



# Machine-learning models for analyzing TSOM images of nanostructures

YUFU QU,<sup>1,\*</sup> JIALIN HAO,<sup>1,2</sup> AND RENJU PENG<sup>1,3</sup>

<sup>1</sup>School of Instrumentation Science and Opto-Electronics Engineering, Beihang University, Beijing 100191, China

<sup>2</sup>[hjl\\_ace@buaa.edu.cn](mailto:hjl_ace@buaa.edu.cn)

<sup>3</sup>[rjpeng@buaa.edu.cn](mailto:rjpeng@buaa.edu.cn)

\* [qyf@buaa.edu.cn](mailto:qyf@buaa.edu.cn)

**Abstract:** Through-focus scanning optical microscopy (TSOM) is an economical and nondestructive method for measuring three-dimensional nanostructures. After obtaining a TSOM image, a library-matching method is typically used to interpret optical intensity information and determine the dimensions of a measurement target. To further improve dimensional measurement accuracy, this paper proposes a machine learning method that extracts texture information from TSOM images. The method extracts feature vectors of TSOM images in terms of the Gray-level Co-occurrence Matrix (GLCM), Local Binary Pattern (LBP), and Histogram of Oriented Gradient (HOG). We tested models trained with these vectors in isolation, in pairs, and a combination of all three to test seven possible feature vectors. Once normalized, these feature vectors were then used to train and test three machine-learning regression models: random forest, GBDT, and AdaBoost. Compared with the results of the library-matching method, the measurement accuracy of the machine learning method is considerably higher. When detecting dimensional features that fall into a wide range of sizes, the AdaBoost model used with the combined LBP and HOG feature vectors performs better than the others. For detecting dimensional features within a narrower range of sizes, the AdaBoost model combined with HOG feature extraction algorithm performs better.

© 2019 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

## 1. Introduction

The semiconductor industry and nanotechnology research require rapid, accurate, and nondestructive three-dimensional measurements of nanoscale shapes. Measurement methods based on electronic excitation scattering such as scanning electron microscopy [1,2] and probe-based measurement methods such as atomic force microscopy [3] are commonly used today. These measurement methods are accurate, but they are inefficient and costly, and they cannot be used for online real-time detection. Through-focus scanning optical microscopy (TSOM) is a novel fast and nondestructive optical measurement method [4–8]. TSOM captures a single in-focus image and scans along the optical axis to acquire an image sequence consisting of an in-focus image and a series of defocused images of the imaging target. A TSOM image is constructed from the image sequence, and it is then matched with the image in a simulated TSOM image library. Finally the parameter to be detected is recorded from the best match [9,10]. Therefore, TSOM avoids the limit of Rayleigh diffraction that constrains traditional optical measurements, and has been applied in mask-defect detection [11], nanoscale dimensional measurements of three-dimensional structures [12], nanoparticle shape evaluation [13] and overlay analysis [14,15].

At present, nanostructure defects and critical dimensions are usually measured from TSOM images using the library-matching method [9,10,13]. This method generally first fits a linear regression model of the key parameters of the target TSOM image and nanostructures, and then uses the linear regression model to predict the size of the target. The library-matching method uses the optical intensity range (OIR) [16] and the mean-square intensity (MSI) [10] of the

TSOM image to fit the model. OIR is defined as the absolute difference between the maximum and minimum optical-intensity values in the normalized TSOM image:

$$\text{OIR} = \max(I) - \min(I) \quad (1)$$

where  $I$  is the normalized TSOM image. The MSI is defined as the average of the sum of the squares of optical intensity values of each pixel in the normalized TSOM image:

$$\text{MSI} = \frac{1}{n} \sum_{i=1}^n I_i^2 \quad (2)$$

where  $I_i$  is the intensity value of the  $i$ -th pixel in the normalized TSOM image, and  $n$  is the number of pixels with intensity  $I$ . The library-matching method based on OIR and MSI considers only the intensity information in the TSOM image, so it requires powerful processing hardware and a controlled measurement environment [17,18]. In addition, the measurable range of the method is limited [10], which limits the practical applications of TSOM.

The texture of a TSOM image is strongly correlated with the size of the measured target. Therefore, this paper proposes a machine-learning image analysis method that uses based texture information in TSOM images. First, the method extracts feature vectors of the TSOM images in terms of the gray-level co-occurrence matrix (GLCM), local binary pattern (LBP), and histogram of oriented gradient (HOG). To determine the feature vectors that offer the highest measurement accuracy, this study combined the three feature vectors in pairs and in combinations, to yield seven feature data sets. Secondly, after normalizing these feature vectors in feature data sets, three machine-learning regression models, random forest, gradient boosting decision tree (GBDT) and AdaBoost, were trained and tested for how accurately they distinguish the dimensions of a target structure. The three machine learning methods were used to obtain the critical dimensions of the target structure. Finally, the regression models were analyzed and evaluated to determine which yields the most-accurate size predictions.

The structure of the rest of this paper is as follows: the second part introduces the process of building the machine learning models and using them for regressions, and discusses the experimental device, the feature-vector extraction algorithms, and the machine-learning algorithms. The third part discusses our experimental methods, including descriptions of the data sets, model-evaluation indices, and analysis of the experimental results. The fourth part discusses our results.

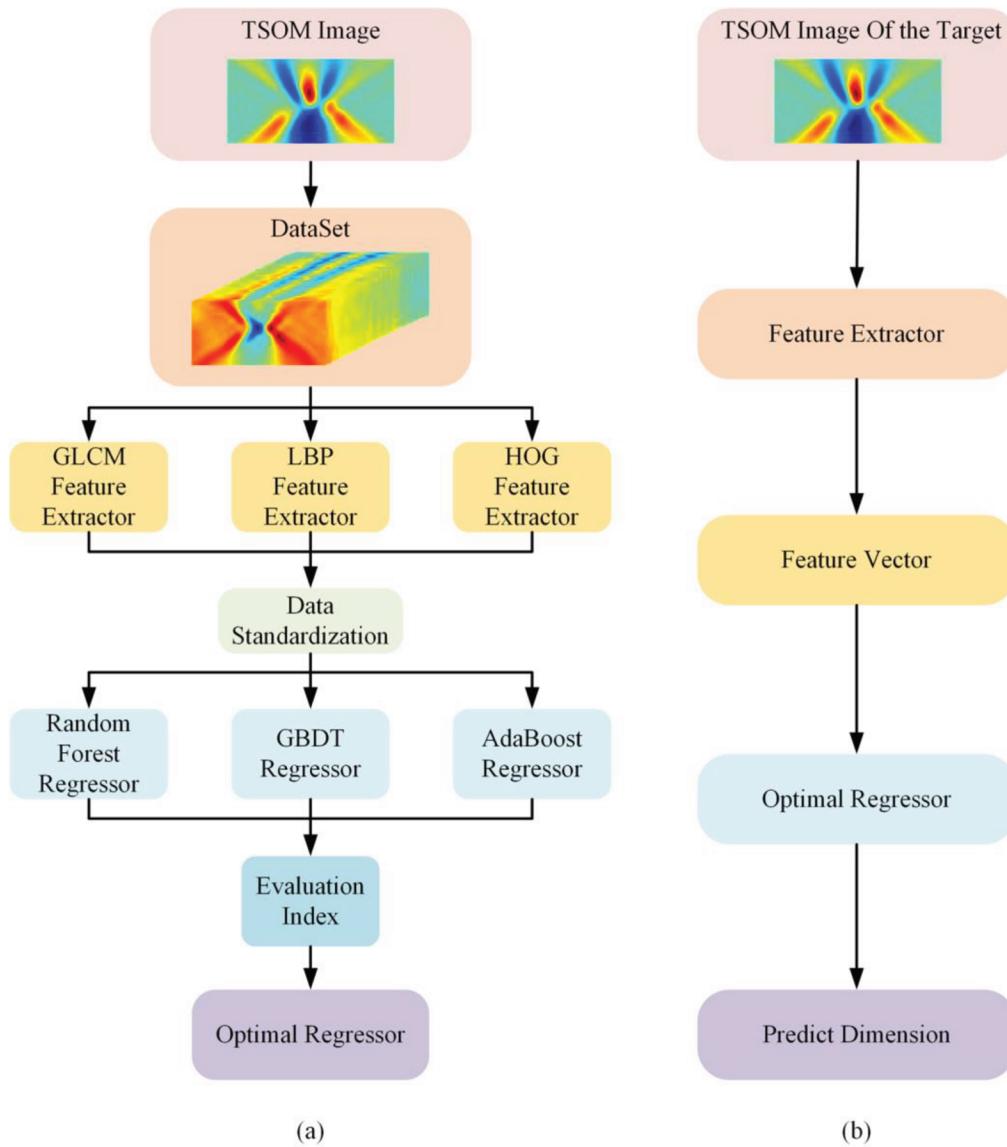
## 2. Principle

### 2.1. Machine-learning TSOM image analysis method

The process of building the TSOM image machine learning model is diagrammed in Fig. 1(a). First, the TSOM images are acquired by the through-focus scanning microscope imaging system. These TSOM images are then used to build a data set for training machine learning models. The indicators of GLCM, LBP, and HOG are used to extract feature vectors from the TSOM images in the data set, and the resulting feature vectors are combined to yield seven groups of feature data sets. Then, after normalizing the feature vectors in these feature sets, the three machine learning regression models of Random Forest, GBDT and AdaBoost are trained. Finally, through the analysis and comparison of experimental data, the optimal machine learning regression model and combination of feature vectors are identified. This optimal machine learning model is then used to evaluate the dimension of the target nanostructures, as shown in Fig. 1(b).

