liu, Q. Liu, C. Ji, and X. Li, "A novel approach for edge detection based on the theory of universal gravity," *Pattern Recognit.*, vol. 40, no. 10, pp. 2766–2775, Oct. 2007.
- [42] O. P. Verma and A. S. Parihar, "An optimal fuzzy system for edge detection in color images using bacterial foraging algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 1, pp. 114–127, Feb. 2017.
- [43] N. Ofir, M. Galun, S. Alpert, A. Brandt, B. Nadler, and R. Basri, "On detection of faint edges in noisy images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 894–908, Apr. 2020.
- [44] K. Özkan and S. Isik, "A novel multi-scale and multi-expert edge detector based on common vector approach," *AEU-Int. J. Electron. Commun.*, vol. 69, no. 9, pp. 1272–1281, Sep. 2015.



Yunhao Cui received the M.S. degree in mechanical design and theory from Northeastern University, Shenyang, China, in 2014. He is currently pursuing the Ph.D. degree with the School of Mechanical Engineering, Dalian University of Technology, Dalian, China.

His research interests include image processing, cloud data processing, intelligent mechanical equipment, and 3-D environmental perception.



Yi An (Member, IEEE) received the B.S. degree in automation and the M.S. and Ph.D. degrees in control theory and control engineering from the Dalian University of Technology, Dalian, China, in 2001, 2004, and 2011, respectively.

From 2007 to 2011, he was a Lecturer with the School of Control Science and Engineering, Dalian University of Technology, where he has been an Associate Professor since 2012. His research interests include pixel cloud data processing, sensing and perception, information fusion, robot vision, and intelligent robot.



Wei Sun received the B.S. degree in mechanical engineering and the M.S. and Ph.D. degrees from the Dalian University of Technology, Dalian, China, in 1988, 1993, and 2000, respectively.

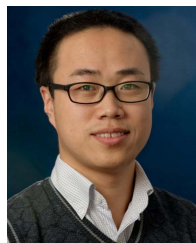
He is currently a Professor with the School of Mechanical Engineering, Dalian University of Technology. His current research interests include intelligent mechanical equipment, 3-D environmental perception, and the design and optimization of complex equipment.



Huosheng Hu (Senior Member, IEEE) received the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K., in 1993.

He is currently a Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., where he is leading the Robotics Research Group. He has authored over 420 articles. His current research interests include robotics, human–robot interaction, embedded systems, mechatronics, and pervasive computing.

Prof. Hu is a Founding Member of the IEEE Robotics and Automation Society Technical Committee on Networked Robots, a fellow of the Institution of Engineering and Technology, and a Senior Member of the Association for Computing Machinery. He currently serves as the Editor-in-Chief for the *International Journal of Automation and Computing* and the online *Robotics Journal*, and an Executive Editor for the *International Journal of Mechatronics and Automation*.



Xueguan Song received the B.S. degree in mechanical engineering from the Dalian University of Technology, Dalian, China, in 2004, and the M.S. and Ph.D. degrees in mechanical engineering from Dong-A University, Busan, South Korea, in 2007 and 2010, respectively.

He is currently a Professor with the School of Mechanical Engineering, Dalian University of Technology. His current research interests include intelligent mechanical equipment, pixel cloud data processing, and multidisciplinary design optimization.