Open-source software for SEM metrology

Iacopo Mochi, Michaela Vockenhuber, Timothée Allenet, and Yasin Ekinci

Paul Scherrer Institut, Villigen, Switzerland

ABSTRACT

We present SMILE, an open source software for the characterization of line and space patterns in SEM images. SMILE has been developed to provide a metrology platform which is open-source and, as such, easy to customize to specific needs and simple to integrate into a chain of analysis. SMILE is used to measure CD, LWR and unbiased LWR. The software is currently available as MATLAB code and under development for open platforms such as Python or Octave. Here we describe the main features of the software, its structure and the algorithms used to perform line edge detection, LWR calculation and LWR unbiasing.

Keywords: SEM, resist, metrology, LWR, LER, critical dimension, edge detection, unbiasing.

1. INTRODUCTION

The development of EUV resists capable of supporting current and future semiconductor technology nodes is a key requirement for the successful deployment of EUV lithography. The evaluation of the fundamental metrics that define the performance of a resist material is usually performed by the analysis of scanning electron microscopy (SEM) images of simple structures like line patterns and contact hole arrays. This approach is commonly used to investigate the quality of a resist platform in terms of roughness, dose sensitivity, and resolution under different exposure and development conditions. A carefully controlled data acquisition procedure and a correct quantitative analysis of the SEM images of the printed wafers are necessary to determine meaningful and comparable results. An essential step in this direction has been the development of the IMEC protocol for line width roughness (LWR) metrology,[1] which describes a set of parameters and rules for the SEM image collection and analysis. Several commercial software packages are available to perform the image analysis following the IMEC protocol and they represent an essential tool for EUV resist metrology. In this paper, we present SMILE (SEM-Measured Image Lines Estimator) a free and open-source software for SEM line and space analysis. The software has been developed for the analysis of SEM images of photoresists and periodic nanostructures. It can perform basic CD measurements, LWR estimation, PSD analysis, bias removal, and defect detection in SEM images. The software features different edge detection algorithms and LWR evaluation protocols that can be customized and extended to address user-specific needs. In this paper, we will describe the general structure of the software, the available edge detection algorithms and the procedures for the LWR measurement. We will also demonstrate the performance of the software with simulated and experimental data.

2. SOFTWARE STRUCTURE

SMILE is a collection of functions and scripts developed in MATLAB.[2] It can be used in combination with MATLAB or it can be compiled and run independently. SMILE consists of a graphical user interface, an image pre-processing module, an edge detection module and a set of functions to perform CD measurements, defects detection and LWR (Line Width Roughness) analysis.

Further author information: E-mail: iacopo.mochi@psi.ch

2.1 Graphical user interface

The GUI (graphical user interface) of SMILE is designed with the intention to provide the users with an easy way to input the SEM images, set the parameters for the analysis and retrieve the results in a convenient format. The GUI consists of a single panel with two tabs. The opening tab, shown in figure 1, includes a display of the currently selected image where the lines profiles and eventual defects are highlighted, and an interactive table showing the main metrics relative to the images under analysis. From this tab, the user can select a folder where to load the images from and can export the results of the analysis as a text file, a MATLAB data file or as a Microsoft Excel spreadsheet. This tab also includes a display to show the LWR power spectral density (PSD), the image local CD uniformity and the LWR height-height correlation function (HHCF).

The second tab, shown in figure 2, is used to set the parameters for the image analysis. These parameters are used to customize the behavior of the image pre-processing module and of the image detection module. They can also control the procedure used by the software to measure the LWR and its PSD. The software includes the possibility to read the header of the SEM images created by different microscope models available to us and extract the image pixel size automatically. This functionality can be easily extended to most SEM-generated images, provided that the structure of the header is known. Through this panel, it is also possible to save and load a specific parameter set once the user is satisfied with it.