### 2.2. TSOM image acquisition device and creation process

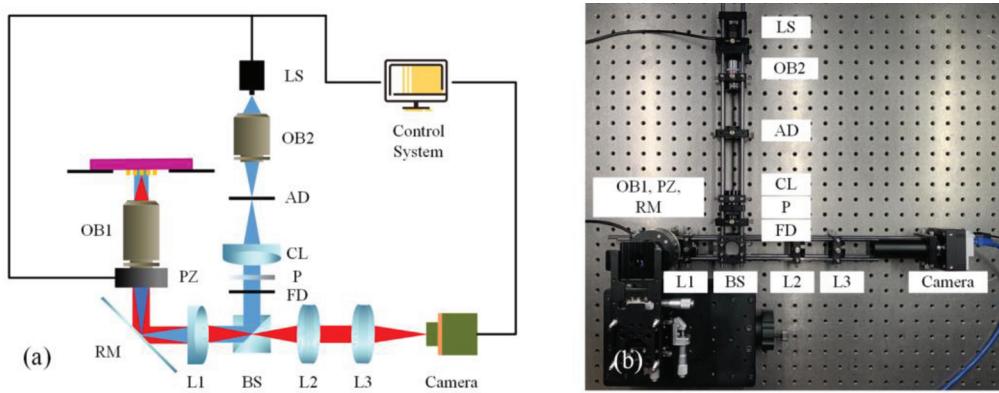
In this paper, TSOM images were acquired with the equipment shown in Fig. 2. The imaging system uses a LED (Boxingyuanzhi, China, BX-PLSL-8-1-G) as the light source (LS), which



**Fig. 1.** Steps establishing a machine-learning model for TSOM image processing. (a) Flowchart of model establishment; (b) Process of machine-learning image analysis

has an illumination wavelength of  $\lambda = 520\text{nm}$ . The light emitted by the LS passes through the beam-expanding objective OB2 (Nikon, Japan, M Plan, 5 $\times$ ), the aperture diaphragm AD (ZLYC Tech, China, ZLYC-008-011), the collimating lens CL (Daheng Optics, China, GCL-010604), the polarizer P (Daheng Optics, China, GCL050003), the field diaphragm FD (ZLYC Tech, China, ZLYC-008-011), the beam-splitter prism BS (ZLYC Tech, China, ZLYC-001-010), and the lens L1 (Daheng Optics, China, GCL-010604,  $f = 100\text{mm}$ ). The light finally converges on the conjugate rear focal plane of the objective lens OB1 (Mitutoyo, Japan, M Plan Apo, 50 $\times$ ,  $f = 4\text{mm}$ ). Then, the light illuminates the surface of the sample via OB1. OB2, AD, CL, P, FD, BS and L1 together form a Kohler illumination system. CNA is same as the NA of the objective lens OB1 which is 0.55. The diameter of the AD DAD is 1mm and the focal length of CL and

L1 is 100mm, so the spot diameter on the focal plane behind the objective lens is 1mm. INA can be calculated that  $\sin \frac{\theta}{2} \approx \tan \frac{\theta}{2} = \frac{D_{AD}}{2 \times f_{OB1}} = 0.125$ , where  $\theta$  is the angle of illumination. The piezoelectric objective locator PZ (Coremorrow, China, P76.Z100S & E53.D) is connected to OB1, and it scans OB1 up and down along the optical axis. OB1, reflecting mirror RM, L1, lens L2 (Daheng Optics, China, GCL-010653,  $f = 60\text{mm}$ ), BS, and lens L3 (Daheng Optics, China, GCL-010605,  $f = 150\text{mm}$ ) form the optical imaging path. The resolution of the CCD camera (Basler, Germany, avA2300-25gc) used in the optical system is  $2330 \text{ px} \times 1750 \text{ px}$ .

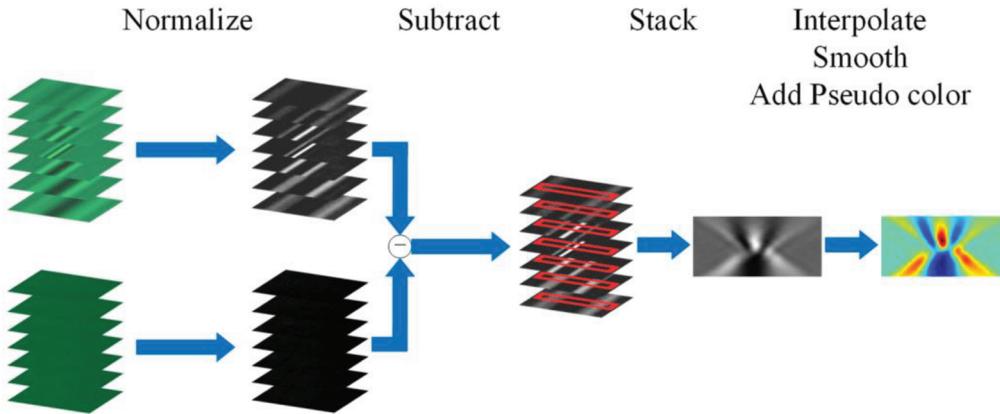


**Fig. 2.** TSOM system based on objective scanning. (a) Schematic diagram of the system; (b) Photograph of the system. LS, LED light source; OB2, beam expanding objective; AD, aperture diaphragm; CL, collimating lens; P, polarizer; FD, field diaphragm; OB1, objective lens; PZ, piezoelectric objective locator; RM, reflecting Mirror; BS, beam-splitting prism; L1, L2, L3, lens.

After the target and background defocused image sequences are acquired by the TSOM system, a three-dimensional normalized image sequence is constructed. The horizontal x- and y-axes, the vertical z-axis, and the gray levels of the normalized image sequence represent the two-dimensional spatial position, the focal position, and the optical intensity, respectively. A two-dimensional cross-section of this sequence is taken to obtain a TSOM image. As shown in Fig. 3, the TSOM image is constructed with the following five steps:

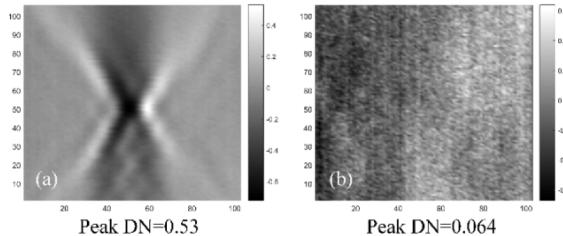
- 1) Record an in-focus image and a series of defocused images of the target and a smooth background surface as the camera scans along the optical axis. The z-axis range is  $-8\text{~}\mu\text{m}$  to  $8\text{~}\mu\text{m}$ , and the z-step increment is  $200\text{nm}$ .
  - 2) Normalize the through-focus images of the target and background as follows:
- $$\text{Normalized Image} = \frac{\text{UnnormalizedImage}}{\text{MeanValue}} - 1 \quad (3)$$
- 3) Subtract the normalized background images from the normalized target images directly.
  - 4) Select the rectangular detection region in each image in the image sequence, and extract the average pixel intensity along the width of the rectangles. Then stack these data according to their original positions to get the TSOM image.
  - 5) Interpolate and smooth the original TSOM image and add pseudocolor. The result is the final TSOM image. First, bilinear interpolation is selected as interpolation algorithm. Then, there are two steps in the smoothing process. The first step is to use convolution kernel  $[1/3, 1/3, 1/3]$  and convolution kernel  $[1/3, 1/3, 1/3]^T$  to average filter the image

along the x and y directions respectively. The second step is to use convolution kernel  $[1/6, 1/6, 1/6, 1/6, 1/6, 1/6]$  and convolution kernel  $[1/6, 1/6, 1/6, 1/6, 1/6, 1/6]^T$  to average filter the image along the x and y directions respectively. The convolution mode is both ‘valid’. During these two process, interpolation will improve nanometer sensitivity, and smoothing will reduce nanometer sensitivity.



**Fig. 3.** Steps in TSOM image construction

During the above steps, a single stacked TSOM image of 239.3nm linewidth and stacked background TSOM are shown in Fig. 4. After the above steps, some measured raw and processed TSOM images can be obtained, as shown in Fig. 5.

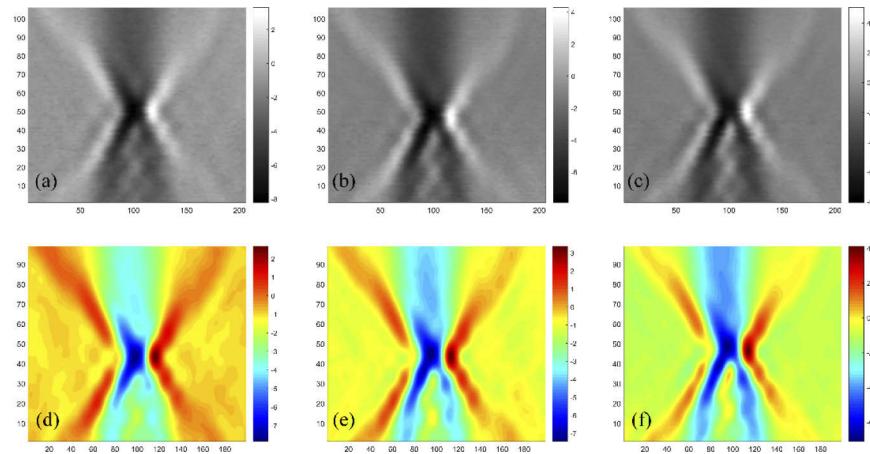


**Fig. 4.** (a) Stacked TSOM image of 239.3nm linewidth, Peak Digital Number(DN) is 0.53; (b) Stacked background TSOM image; Peak Digital Number(DN) is 0.064, noise mean is  $-5.7 \times 10^{-17}$ , noise std is 0.02.

### 2.3. Algorithms for feature vector extraction

#### 2.3.1. Gray-level Co-occurrence Matrix

The Gray-level Co-occurrence Matrix (GLCM) is defined as the joint distribution probability of pixel pairs [19]. The TSOM image is described using GLCM is as follows: Q is an operator that defines the position of two pixels relative to each other. And it is for extracting GLCM ranges from 1 pixel to 50 pixels. At each pixel distance, GLCMs of TSOM image are extracted in the orientations of  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$ , respectively. A total of 200 GLCMs are obtained as a result. Then, the correlation, contrast, energy, homogeneity, and entropy of each GLCM are calculated according to Eqs. (4) - (8). Finally, five descriptors of each GLCM are combined to obtain a feature vector with the length of 1000 ( $50 \times 4 \times 5$ ). The feature extraction process is



**Fig. 5.** (a) Raw TSOM image of 239.3nm linewidth; (b) Raw TSOM image of 299.3nm linewidth; (c) Raw TSOM image of 355nm linewidth; (d) Processed TSOM image of 239.3nm linewidth; (e) Processed TSOM image of 299.3nm linewidth; (f) Processed TSOM image of 355nm linewidth.

illustrated in Fig. 6. The five descriptors are defined as follows:

$$\text{Correlation} = \sum_{i=1}^K \sum_{j=1}^K \frac{(i - m_r)(j - m_c)P_{ij}}{\sigma_r \sigma_c} \quad (4)$$

$$\text{Contrast} = \sum_{i=1}^K \sum_{j=1}^K (i - j)^2 p_{ij} \quad (5)$$

$$\text{Energy} = \sum_{i=1}^K \sum_{j=1}^K p_{ij}^2 \quad (6)$$

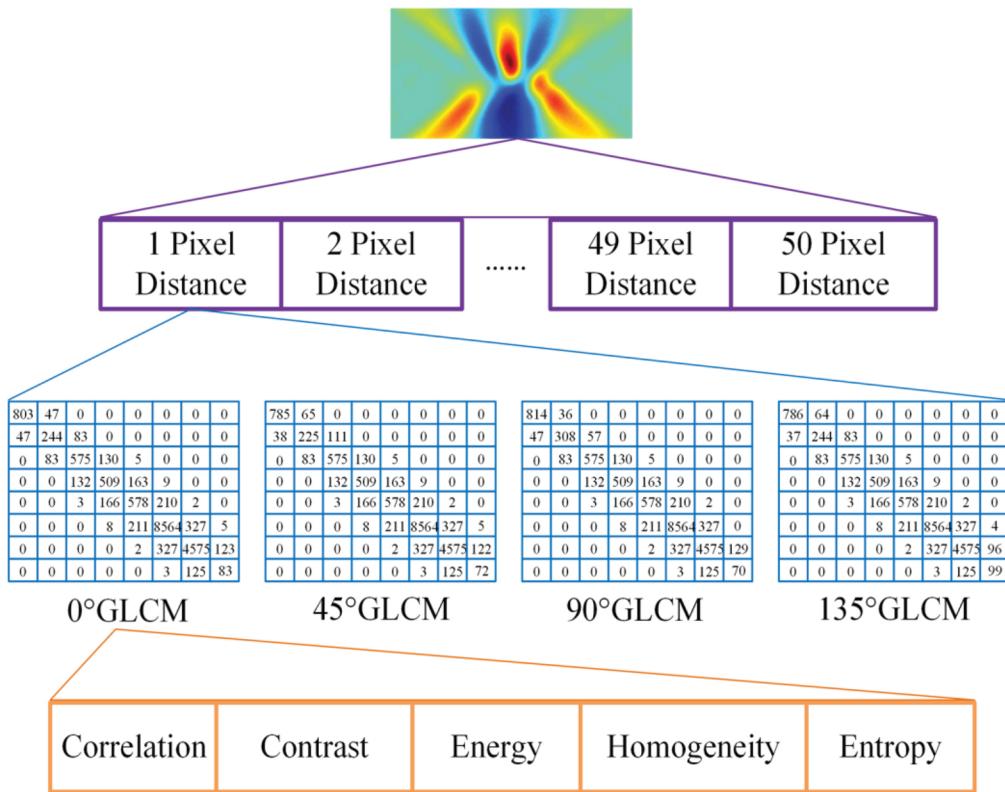
$$\text{Homogeneity} = \sum_{i=1}^K \sum_{j=1}^K \frac{p_{ij}}{1 + |i - j|} \quad (7)$$

$$\text{Entropy} = - \sum_{i=1}^K \sum_{j=1}^K p_{ij} \log_2 p_{ij} \quad (8)$$

in the above equations,  $i$  and  $j$ , respectively, are the row and column numbers of matrix  $G$  that represents the GLCM, and  $1 \leq i, j \leq L$ ;  $L$  is the gray level of the whole TSOM image;  $K$  is the total number of rows or columns in  $G$ ;  $m_r = \sum_{i=1}^K i \sum_{j=1}^K p_{ij}$  is the mean value calculated along the rows of  $G$ ;  $m_c = \sum_{j=1}^K j \sum_{i=1}^K p_{ij}$  is the mean value calculated along the columns of  $G$ ;

$\sigma_r^2 = \sum_{i=1}^K (i - m_r)^2 \sum_{j=1}^K p_{ij}$  is the variance calculated along the rows of  $G$ ;  $\sigma_c^2 = \sum_{j=1}^K (j - m_c)^2 \sum_{i=1}^K p_{ij}$

is the variance calculated along the columns of  $G$ ;  $p_{ij} = \frac{g_{ij}}{n}$  is the estimated probability of point pairs, where  $g_{ij}$  is the element of  $G$  and  $n$  is the total number of pixel pairs that satisfy condition Q.



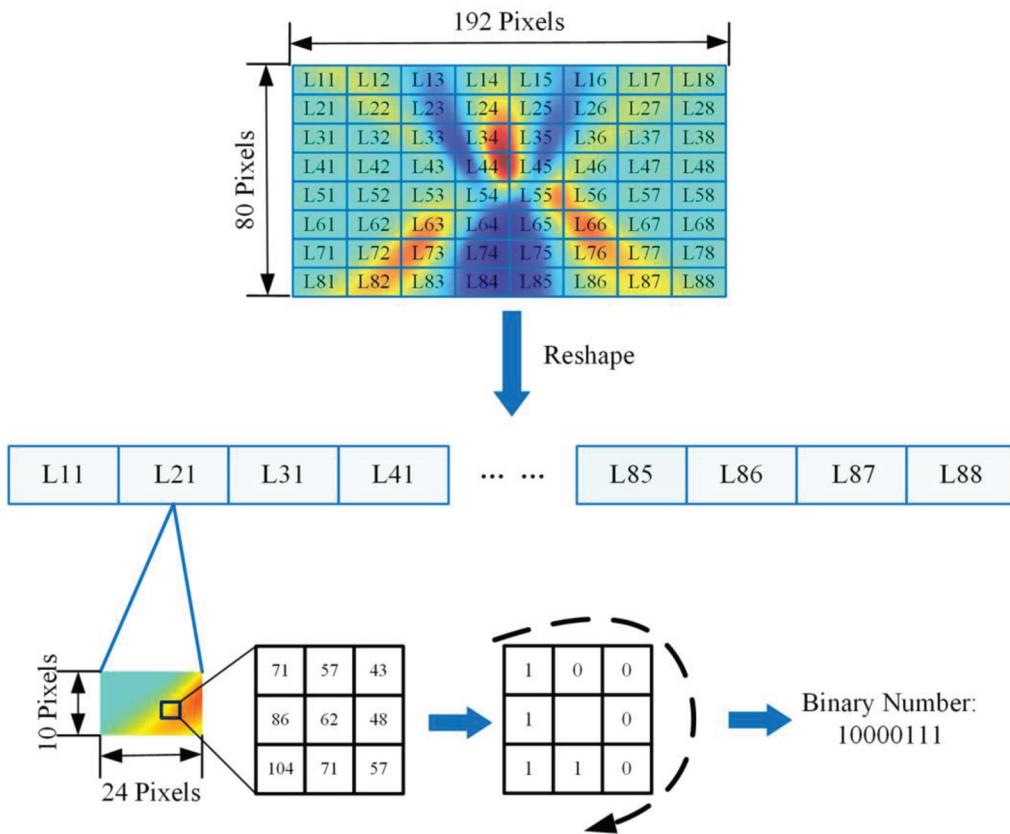
**Fig. 6.** Process for GLCM feature extraction

### 2.3.2. Local Binary Pattern

The LBP descriptor describes local texture features of an image, and the feature vector is formed by calculating the LBP histogram of each cell [20]. The process of LBP feature extraction is illustrated in Fig. 7. First, the TSOM image is divided into  $8 \times 8$  cells. In a  $3 \times 3$  window, the center pixel of the window is used as a threshold, and the gray value of the eight neighboring pixels is compared to this threshold. If the gray value of the neighboring pixel is greater than the gray value of the center pixel, the cell is marked as 1, otherwise it is marked as 0. Then, starting from the upper left corner of the element, this process proceeds clockwise to get an 8-bit binary number, which is the LBP value of the center element of the window. The 8-bit LBP value ranges from 0 to 255. To equalize the length of LBP feature vector with the lengths of the GLCM and HOG feature vectors, we use uniform patterns and miscellaneous patterns to simplify the LBP features of the TSOM image. The specific methods are as follows: each LBP value that occurs no more than twice and jumps from 1 to 0 or 0 to 1 is classified as a uniform pattern, so 58 LBP uniform patterns values can be obtained. LBP values with three or more jumps from 1 to 0 or 0 to 1 are classified as miscellaneous patterns. In this way, the range of LBPs is reduced from 256 to 59, and the length of the LBP feature vector is reduced to 3776 ( $8 \times 8 \times 59$ ).