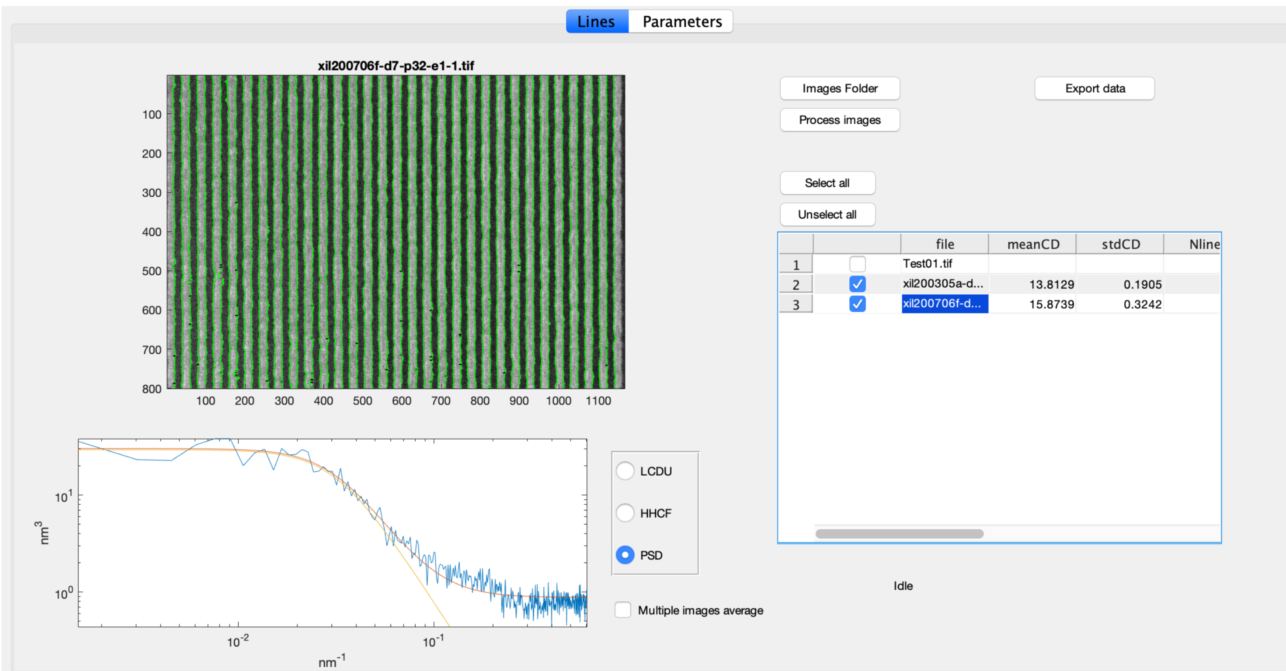

Figure 1. Opening tab of the GUI.

2.2 Image pre-processing

SEM images can be affected by several artifacts that have a negative impact on the quality and reliability of CD and LWR measurements. For this reason, SMILE allows the user to process the images before the analysis. The pre-processing step is crucial to obtain reliable results.

2.2.1 Region of interest selection

Some SEM images may be affected by field-dependent aberrations that cause the edges of the image to be distorted, they may include regions of the object that should not be analyzed, or may have a label that needs to be avoided. SMILE allows the user to select a region of interest (ROI) either by clicking and dragging a rectangle over the image displayed in the parameter tab or by specifying the edges, the width and the height of the ROI in four editable text fields.

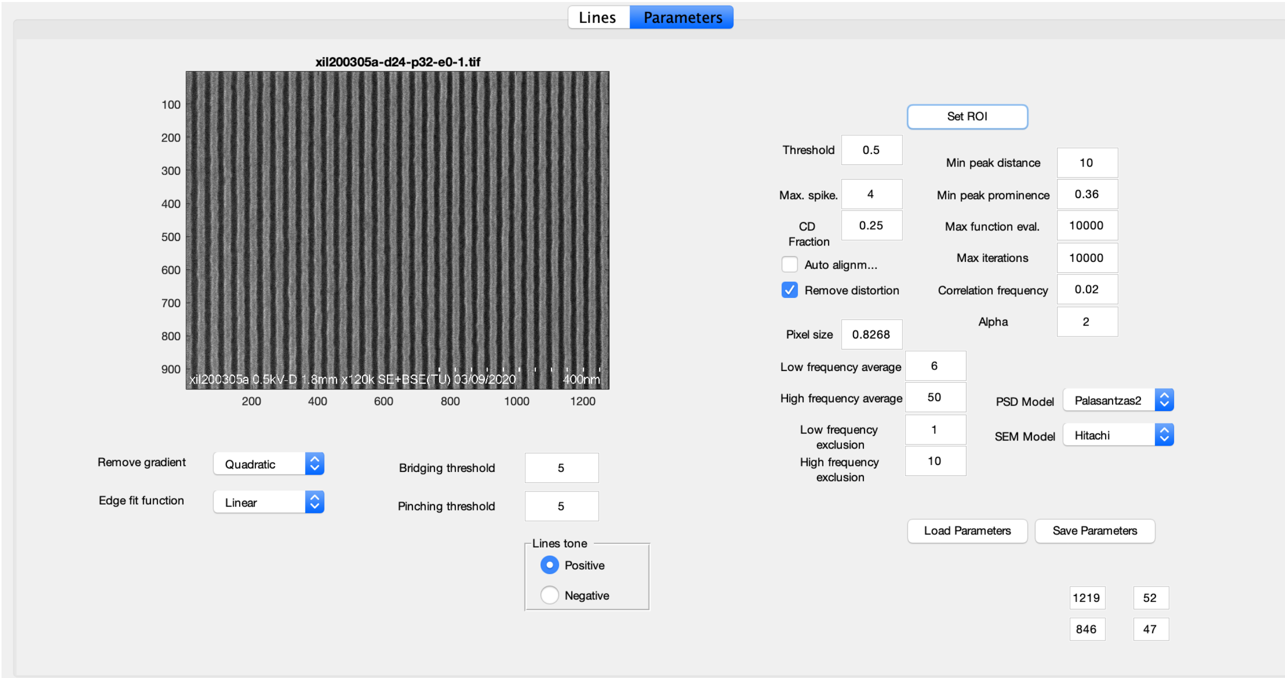

Figure 2. Parameter panel.