### 2.3.3. Histogram of Oriented Gradient

HOG extracts local texture features from the image by calculating oriented gradient histogram of local regions [21]. The feature-extraction process for HOG is illustrated in Fig. 8. First, the TSOM image is divided into  $24 \times 10$  cells, each of which is  $8 \text{ px} \times 8 \text{ px}$ . Each block is defined to



**Fig. 7.** Process for LBP feature extraction

contain  $2 \times 2$  cells. The HOG feature vector is extracted by sliding these blocks from left to right in each row. When the block reaches the right end of the row, the block restarts at the left of the next row and continues to slide from left to right. The block covers the length of a cell in each step, or the distance of 8 pixels. As the block slides, the horizontal and vertical gradient operators  $[-1, 0, 1]$  and  $[-1, 0, 1]^T$  are firstly used to calculate the pixels in each cell  $H$  of the block, and the horizontal and vertical gradient magnitudes result:

$$G_x(x, y) = H(x + 1, y) - H(x - 1, y) \quad (9)$$

$$G_y(x, y) = H(x, y + 1) - H(x, y - 1) \quad (10)$$

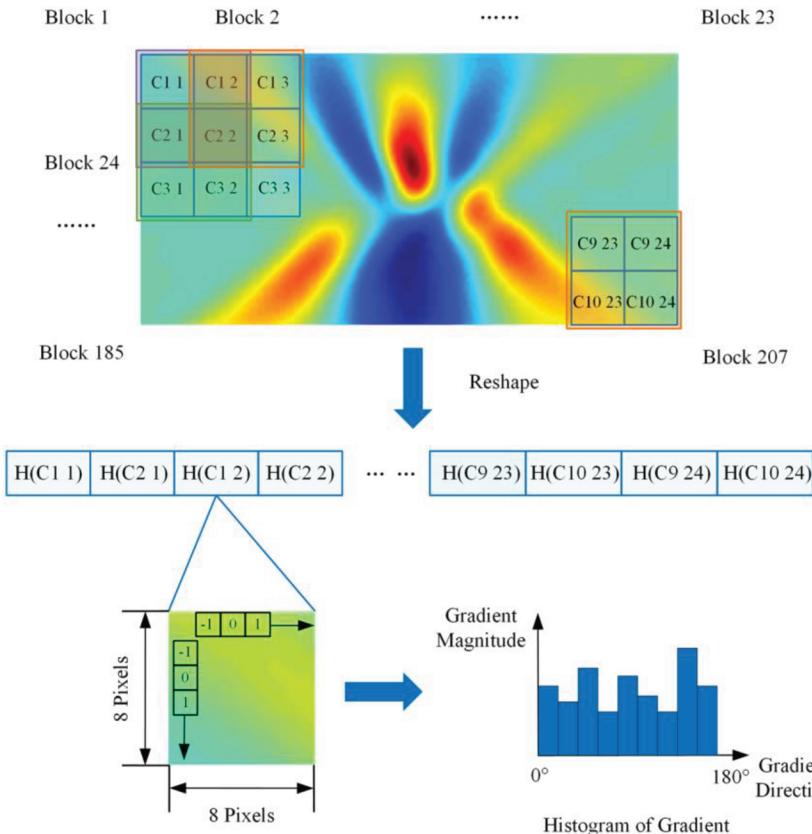
then, the magnitude and direction of the gradient of pixels in each cell can be obtained:

$$G(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)} \quad (11)$$

$$\alpha(x, y) = \tan^{-1} \left( \frac{G_y(x, y)}{G_x(x, y)} \right) \quad (12)$$

as the gradient direction ranges from  $0^\circ$  to  $180^\circ$ , it is divided into nine intervals, after which the direction and magnitude of the pixel gradient in each cell are counted. First, the gradient direction of the pixel is used to determine the interval that the pixel falls into, and then the statistical magnitude of this interval is added to the gradient magnitude of the pixel. The HOG

histogram of the cell is obtained after the block traverses each cell. The histograms of four cells in a block are straightened and connected in order. The resulting HOG feature vector of the block has length of  $4 \times 9$ . Finally, the feature vectors obtained by each block are normalized using L2 regularization, and the vectors obtained from each block are connected to obtain the HOG feature vector. The length of the final HOG feature vector is 7452 ( $23 \times 9 \times 4 \times 9$ ).



**Fig. 8.** Process for HOG feature extraction

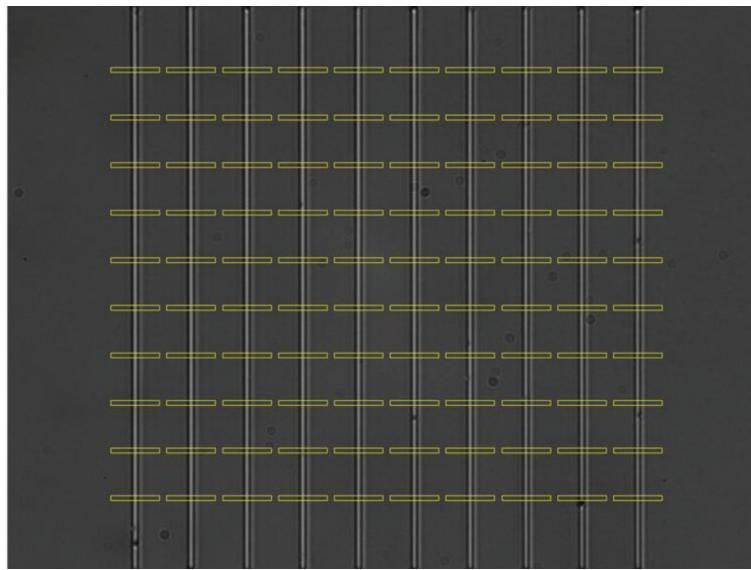
To study the influence of the above three feature-vector-extraction algorithms on the model fitting process, we tested seven combinations of vectors, which are listed in Table 1. Data set  $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$  results, where  $m$  is the number of members in the data set. For each sample in the data set  $(\mathbf{x}_i, y_i)$ ,  $i = 1, 2, \dots, m$ ,  $\mathbf{x}_i = \{x_{i1}; x_{i2}; \dots; x_{in}\}$  is an  $n$ -dimensional feature vector, and  $y_i$  is its label.

**Table 1. Combination of feature vectors**

Number of feature vectors	Combination of feature vectors		
1	GLCM	LBP	HOG
2	GLCM & LBP	GLCM & HOG	LBP & HOG
3		GLCM & LBP & HOG	

## 2.4. Data set collection

The measured samples are a series of gold lines fabricated by evaporation on a silicon substrate, with the design length of 100  $\mu\text{m}$  and design height of 100 nm. The sample includes 40 regions, each of which contains 10 gold lines. The gold lines are 10  $\mu\text{m}$  apart. The linewidths of the gold lines within a region is the same, and differ in different regions. The linewidths increase from 220 nm to 1000 nm, in increments of 20 nm. When collecting the dataset, the TSOM image is generated by selecting 10 equally spaced regions over each gold line of each region, as shown in Fig. 9. 100 TSOM images of the linewidth of each region can be obtained, and a total of 4000 TSOM images can be obtained from 40 regions.



**Fig. 9.** Image of gold-line region with linewidth of 1000 nm. Yellow rectangular boxes mark the regions imaged by TSOM.

From the 100 TSOM images obtained from each region, 80 images are randomly selected to form the training set, and the remaining 20 images form the test set. In this way, the dataset  $D$  includes information about 40 sets of gold lines with different linewidths, and each subset corresponds to a region of the experimental sample. Therefore, the dataset  $D$  in this study contains the training set  $S$  of 3200 TSOM images and the test set  $T$  of 800 TSOM images, where  $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_{3200}, y_{3200})\}$  and  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_{800}, y_{800})\}$ .

## 2.5. Feature vector standardization

When training a machine learning model with the seven feature data sets introduced in Section 2.3, all features in the feature data set must be standardized to improve accuracy, as shown in Eq. (13):

$$x_{ij}^* = \frac{x_{ij} - \mu_{x_j}}{\sigma_{x_j}} \quad (13)$$

where  $x_{ij}$  is any feature of any sample in the data set,  $\mu_{x_j} = \frac{1}{m} \sum_{i=1}^m x_{ij}$  is the mean value of the feature,  $\sigma_{x_j} = \sqrt{\frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_{x_j})^2}$  is the variance of the feature, and  $x_{ij}^*$  is the value of the feature property after standardization.

## 2.6. Machine learning regression model training

The Random Forest, Gradient Boosting Decision Tree, and AdaBoost regression models were trained using the feature datasets.