2.2.2 Rotation correction

For SMILE to perform a correct analysis, the lines in the image should be vertically oriented, i.e. aligned with the columns of the image grid. The image alignment in industrial CD-SEM tools is usually controlled to a very high degree, however, especially when using a standard SEM, images may have a slight rotation that can influence the edge detection algorithm and induce a bias in the measured critical dimension value. To remove the effect of image rotation, the user can select the *Auto alignment* checkbox. SMILE will use the Radon transform to determine the angle of the lines and will rotate the image back accordingly. It is important to notice that the interpolation performed by the rotation function acts as a low-pass filter and attenuates the highest frequencies in the image and can therefore influence the measurement of the LWR.

2.2.3 Image flattening

In some cases, SEM image can display an artificial intensity gradient that will induce a local change in the CD of the features in the image. To correct for this effect, SMILE offers the possibility to fit a low-degree polynomial to the image. Dividing the image by the fitted polynomial, the intensity becomes uniform over the field of view. An example of this process is shown in figure 3A where a strong intensity gradient was added to the image before applying a quadratic polynomial fit to it. The corrected image is shown in figure 3B. In this example the artificial intensity gradient we applied is much stronger than the cases we observe in experimental data, however, even a relatively small intensity gradient can result in a difference of several nanometers in the CD measured in different areas of the image. Through the GUI, the user can choose to fit a second or third degree polynomial to the image. The polynomial fit and the renormalization of the image intensity is performed twice, to improve the result quality.

2.2.4 Distortion correction

Another artifact that can be observed in SEM images is a global image distortion or shearing that arises from thermal or mechanical drifts of the sample stage. If the drift is constant throughout the image acquisition, this distortion is the same in all the lines. The software gives the user the possibility to mitigate this problem by subtracting the median shape of the edges profiles from each edge. After calculating the median of all the

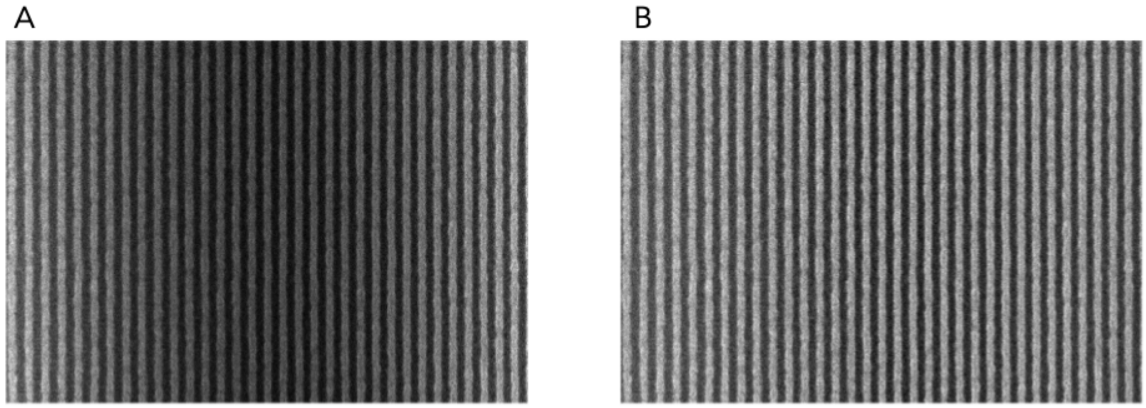

Figure 3. **A.** SEM image with an artificial intensity gradient across the horizontal direction. **B.** The same image after applying the flattening procedure with a quadratic polynomial.

profiles, the result is further smoothed with a moving-average filter with a $N = 11$ pixel kernel. Let $E_i(x)$ be the detected edge profiles in the image, the distortion correction takes the form:

$$E_i^*(x) = E_i(x) - \frac{1}{N}\sum_{x-N/2}^{x+N/2} \mathrm{M}_i\left(E_i(x)\right), \tag{1}$$

where M_i is an operator that calculates the median of the edge profiles for each row of the image. This process is carried out separately for leading and trailing edges. Since this correction is applied after the edge detection step, it is not a pre-processing operation in itself, but we included it here because it addresses a problem inherent to the SEM image.

As a last step in the image pre-processing module, the image is normalized to its median value. If I is the processed image and S its median value calculated along the image columns. The normalized image is $\hat{I}$ becomes:

$$\hat{I} = \frac{I - \min(S)}{\max(S) - \min(S)} \tag{2}$$

2.3 Edge detection algorithms

Once the images have been prepared for analysis, the software proceeds to detect the edges of the lines. In SMILE, this is a two-step process. The first step consists in the identification of the approximate location of the edges. This is done by averaging the image along its columns and determining the average profiles of the lines. Let $S(x)$ be the result of this average, shown in figure 4A. We identify the approximate edge positions as the locations where $S(x)$ has maximum positive and negative slopes. Note that this is an arbitrary choice since the effective position of the edge has also been identified as the location of the intensity peak generated by the secondary electron effect.[3] The extreme slope points of $S(x)$ are selected among the local maxima and minima of $dS(x)/dx$ as shown in figure 4B. The selection criterion is based on the minimum distance between peaks and on the peaks prominence which is defined as the minimum drop on either side of the peak before encountering a value higher than the peak itself. These two parameters are controlled by the user through the parameter tab in the GUI. The edges are subdivided into leading edges, where the derivative is positive and trailing edges, where the derivative is negative. The user can also set the tone of the lines in the GUI: setting the tone to *negative* will flip the the sign convention for the edge discrimination swapping the leading and the trailing edges. The edges locations are sorted and paired to define the single lines and a first estimate of the lines CD and pitch is calculated.

Once the general edge locations are known, the program determines the effective edge profiles for each line. The algorithm iterates over all the rows in a line and extracts the local profile of the line, then looks for the intercept of the profile with a user-defined threshold. SMILE allows the user to choose between three edge

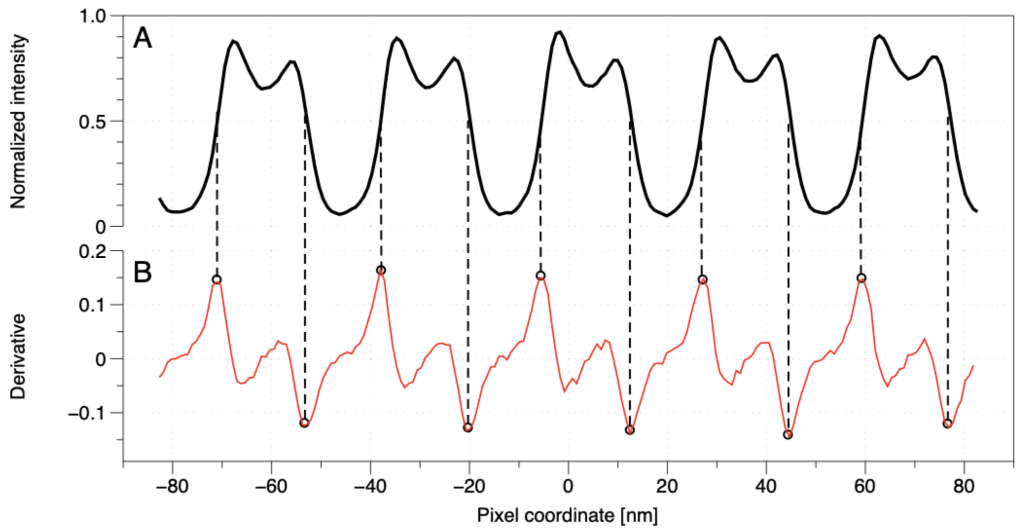

Figure 4. **A.** Average line profiles obtained from the sum of the image rows. **B.** Derivative of the line profiles shown in A. The leading edges and the trailing edges are expected in the local maxima and minima of the derivative.

detection algorithms. For each method the user defines an interval centered on the average edge position where to search for the intercept. The width of the interval is specified as a fraction of the estimated CD.

The *Threshold* method, represented in figure 5B, consists in smoothing the line profile with a convolution filter. The kernel of the filter is a Gaussian profile with a size of 11 pixels and a full-width-half-maximum of three pixels. The smoothing is performed in the direction orthogonal to the line and should therefore affect the line roughness less than filters that average multiple rows in order to improve the signal to noise ratio. The software searches for the two profile points (x_1, I_1) and (x_2, I_2) that are immediately above and below the threshold T and determines the edge position E as:

$$E = \frac{x_1 - x_2}{I_2 - I_1}(T - I_1) + x_2 \tag{3}$$

For this method the CD fraction should be kept between 0.4 and 0.5. Lower values will run the risk of not finding and intercepting with the threshold and higher values may detect the threshold intercept with the previous or with the following line.