### 2.6.1. Random Forest

Random Forest is an extended variant of the bagging algorithm [22–24]. This algorithm, based on the bootstrapping method, extracts  $T$  samples with  $m$  samples from the training set  $S$  to the sampling set  $D_{bs}$ . Then a classification and regression tree (CART) is trained on each sample set. When making predictions, each decision tree is averaged with the same weight. Based on bagging integration, the Random Forest algorithm introduces random feature selection when training the decision tree. In the training process, for each node of the CART, a subset containing  $k$  features is randomly selected from the feature set at this node, and then an optimal feature is selected from this subset for division. The mean square error (MSE) loss function was used in the training process. The Random Forest algorithm used in this study is described with the following pseudocode:

Algorithm 1: Random Forest

1. Input the standardized training set  $S = \{(\mathbf{x}_1^*, y_1), (\mathbf{x}_2^*, y_2), \dots, (\mathbf{x}_m^*, y_m)\}$ ;
2. For  $t = 1$  to  $T$  :
  - (a) Use the bootstrapping method to extract  $m$  samples from the training set  $S$  to form the sample set  $D_{bs}$ ;
  - (b) For each node of CART, a subset containing  $k$  features is randomly selected from the feature set of this node;
  - (c) Select an optimal feature from the subset and divide the current node;
  - (d) Training for CART  $h_t(\mathbf{x}^*)$ ;
3. Output:  $H(\mathbf{x}^*) = \frac{1}{T} \sum_{t=1}^T h_t(\mathbf{x}^*)$

### 2.6.2. AdaBoost

AdaBoost is a boosting-class integrated learning algorithm [25]. Like the Random Forest algorithm, the base estimator of AdaBoost also uses CART. First, the weight distribution of samples in the training set  $S$  is initialized before training a CART. Next, the weight distribution of the training samples is adjusted according to the performance of the training set  $S$  in the CART, so that the samples with large prediction errors in the previous CART receive more attention in subsequent training steps. Then the next CART is trained with the training samples whose weight distributions have been adjusted. These steps are repeated until the number of estimators reaches the pre-specified value  $T$ . The prediction results of these base estimators are combined. The loss function of AdaBoost is an exponential. Pseudocode for the AdaBoost algorithm in this study is as follows:

## Algorithm 2: AdaBoost

1. Input the standardized training set  $S = \{(\mathbf{x}_1^*, y_1), (\mathbf{x}_2^*, y_2), \dots, (\mathbf{x}_m^*, y_m)\}$ , basic learning algorithm  $\mathcal{L}$ ;
2. Initialize weights  $\mathcal{D}_1(\mathbf{x}) = (w_{1,1}, \dots, w_{1,i}, \dots, w_{1,m})$ ,  $w_{1,i} = \frac{1}{m}, i = 1, 2, \dots, m$ ;
3. For  $t = 1$  to  $T$  :
  - (a) Train base estimator  $h_t = \mathcal{L}(S, \mathcal{D}_t)$ ;
  - (b) Calculate the maximum sample error on the training set
 
$$E_t = \max |y_i - h_t(\mathbf{x}_i^*)|, i = 1, 2, \dots, m$$
  - (c) Calculate the relative error for each sample
 
$$e_{ii} = \frac{|y_i - h_t(\mathbf{x}_i^*)|}{E_t}$$
  - (d) Calculate the regression error rate of the base estimator on the training set
 
$$\epsilon_t = \sum_{i=1}^m w_{t,i} e_{ii}$$
  - (e) Calculate the weight coefficient of the base estimator
 
$$\alpha_t = \frac{\epsilon_t}{1 - \epsilon_t}$$
  - (f) Update the sample weight distribution  $\mathcal{D}_{t+1}(\mathbf{x})$  of the training set
 
$$w_{t+1,i} = \frac{w_{t,i}}{Z_t} \alpha_t^{1-e_{ii}}$$
 where, the normalization factor is  $Z_t = \sum_{i=1}^m w_{t,i} \alpha_t^{1-e_{ii}}$ .
4. Output:  $H(\mathbf{x}^*) = \sum_{t=1}^T \left( \ln \frac{1}{\alpha_t} \right) g(\mathbf{x}^*)$ ,  $g(\mathbf{x}^*)$  is the median of all  $\alpha_t h_t(\mathbf{x}^*)$ ,  $t = 1, 2, \dots, T$ .

## 2.6.3. Gradient Boosting Decision Tree

The GBDT algorithm is also a boosting-integrated learning algorithm [26]. Unlike AdaBoost, the base estimator of the GBDT algorithm is limited to the CART regression model. The loss function can be any loss function, and we selected the least squares function. In the training process, GBDT differs from AdaBoost in that AdaBoost needs to update the distribution of the samples in the training set for each round of iteration. GBDT, in contrast iteratively learns the residual between the prediction result and the real value of the addition model obtained in the previous round and uses that residual to increase the weight of the error sample. The pseudocode for the GBDT algorithm is as follows:

## Algorithm 3: GBDT

1. Input the standardized training set  $S = \{(\mathbf{x}_1^*, y_1), (\mathbf{x}_2^*, y_2), \dots, (\mathbf{x}_m^*, y_m)\}$ , loss function  $L$ ;
2. Initialize model  $H_0(\mathbf{x}^*)$ , estimate the constant value  $\gamma$  that minimizes the loss function; the initial model is a regression tree with only one root node:  

$$H_0(\mathbf{x}^*) = \arg \min_{\gamma} \sum_{i=1}^m L(y_i, \gamma)$$
3. For  $t = 1$  to  $T$  :
  - (a) Calculate the generalized residuals of the current model for each sample  

$$r_{ti} = -\left[ \frac{\partial L(y_i, H(\mathbf{x}_i^*))}{\partial H(\mathbf{x}_i^*)} \right]_{H(\mathbf{x}_i^*)=H_{t-1}(\mathbf{x}_i^*)}$$
  - (b) Fit a regression tree with  $(\mathbf{x}_i^*, r_{ti}), i = 1, 2, \dots, m$  to obtain the  $t$ -th tree's leaf-node region  $\mathbf{R}_j, j = 1, 2, \dots, J$ ;
  - (c) For each leaf-node region, calculate the predictive value  $\gamma_j$  that minimizes the region's loss function  $\mathbf{R}_j$ .  

$$\gamma_j = \arg \min_{\gamma} \sum_{x_i \in \mathbf{R}_j} L(y_i, H_{t-1}(\mathbf{x}_i^*) + \gamma)$$
  - (d) The best-fit regression tree of this iteration is obtained:  

$$h_t(\mathbf{x}^*) = \sum_{j=1}^J \gamma_j I(\mathbf{x}^* \in \mathbf{R}_j)$$
  - (e) Update the addition model for this iteration:  

$$H_t(\mathbf{x}^*) = H_{t-1}(\mathbf{x}^*) + \sum_{j=1}^J \gamma_j I(\mathbf{x}^* \in \mathbf{R}_j)$$
4. Output:  $H(\mathbf{x}^*) = H_T(\mathbf{x}^*) = \sum_{t=1}^T \sum_{j=1}^J \gamma_j I(\mathbf{x}^* \in \mathbf{R}_j)$

### 3. Results and discussion

To analyze and evaluate the performance of our machine-learning TSOM image analysis method, we compare the proposed algorithm against the commonly used library-matching method.

#### 3.1. Model evaluation index

The linewidth of the gold-line target is predicted by a regression model, and we use the mean absolute error (MAE) and mean square error (MSE) to evaluate the trained models.

The definitions of MAE and MSE are given in Eqs. (14) and (15), respectively:

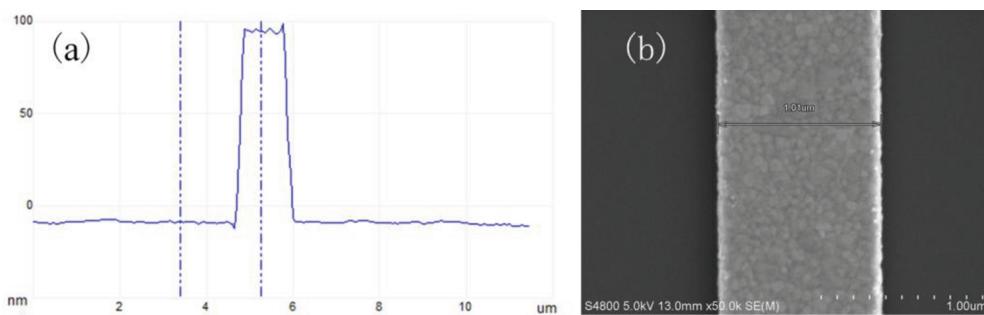
$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (14)$$

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (15)$$

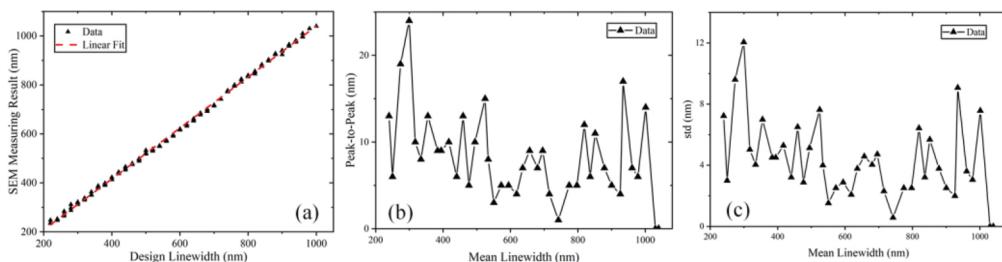
where  $y_i$  is the label of test data,  $\hat{y}_i$  is the predicted value of the test data, and  $n$  is the total number of test data.

### 3.2. Experimental sample actual-size measurements

In this paper, AFM was first used to take ground-truth measurements of the height of the gold lines in 40 regions of the sample. The heights had an average value of 103.89 nm and standard deviation of 2.92 nm. Then SEM was used to take ground-truth measurements of the linewidths of the targets by randomly selecting positions in the 40 regions. The AFM and SEM measurement results are shown in Fig. 10(a) and (b), respectively. The SEM measurement results are plotted against the designed linewidths in Fig. 11(a). The linear fit of the measurement results and the designed linewidths is  $y = 1.0402x + 0.1069$ , and the correlation coefficient is  $R^2 = 0.9990$ . The actual linewidths of the sample are consistent with the designed linewidths, the peak-to-peak values and the standard deviation values are shown in Fig. 11(b) and (c), respectively. Therefore, the SEM measurement results of each region are taken as the ground-truth linewidths of the gold lines.