The *Linear* method, represented in figure 5C, fits a line to the profile in the specified interval and determines the position of the edge as the intercept of the line with the threshold. This method is the fastest and it does not require any smoothing of the data, however, it requires for the user to select a suitable CD fraction interval. Typical values range from 0.1 to 0.2, depending on the CD of the lines and on the quality of the SEM image. Using larger values may result in a line with a steepness lower than the one of the actual profile which may cause a significant error in the estimated edge position. Selecting a small interval though, limits the number of profile points used for the linear fit, which becomes more sensitive to noise.

The *4-poly* method, represented in figure 5D, fits a 4-degree polynomial to the selected profile interval and determines all the real-valued solutions of the equation

$$P(x) - T = 0, \tag{4}$$

where $P(x)$ is the polynomial and T is the threshold. The algorithm determines the edge position as the closest of these solutions to the average edge location. This method does not require any data smoothing, but is the most computationally intensive. Compared to the *linear* method though, it can make use of a larger interval which results in a higher quality of the fit and can better account for the secondary electron peak effect.[3] The CD fraction for this method can be set to values up to about 0.3, depending on the CD of the lines and on the quality of the SEM image.

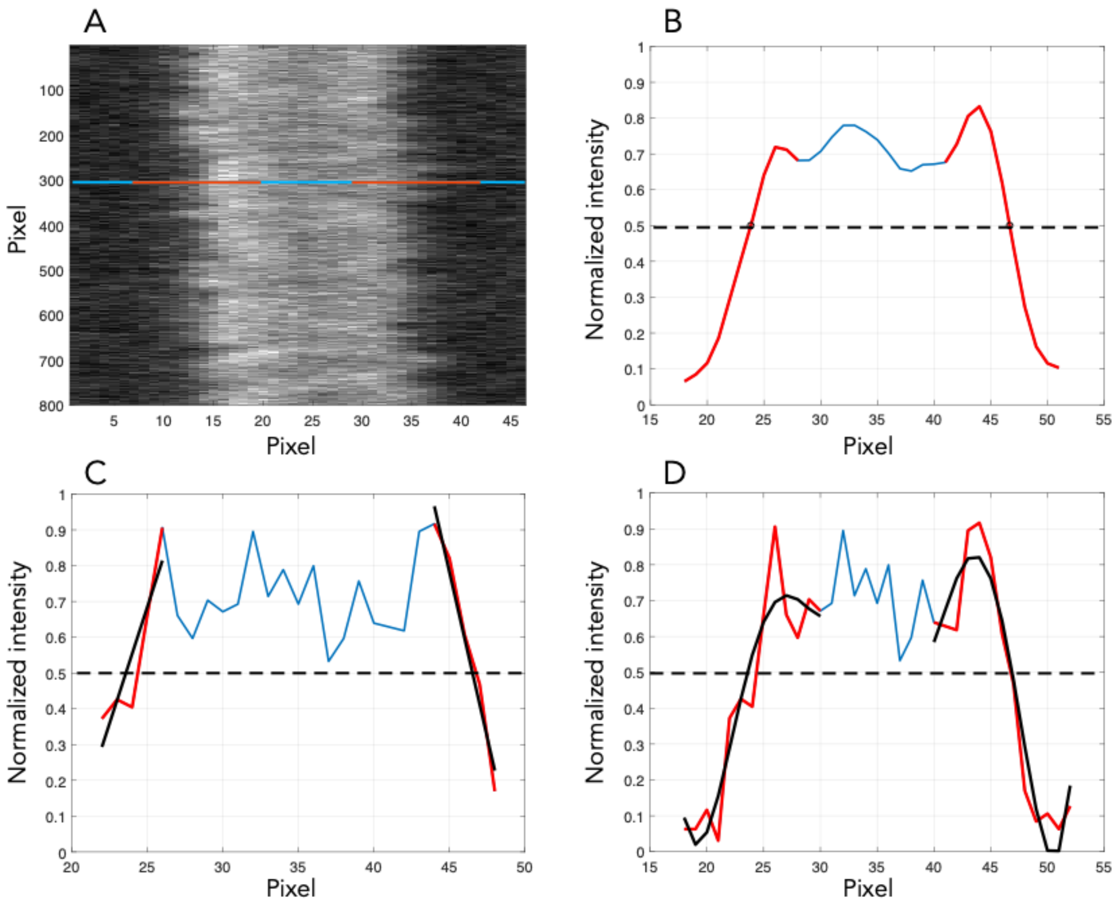

Figure 5. **A.** SEM line segment. The linear cross section of the line is shown in blue and the intervals where the software looks for the edges are shown in red. **B.** Threshold edge detection example. The profile is filtered before the software calculates its intercept with the specified threshold. **C.** Linear edge detection example. The software fits a line to the profile in the interval where the edge is expected. **D.** Polynomial edge detection example. The software fits a 4 degree polynomial to the profile in the interval where the edge is expected.

Once the edges have been determined, the software proceeds to check for undetected edge points, spikes, bridge defects and pinch defects. Undetected edge segments may be present when using the *4-poly* algorithm, if there are no real solutions of equation 4. The software will display these points with yellow dots on the image displayed on the *lines* tab panel. Profile points where the absolute value of the detected edge's gradient is higher than a user-specified threshold are marked as spikes. The gradient threshold is specified in pixels in the *parameters* tab panel on the *Max.spike.* text box. Spikes are visualized as black dots on the profiles. Pinch defects are determined as points where the line width falls below a user-specified pixel threshold. The threshold can be set using the *Pinching threshold* text box in the *parameters* tab panel and the pinch defects are visualized as blue dots on the profiles. Bridge defects are detected as profile points where the distance between the trailing edge of a line and the leading edge of the following line falls below a user-specified pixel threshold that can be set using the *Bridging threshold* text box in the *parameters* tab panel. The bridge defects are visualized as red dots on the profiles.

In order to perform further roughness analysis, SMILE needs continuous edge profiles and for this reason, undetected profile points, spikes, bridge and pinch defects are substituted with the median value of the non-defective profile points.

2.4 Roughness calculation

Line width roughness and line edge roughness can be estimated from the detected profiles. A simple, rough estimation of the LWR can be done calculating the average standard deviation of the line widths. Let $T_i(x)$ and $L_i(x)$ be the trailing edge and leading edge profiles of the i_{th} line and let $1 < x < M$ be the pixel coordinate of the profiles. The corresponding line width profile is calculated as

$$W_i(x) = T_i(x) - L_i(x). \tag{5}$$

The average standard deviation of the N lines detected in a single image is:

$$\overline{\sigma} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{1}{M-1}\sum_{x=1}^{M}\left(W_i\left(x\right) - \mu_i\right)^2} \tag{6}$$

where μ_i is the average value of $W_i(x)$. The line width roughness is calculated as:

$$\mathrm{LWR} = 3\overline{\sigma}. \tag{7}$$

This LWR definition is affected by two potential systematic effects.[4] The first one is the finite length of the measured lines which limits the sampling of the low frequencies. If the lines in the images are too short, the low frequency component of the roughness are not influencing the measured standard deviation thereby leading to an underestimation of the roughness value. The second one is the presence of high frequency noise that affects SEM images and propagates to the detected line edges artificially increasing the standard deviation. This bias can be removed by fitting a suitable model to the LWR power spectral density (PSD) or to the height-height-correlation function.[5] The PSD can be estimated as the square module of the line width profile's Fourier transform:

$$PSD(\nu) = \frac{\Delta x^2}{L}\left|\sum_{x=1}^{L} W(x)e^{-i\nu\Delta x}\right|^2. \tag{8}$$

One of the properties of the PSD is that its integral corresponds to the variance of the signal, therefore the LWR can also be calculated from the square root of the area subtended by the PSD curve. SMILE calculates and displays the average PSD for all the lines detected in the selected region of interest as shown in figure 1. A model commonly used to describe the height-height correlation function of resist edge profiles is the Kohlrausch-Williams-Watts function:

$$HHCF(x) = \sigma^2 e^{-(x/\xi)^{2\alpha}} \tag{9}$$

where ξ is the correlation length and α is the roughness exponent, or Hurst coefficient. The autocorrelation function $G(x)$ can be expressed using the height-height correlation function as $G(x) = 2\sigma^2 - 2HHCF(x)$. The autocorrelation function and the power spectral density are Fourier transform pairs:

$$PSD(\nu) = \int 2\sigma^2\left(1 - e^{\left(-\frac{R_v(x)}{\xi}\right)^{2\alpha}}\right)e^{-ix\nu}\mathrm{d}x + Nl \tag{10}$$

When α is equal to 0.5, it is possible to approximate the PSD with the analytical expression:

$$PSD(\nu) = \frac{\xi\sigma^2}{\left(1+\xi^2\nu^2\right)^{2\alpha}} + Nl \tag{11}$$

where Nl represents the contribution of the white noise. To allow α values different from 0.5, it is possible to introduce an additional parameter at the denominator of eq[6]

$$PSD(\nu) = \frac{\xi\sigma^2}{\left(1+a\xi^2\nu^2\right)^{1+\alpha}} + Nl \tag{12}$$

If a more accurate autocorrelation function model is available, it is always possible to build a corresponding PSD model to fit the measured data and remove the image noise more effectively using an integral expression as the