**Fig. 10.** Example measurements of gold lines with designed linewidths up to 1000 nm. (a) AFM measurement; (b) SEM measurement.



**Fig. 11.** Designed linewidth of sample and SEM measurement results. (a) Linear equation is  $y = 1.0402x + 0.1069$ ,  $R^2 = 0.9990$ ; (b) Peak-to-peak values of SEM measurement results, the mean peak-to-peak value is 8.3nm; (c) Standard deviation of SEM measurement results, the mean standard deviation value is 4.3nm.

### 3.3. Dataset description

The library-matching method is best suited for predicting target sizes in a narrow range. Therefore, this paper defines the TSOM image data sets of 40 linewidths obtained in Section 2.4 as the data set with a wide size range with linewidths ranging from 239.7 nm to 1040 nm. 6 sets of data from 394.3 nm to 494.7 nm are defined as the narrow-range data set. The training set and test set of the wide-range data set contain 3200 and 800 TSOM images, respectively. The training set and test set of the narrow-range data set contain 480 and 120 TSOM images, respectively.

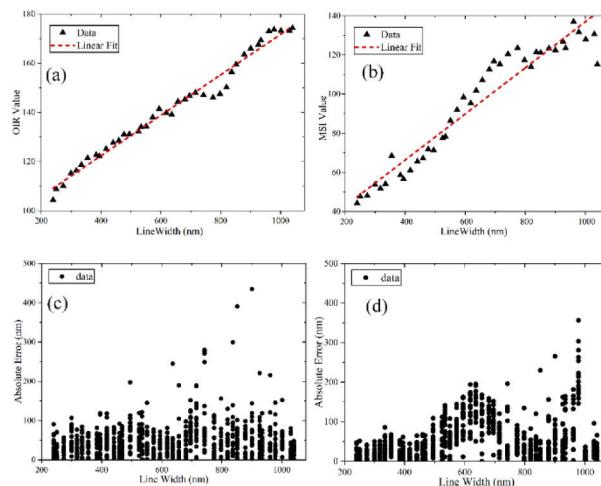
First, the key parameters (OIR and MSI) of all TSOM images in the datasets were extracted. Next, the key parameters extracted from each TSOM image in the test set were used to form

a test set for the library-matching method. Finally, the key parameters of the same class of TSOM image data in the training set were averaged, and these average values were used to fit the linear model of the library-matching method. The linear model obtained by applying the library-matching method to the wide-range data set was obtained by fitting 40 data points, and the test set is composed of 800 ( $40 \times 20$ ) test data. The linear model obtained by applying the library-matching method to the narrow-range data set was obtained by fitting 6 data points, and the test set is composed of 120 ( $6 \times 20$ ) test data.

### 3.4. Comparison and analysis

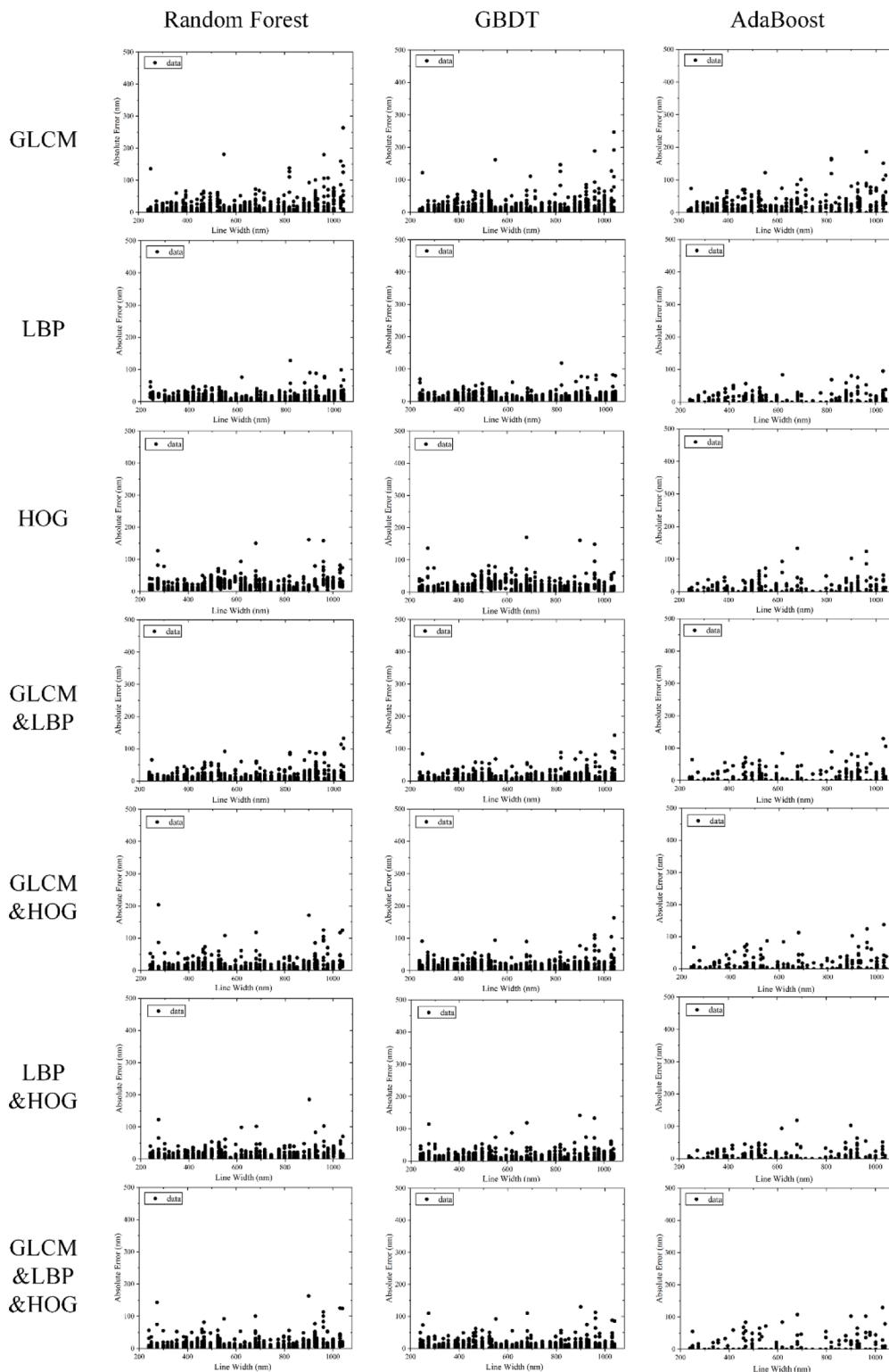
#### 3.4.1. Model comparison on wide-range dataset

The library-matching method was first tested with the wide-range test set. The linear regression results of OIR and MSI for the wide-range dataset are shown in Fig. 12. The MAEs of the OIR and MSI models on the test set were 47.0nm and 55.7nm, respectively. The MSEs were  $4046.6\text{nm}^2$  and  $5874.7\text{nm}^2$ , respectively. The OIR-based linear model performs better than the MSI-based linear model when using the wide-range dataset.



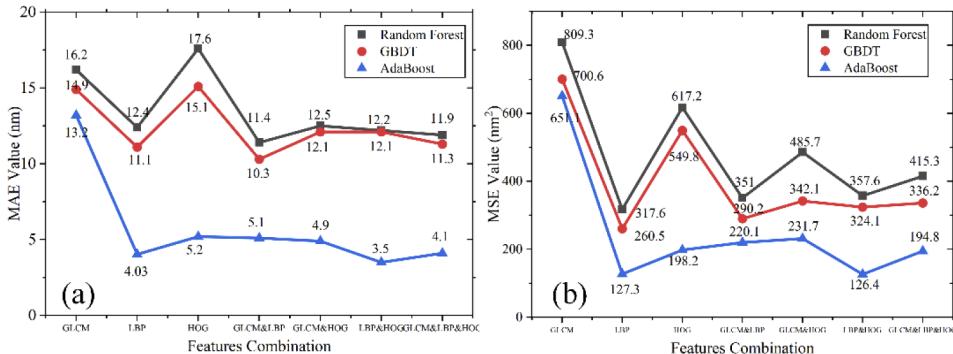
**Fig. 12.** Linear regression results of OIR and MSI in the wide-range data set. (a) OIR-based linear regression model of the wide-range test set; the model is  $y = 0.086x + 88.33$ , MAE is 47.0nm, and MSE is  $4046.6\text{nm}^2$ ; (b) MSI-based linear regression model of the wide-range test set, the model is  $y = 0.123x + 17.41$ , MAE is 55.87nm, and MSE is  $5874.7\text{nm}^2$ ; (c) Absolute error of OIR-based linear regression model on wide-range test set; (d) Absolute error of MSI-based linear regression model on wide-range test set. The horizontal axis is the label of the test data, and the vertical axis is the absolute error of the prediction result.