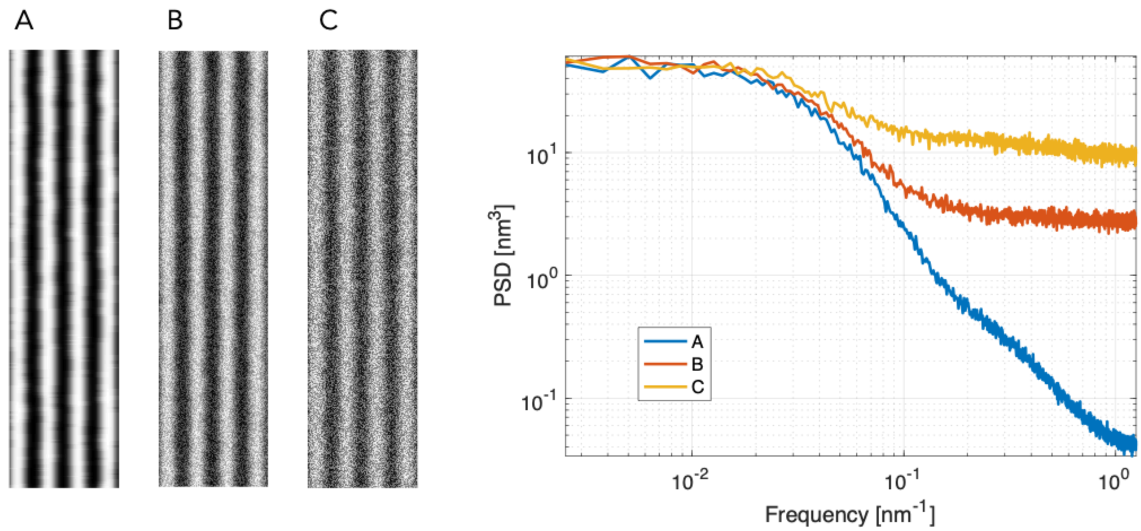

Figure 6. The images shown in this picture are samples of the line space patterns used to test the LWR unbiasing procedure. **A** is the image without white noise, **B** is the image with an intermediate amount of noise and **C** is the image with the highest amount of noise. The plot on the right shows the corresponding average PSD curves obtained using the *Linear* edge detection algorithm.

one presented in equation 10. In SMILE, the user can currently choose between a few PSD models, including the ones described by equations 11, 12 and 10. The fit is performed using the MATLAB fminsearch function, based on the Nelder-Mead Simplex method.[7] For best results, it is necessary to define a reasonable initial guess for the model's parameters. The user must specify an initial value for the correlation frequency (in nm^{-1}) and for *alpha* in the parameters tab. The software will estimate the value Nl by averaging the high-frequency values of the PSD. The user can use the *High frequency average* text box in the parameters tab to specify how many spectral components to average. The software will also estimate the value of PSD(0) as the difference between the average of the low-frequency components of the PSD and Nl. The user can specify the number of low-frequency spectral components to average using the *Low frequency average* text box in the parameters tab. It is also possible to specify the number of spectral components of the PSD to exclude from the fit using the *High/Low frequency exclusion* text boxes in the parameters tab. This can prove useful when using the *Auto alignment* feature which will introduce a high frequency filter in the data.

Once the fit is performed, SMILE visualizes the fitted model and its unbiased version where the Nl parameter is set to 0. The unbiased value of the LWR is calculated as the integral of the measured PSD minus the fitted value of Nl.

2.5 Simulations

We performed a simulation to test the performance of the LWR-unbiasing procedure. We generated a set of artificial SEM images with increasing amounts of white noise as shown in figure 6. Each image consists of a 1:1 line and space pattern with 183 lines with a CD of 22 nm and an image pixel size of 1 nm. The lines' roughness was generated using the PSD model described by equation 11 to simulate a LWR of 2.5 nm. We then processed the images using SMILE with the *Linear* edge detection algorithm and the matching PSD model. The unbiased LWR results are shown in figure 7 and demonstrate that the edge detection algorithm and the unbiasing procedure are effective, provided that the PSD model used to fit the data is correct.