The results from training the machine-learning models on the wide-range dataset are shown in Fig. 13. The scatter plots shown in the figure give the performance of each regression model on the test set. The MAE and MSE of the three machine learning models on the seven feature-set combinations are shown in Fig. 14. The training time and process time of the three machine learning models on the seven feature-set combinations are shown in Fig. 15. According to the two evaluation indexes, one can see that the error of the machine-learning method is much smaller than that of the library-matching method. Moreover, the overall performance of the GBDT model is better than that of the random forest model, and the AdaBoost model performs better than both. From Fig. 15(a), mean training times of Random Forest, GBDT and AdaBoost are 632.8s, 2376.0s, 2358.0s, respectively. It can be seen that the training time of random forest model is

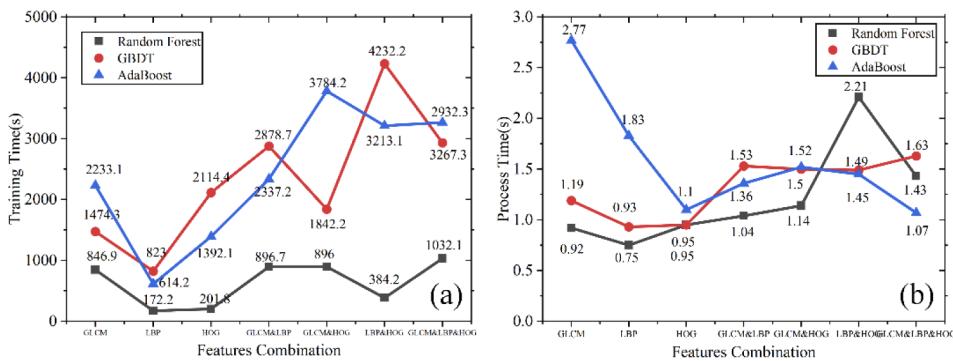


**Fig. 13.** Results of machine learning model in the wide-range test set. The horizontal axis is the label of the test data, and the vertical axis is the absolute error of the prediction results.

less than AdaBoost and GBDT, because of it is parallel processing during training process. From Fig. 15(b), it can be seen that the three models have average process time about 1.4s.



**Fig. 14.** Evaluation indices of 21 machine-learning models using the wide-range data set.  
(a) MAEs of 21 machine-learning models; (b) MSEs of 21 machine-learning models.



**Fig. 15.** (a) Full training times of 21 machine-learning models; (b) Process times of 21 machine-learning models. Full training time is the time from extracting the feature vectors of the TSOM images in the dataset to the completion of training. Process time is the time from extracting the feature vector of the TSOM image to the outputting predicted result. The average time to construct TSOM image is 18.3s.

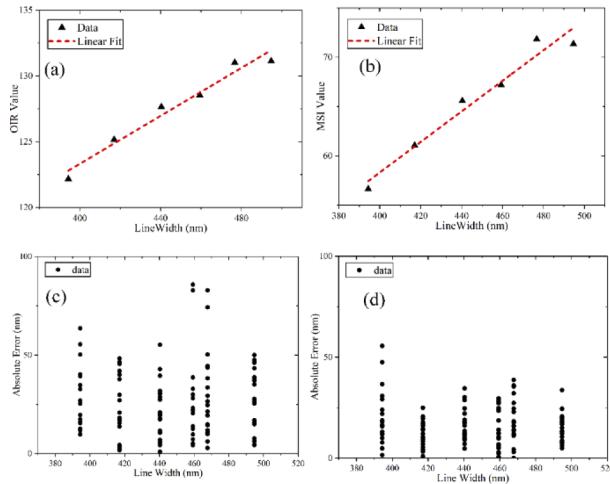
As one can see in Fig. 14, among the 21 models trained and tested with the wide-range dataset, the best performing model is AdaBoost model with the combined LBP and HOG feature-vector extraction algorithms, the MAE is 3.5nm and the MSE is 126.4nm<sup>2</sup>. Therefore, using the wide-range dataset, the best performing machine-learning model were obtained and these were compared with the best model of the library-matching method, as shown in Table 2. Then compare the MSEs of the two models and select the model with the smallest MSE as the best model of the wide-range dataset. The AdaBoost model with the combined LBP and HOG feature-vector extraction algorithms proves the optimum model of the wide-range dataset.

**Table 2. Model comparison with wide-range dataset**

Model	AdaBoost(LBP&HOG)	library-matching method (OIR)
MAE	3.5nm	47.0nm
MSE	126.4nm <sup>2</sup>	4046.6nm <sup>2</sup>

### 3.4.2. Model comparison on narrow-range dataset

Similarly, the library-matching method was first used to test the data of the narrow-range test set, and the fitting results are shown in Fig. 16. The MAEs of the OIR and MSI models were 25.4nm and 16.0nm, respectively, and the MSEs were  $968.1\text{nm}^2$  and  $356.1\text{nm}^2$ . The MSI-based linear model performs better than the OIR-based linear model on the narrow-range dataset.

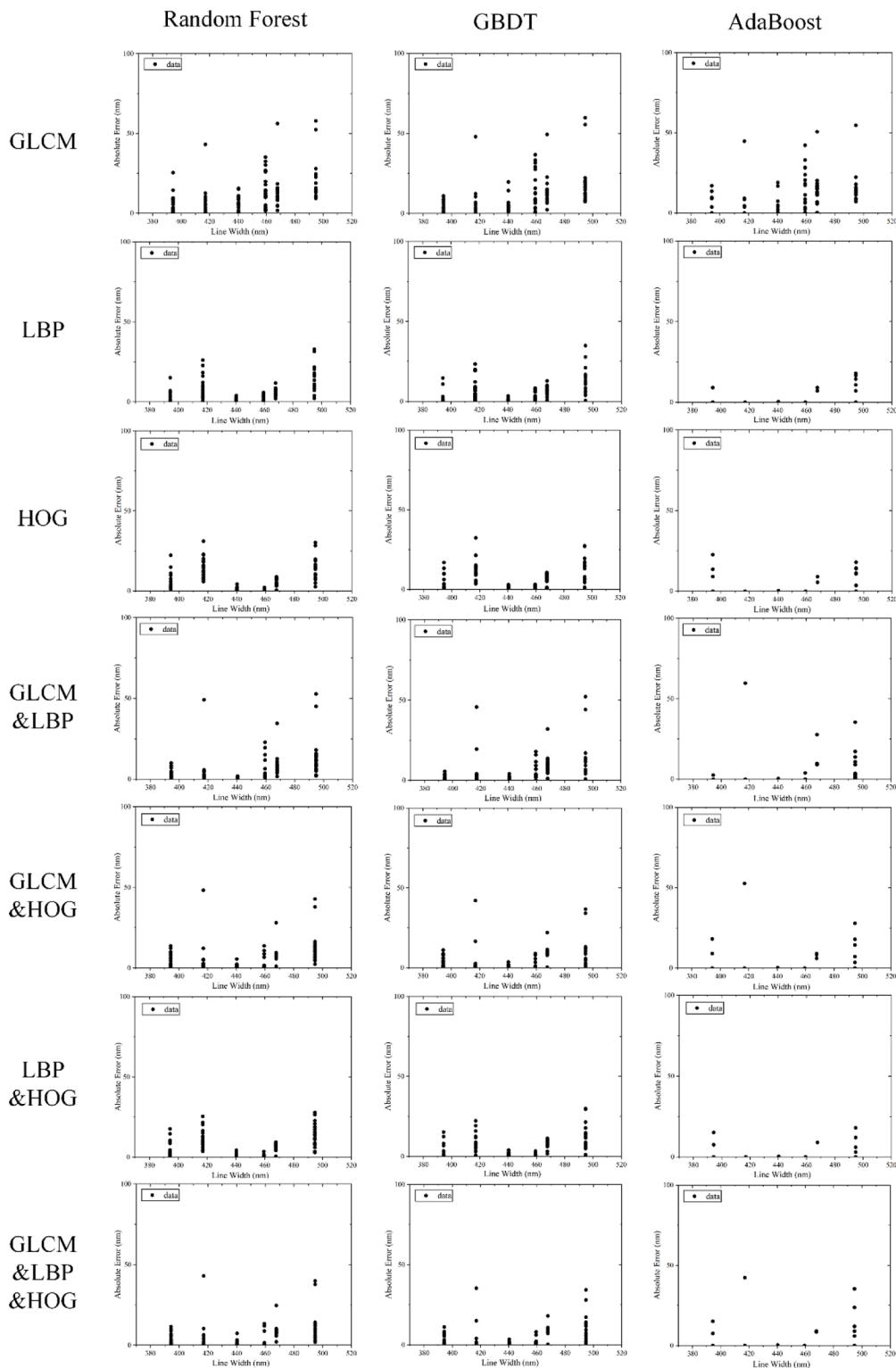


**Fig. 16.** Linear regression results of OIR and MSI on the narrow-range dataset. (a) OIR-based linear regression model on the narrow-range test set; the model is  $y = 0.079x + 92.50$ , MAE is 25.4nm, and MSE is  $968.1\text{nm}^2$ ; (b) MSI-based linear regression model on the narrow-range test set, the model is  $y = 0.186x - 16.44$ , MAE is 16.0nm, and MSE is  $356.1\text{nm}^2$ ; (c) Absolute error of OIR-based linear regression model of the narrow-range test set; (d) Absolute error of MSI-based linear regression model of the narrow-range test set. The horizontal axis is the label of the test data, and the vertical axis is the absolute error of the prediction result.

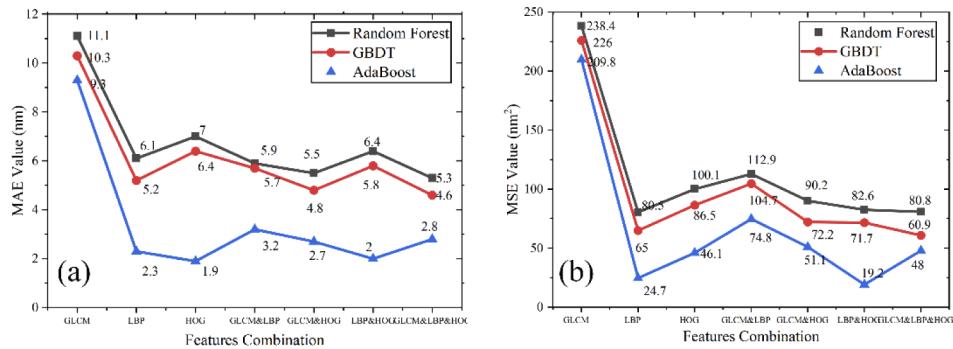
The results from training the machine-learning models on the narrow-range dataset are shown in Fig. 17. The MAE and MSE of the three machine learning models on the seven feature-set combinations are shown in Fig. 18. The training time and process time of the three machine learning models on the seven feature-set combinations are shown in Fig. 19. According to the two evaluation indexes, the machine-learning methods are clearly more accurate than the linear regression. Like above, the overall performance of the GBDT model is better than that of the random-forest model, but the AdaBoost model performs better than both. From Fig. 19(a), mean training times of Random Forest, GBDT and AdaBoost are 77.2s, 160.2s, 163.7s, respectively. It can be seen that the training time of random forest model is also less than AdaBoost and GBDT. From Fig. 19(b), it can be seen that the three models have average process time about 0.8s.

As one can see from Fig. 18, among the 21 models of the narrow-range dataset, the minimum MAE is 1.9nm. Setting a threshold of 0.5, models with MAE less than  $2.4 (1.9 + 0.5)\text{nm}$  are considered acceptable. Therefore, using the narrow-range dataset, three machine learning models were acceptable and these were compared with the best model from the library-matching method, as shown in Table 3. Then we compared the MSEs of the four models and selected the model with the smallest MSE value as the best model. Here, the AdaBoost model trained using the LBP and HOG feature vector extraction algorithm is selected.

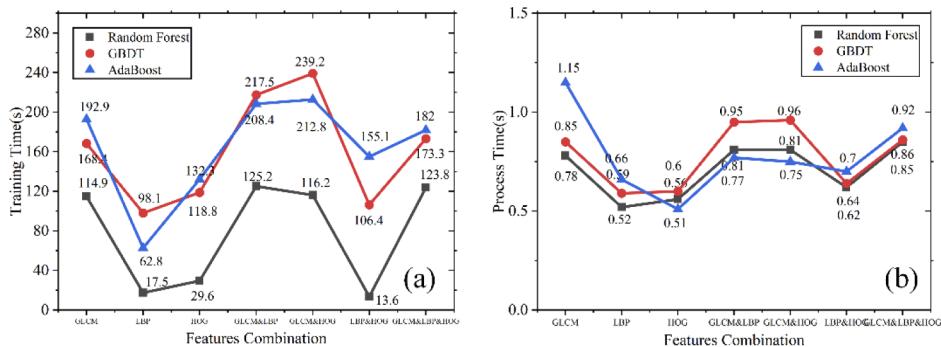
We have compared the results of predicting the linewidths of gold lines corresponding to the three machine learning models on seven combinations of three feature-vector extraction algorithms using wide- and narrow-range datasets. The results show that the AdaBoost regression



**Fig. 17.** Results of machine learning model on the narrow-range test set. The horizontal axis gives the labels of the test data, and the vertical axis gives the absolute error of the prediction results.



**Fig. 18.** Evaluation indexes of 21 machine learning models of the narrow-range dataset. (a) MAEs of 21 machine learning models; (b) MSEs of 21 machine learning models.



**Fig. 19.** (a) Full training times of 21 machine-learning models; (b) Process times of 21 machine-learning models. Full training time is the time from extracting the feature vectors of the TSOM images in the dataset to the completion of training. Process time is the time from extracting the feature vector of the TSOM image to the outputting predicted result. The average time to construct TSOM image is 18.3s.

**Table 3. Model comparison with narrow-range dataset**

Model	AdaBoost(HOG)	AdaBoost(LBP&HOG)	AdaBoost(LBP)	library-matching method (MSI)
MAE	1.9nm	2nm	2.3nm	16.0nm
MSE	46.1nm <sup>2</sup>	19.2 nm <sup>2</sup>	24.7nm <sup>2</sup>	356.1nm <sup>2</sup>

model performs best with both the wide-range and small-range datasets. When using the wide-range dataset, the AdaBoost model using the combined LBP and HOG feature-vector extraction algorithms performs best; on the narrow-range dataset, AdaBoost model using the LBP and HOG feature vector extraction algorithm alone performs best. Among all the above models, the minimum MSE is 19.2nm<sup>2</sup>, the standard deviation is 4.4nm. Within two sigma prediction, the measurement uncertainty is 8.8nm.

#### 4. Conclusion

The method proposed in this paper predicts the linewidth of isolated gold lines from TSOM images with nanoscale accuracy. The method extracts texture features from a TSOM pattern of the measurement target and uses these features to train machine-learning regression models. The proposed method makes full use of all texture information in the TSOM image. This use

of all available data allows the machine-learning approach to have a much wider measurement range than library-matching methods that use linear-regression models. The method also offers significantly improved accuracy and robustness in measurement results. This method is simple and fast, and will make TSOM imaging useful for online real-time and mass-scale measurements of nanoscale patterns. In future work, we plan to develop this method further to include multiple dimensional measurements of nanostructures, more-complex prediction models, and more-effective feature vector extraction algorithms.

## Funding

National Natural Science Foundation of China (51675033).

## References

1. B. D. Bunday, M. Bishop, D. W. McCormack, J. S. Villarrubia, A. E. Vladar, R. Dixson, T. V. Vorburger, N. G. Orji, and J. A. Allgair, "Determination of optimal parameters for CD-SEM measurement of line-edge roughness," *Proc. SPIE* **5375**, 515 (2004).
2. A. Eberle, S. Mikula, R. Schalek, J. Lichtman, M. K. Tate, and D. Zeidler, "High-resolution, high-throughput imaging with a multibeam scanning electron microscope," *J. Microsc.* **259**(2), 114–120 (2015).
3. T. Bao, L. Mininni, and D. Dawson, "Improving sidewall profile metrology with enhanced 3D-AFM," *Proc. SPIE* **7140**, 71400H (2008).
4. R. Attota, R. M. Silver, and J. Potzick, "Optical illumination and critical dimension analysis using the through-focus focus metric method," *Proc. SPIE* **6289**, 62890Q (2006).
5. R. Attota, R. Silver, and R. Dixson, "Linewidth measurement technique using through-focus optical images," *Appl. Opt.* **47**(4), 495–503 (2008).
6. R. Attota, R. Silver, and B. M. Barnes, "Optical through-focus technique that differentiates small changes in line width, line height, and sidewall angle for CD, overlay, and defect metrology applications," *Proc. SPIE* **6922**, 69220E (2008).
7. R. Attota, T. A. Germer, and R. M. Silver, "Through-focus scanning-optical-microscope imaging method for nanoscale dimensional analysis," *Opt. Lett.* **33**(17), 1990–1992 (2008).
8. R. Attota, R. G. Dixson, and A. E. Vladár, "Through-focus scanning optical microscopy," *Proc. SPIE* **8036**, 803610 (2011).
9. H. Kang, R. Attota, V. Tondare, A. E. Vladár, and P. Kavuri, "A method to determine the number of nanoparticles in a cluster using conventional optical microscopes," *Appl. Phys. Lett.* **107**(10), 103106 (2015).
10. R. Attota, P. P. Kavuri, H. Kang, R. Kasica, and L. Chen, "Nanoparticle size determination using optical microscopes," *Appl. Phys. Lett.* **105**(16), 163105 (2014).
11. R. Attota and V. Jindal, "Inspecting mask defects with through-focus scanning optical microscopy," *SPIE Newsroom* (2013).
12. R. Attota and R. G. Dixson, "Resolving three-dimensional shape of sub-50 nm wide lines with nanometer-scale sensitivity using conventional optical microscopes," *Appl. Phys. Lett.* **105**(4), 043101 (2014).
13. B. Damazo, R. Attota, P. Kavuri, and A. E. Vladár, "Nanoparticle size and shape evaluation using the TSOM method," *Proc. SPIE* **8324**, 832436 (2012).
14. R. K. Attota, "Through-focus scanning optical microscopy applications," *Proc. SPIE* **10677**, 106770R (2018).
15. R. Attota, M. Stocker, R. Silver, A. Heckert, H. Zhou, R. Kasica, L. Chen, R. Dixson, G. Orji, B. Barnes, and P. Lipscomb, "Through-focus Scanning and Scatterfield Optical Methods for Advanced Overlay Target Analysis," *Proc. SPIE* **7272**, 727214 (2009).
16. R. Attota, B. Bunday, and V. Vartanian, "Critical dimension metrology by through-focus scanning optical microscopy beyond the 22 nm node," *Appl. Phys. Lett.* **102**(22), 222107 (2013).
17. R. Attota, "Noise analysis for through-focus scanning optical microscopy," *Opt. Lett.* **41**(4), 745–748 (2016).
18. R. K. Attota and H. Kang, "Parameter optimization for through-focus scanning optical microscopy," *Opt. Express* **24**(13), 14915–14924 (2016).
19. R. M. Haralick and K. Shanmugam, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern. SMC-3*(6), 610–621 (1973).
20. T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 971–987 (2002).
21. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2005), pp. 886–893.
22. L. Breiman, "Bagging predictors," *Mach. Learn.* **24**(2), 123–140 (1996).
23. L. Breiman, "Stacked regressions," *Mach. Learn.* **24**(1), 49–64 (1996).
24. L. Breiman, "Randomizing outputs to increase prediction accuracy," *Mach. Learn.* **40**(3), 229–242 (2000).
25. Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.* **55**(1), 119–139 (1997).
26. J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Ann. Stat.* **29**(5), 1189–1232 (2001).