3. CONCLUSIONS

We presented SMILE, an open source software for the metrology of line and space patterns on SEM images. SMILE was developed for the analysis of SEM images of photoresists and wafer structures. We have been using this software for EUV resist screening of wafers produced with an interference lithography tool operating at the

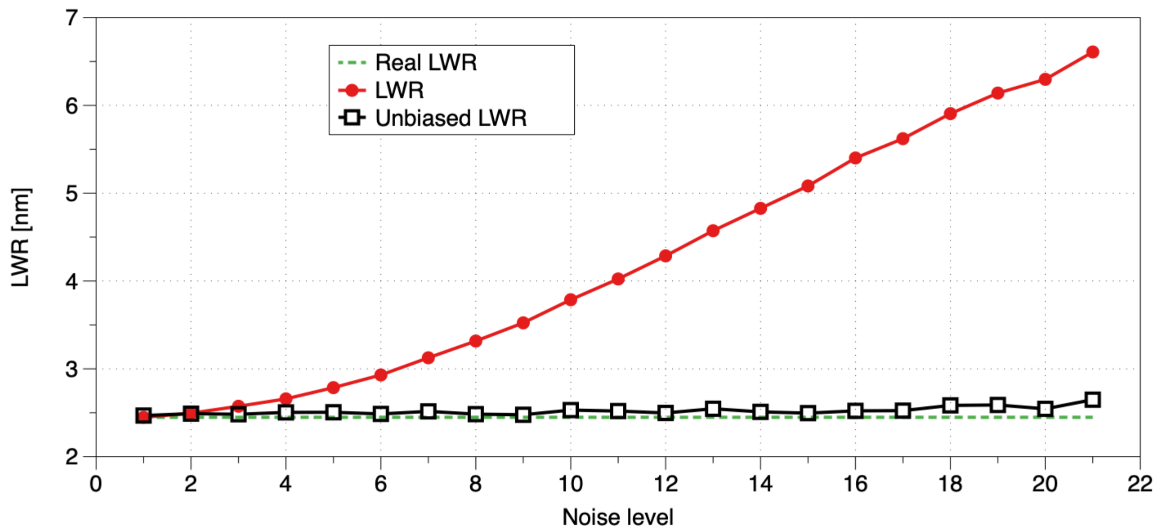

Figure 7. Biased and unbiased LWR with increasing amounts of white noise.

Paul Scherrer Institut. Before implementing this software for our regular use, we have tested it rigorously and benchmarked it with some of the commercial and noncommercial software availabe to us. We compared the relevant parameters, such as CD, LWR and unbiased LWR, with those obtained by other software and gained a high level of confidence.

The main motivation for the development of an open source software was the necessity to adapt the metrology procedure to specific experimental conditions (SEM model, aerial image generation equipment, wafer development environment). The software consists of a simple graphical user interface and a collection of functions and procedures to determine the CD of the lines in the image and evaluate their roughness. We implemented and tested several simple edge detection algorithms and we investigated the performance of the procedure we choose for the noise bias removal in the LWR measurement.

SMILE is being developed using MATLAB 2018a and it relies on the signal processing toolbox, the image processing toolbox and the optimization toolbox. Our plan is to migrate the software to an open source platform such as Python or Octave, to make it easier for a broader community to access it. We also plan to add a module for contact-holes metrology which has already been developed and tested separately.

You can obtain the current beta version by contacting the authors and the advanced versions will be available on a web platform. We hope that with SMILE we can contribute to the lithography community by offering a open-source metrology software which can be improved and expanded by single users and by the community at large.

ACKNOWLEDGMENTS

The authors wish to thank all the users and early adopters of SMILE for their support and feedback.

REFERENCES

[1] Lorusso, G. F., Sutani, T., Rutigliani, V., van Roey, F., Moussa, A., Charley, A.-L., Mack, C., Naulleau, P., Constantoudis, V., Ikota, M., Ishimoto, T., and Koshihara, S., "The need for LWR metrology standardization: the imec roughness protocol," in [*Metrology, Inspection, and Process Control for Microlithography XXXII*], Ukraintsev, V. A., ed., **10585**, 37 – 47, International Society for Optics and Photonics, SPIE (2018).

[2] MATLAB, [*version 9.4.0 (R2018a)*], The MathWorks Inc., Natick, Massachusetts (2018).

[3] Seiler, H., "Secondary electron emission in the scanning electron microscope," *Journal of Applied Physics* **54**(11), R1–R18 (1983).

[4] Lorusso, G. F., Rutigliani, V., Van Roey, F., and Mack, C. A., "Unbiased roughness measurements: Subtracting out sem effects, part 2," *Journal of Vacuum Science & Technology B* **36**(6), 06J503 (2018).

[5] Papavieros, G., Constantoudis, V., and Gogolides, E., "Allowable SEM noise for unbiased LER measurement," in [*Metrology, Inspection, and Process Control for Microlithography XXXII*], Ukraintsev, V. A., ed., **10585**, 471 – 480, International Society for Optics and Photonics, SPIE (2018).

[6] Palasantzas, G., "Roughness spectrum and surface width of self-affine fractal surfaces via the k-correlation model," *Phys. Rev. B* **48**, 14472–14478 (Nov 1993).

[7] Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E., "Convergence properties of the nelder–mead simplex method in low dimensions," *SIAM Journal on Optimization* **9**(1), 112–147 (1998).